UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingenieros Informáticos

On the Limitations of Black-Box Constructions in Cryptography

D O C T O R A L T H E S I S

Submitted for the degree of Doctor by:

Emanuele Giunta
M.Sc. in Mathematics

Madrid, 2025


UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingenieros Informáticos

Doctoral Degree in Software, Systems and Computing

On the Limitations of Black-Box Constructions in Cryptography

D O C T O R A L T H E S I S

Submitted for the degree of Doctor by:

Emanuele Giunta
M.Sc. in Mathematics

Under the supervision of:
Dr. Ignacio Cascudo

Madrid, 2025

Title: On the Limitations of Black-Box Constructions in Cryptography

Author: Emanuele Giunta

Doctoral Programme: Software, Systems and Computing

Thesis Supervision:

Dr. Ignacio Cascudo, Associate Research Professor, IMDEA Software Institute (Supervisor)

External Reviewers:

Dr. Dennis Hofheinz, Full Professor at the Department of Computer Science, ETH Zurich

Dr. Luisa Siniscalchi, Assistant Professor at the Department of Applied Mathematics and Computer Science, DTU Denmark

Thesis Defense Committee:

Thesis Defense Date:

This thesis is partially supported by the PRODIGY Project (TED2021-132464B-I00) the SECURING Project (PID2019-110873RJ-I00), both funded by MCIN/AEI/10.13039/501100011033. This work was also supported by the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme in the scope of the CONFIDENTIAL6G project under Grant Agreement 101096435. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

To my sister
Ester

Acknowledgement

First, my thanks go to my advisor, Ignacio Cascudo. For having welcomed me in such a beautiful and lively city which became a place I would call home during the last four years. For having been by my side as I made my first shaky steps into research, always available to discuss regardless of how silly my questions were. Thanks also for having always granted me the freedom to explore new topics, and having had the patience for this explorations to bear fruits without ever pressuring me. Special thanks finally for having spent uncountably many hours with me trying to figure out the new system to deposit this thesis and eventually defend it.

Thanks to Dario Catalano, who has been to me a de facto co-advisor. For your incredible enthusiasm in research, which I admire, and the many thrilling meetings discussing ideas we believed (or wished) to be breakthroughs only to break them the day after. Thanks for your trust, for having always been open to sharing problems you were interested in, and listen to those I would from time to time come up with. Finally, thanks for having advised me to read the one paper that started my journey on impossibility results and lower bounds, eventually leading to this thesis.

Thanks also to Dario Fiore, who, along with Ignacio, has helped and guided me throughout these years. I have always been impressed by the great range of topics you work on, and the dedication you put in each of them (I hope one day to have half the time managing skills you have). Also, I wanted to thank you and Ignacio, for having made IMDEA such a healthy and stimulating environment to work in, for having always pushed us students to share our interests and cooperate, and for the many organized activities and events (last but definitely not least, Eurocrypt '25!).

A huge thank go to Dennis Hofheinz and Luisa Siniscalchi, who promptly agreed to review this thesis and go through the paperwork this involves. Thanks also to Matteo Campanelli, Oriol Farras, Dario Fiore, Luisa Siniscalchi and Eduardo Soria for having accepted to join as committee members and Maria Isabel Gonzalez Vasco and Ida Tucker for having offered as substitute.

I would like to thank Alistair Stewart and all the researchers and people who work at Web 3.0 Foundation for having welcomed me as an intern for six months. Working with you all has been an incredibly enriching opportunity for me to peek beyond theory. I also personally feel lucky to have been able to work in person from Zug, which is a beautiful city with gorgeous surroundings.

Research, as I came to realize during my PhD, is hardly ever the product of a single individual bashing his head on the wall, but rather the joint effort of many people, all with different insights and perspectives. I feel for this reason extremely grateful to and proud of all my coauthors: Gennaro Avitabile, Vincenzo Botta, Davide Carnemolla, Daniele Cozzo, Ignacio Cascudo, Dario Catalano, Dario Fiore, Rosario Gennaro, Kristina Hostakova, Marcin Mielniczuk, Francesco Migliaro and Alistair Stewart.

A big part of the credit for making my IMDEA journey a great growth opportunity goes to

all the talented cryptographers and amazing people in my group. To the past "cryptonians" generation: Matteo Campanelli, Mario Carrillo, Peter Chvojka, Antonio Faonio, Lydia Garms, Dimitris Kolonelos, Chrysoula Oikonomou, Anaïs Querol, Miguel Morona, Istvan Seres, Ida Tucker and Dimitrios Vasilopoulos. And to the current one: Hamza Abusalah, Gaspard Anthoine, Gennaro Avitabile, David Balbas, Claudia Bartoli, Diego Castejón, Pablo Castellanos, Daniele Cozzo, Javier Gómez, Antonio Guimarães, Aoxuan (Douglas) Li, Mahak Pancholi and Damien Robissout. Thanks to all of you for all the moments we shared, from the jokes at lunchtime to the in-depth discussions at the whiteboard.

I also want to thank all the people who have been working at IMDEA during those years. Thanks for all the fun times we had, for having introduced me to hiking (special thanks to Ida and Nicolas) and climbing. Thanks also for having withstood all my boardgames-related discussions (special thanks to Dimitris) and having played with me on several occasions. Many thanks also to those who shared the office with me: Daniel Jurjo, Ignacio Ballesteros, Kyveli Doveri and Remy Defossez.

Having exhausted all the work-related acknowledgments, I would like to conclude by thanking all those people who did not directly influence my work, but were nevertheless essential for me to get through these years.

First, thanks to all the friends who have always been there. To Alessio, Andrea (Dedo), Jeremy, Laura, Lorenzo (Lollo) and Matteo (Cava), for having been there to listen to me when I needed the most. To Anthony, Ivan, Marco and Stefano, for all the summer reunions that always reminded me how Sicily is still a place I belong to.

Thanks to the first groups of people I met in Madrid and welcomed me as a family at a time when I could not feel more lost: to Qingyang (Jaina) and Tan, and later to Bea, Liliu (Lili) and Xinye (Olivia). Thanks for all the travels, the fun times at Tan & Olivia's house between hot-pots and *landlord* or *Coup* games, the marathons, the beach volley matches, and having inspired me to learn Chinese.

Thanks to all those people I discovered the *Estar Cafe* with for the first time and, more generally, had a chance to play boardgames with: Daniel, František, Ida, Kasra, Lubica, Lydia, Nicolas, Nikita, Paloma, Raluca, Silvia.

Thanks to the Estar Cafe family, for having so openly welcomed me, given me a reason to learn Spanish and a place to practice. I feel extremely grateful to all the amazing people I met there. To list some of them: Adrian, Agustina, Alberto (no te vas!), Antonio, Cesar, Claudia, Dani (lobo), Diego (estrella sin forma), Edo (narrador), Fran (dueño y lobo albino), Gema, Jeff, Korbi, Lisa (flautista), Luciana, Lupe, Maria, Monica, Nacho (alcalde), Raul (moon lover and bringer of nightmares), Sergio, Vincent (reina de hielo). Thanks also to those among them who introduced me to the *Spirit Island Sect*™, and to karaoke (along with Nana), which made me realize I was about to leave Spain without knowing any Spanish songs.

Thanks to Giorgia, who stood by my side during the troubled beginnings of my PhD.

Thanks to Ziyi (Sara) who, among other things, made me realize the importance of a healthy

work/life balance.

Finally, thanks to my family: my sister Ester, my mother Maria, my father Francesco, and my grandmother Maria Pia. Words are really of no use here.

Abstract

Cryptography is the science of secure communication. Originating as an esoteric discipline based on heuristics, it underwent a mayor paradigm shift in the past century. Modern cryptographic schemes are indeed formally proven secure through reductions. Such approach allows arguing that security of a scheme holds as long as its base primitives are secure. A popular approach to achieve this is through black-box constructions. These informally rely on the underlying primitives only through its intended interface, therefore not utilizing any additional structure that only concrete instantiations may enjoy.

Being black-box is a desirable feature for many constructions. As a design principle, it allows for greater modularity and a higher level of abstraction. For concrete efficiency and security, it means no expensive non-black techniques were deployed. The latter indeed often significantly affects performances, as in the case of garbling or indistinguishability obfuscation, or requires strong assumptions, as for SNARKS. Although desirable, a large body of research starting from Impagliazzo and Rudich's results (STOC '89), has shown that in some cases black-box constructions are not possible, or suffer limitations.

In this thesis we push forward our understanding of the limitation of black-box constructions. We do so by providing new negative results and lower bounds for several primitives from standard cryptographic objects.

First, we focus on primitives that can be realized from a cryptographic prime order group, i.e. where the discrete logarithm problem is hard. Such groups are the cornerstone of modern public key cryptography and many primitives are known to exist based on them. In the first three chapters, we study whether digital signatures, vector commitments, and non-interactive zero-knowledge (NIZK) could be realized from a black-box group. We find that in this setting, signatures only exist for polynomially small message spaces, vector commitments cannot be succinct, and NIZK do not exist for many useful languages.

Second, we turn our attention to cryptographic group actions, discussed in the fourth chapter. These represent one of the few post-quantum sources of hard problems currently available. A problem of many threshold schemes in this setting is that computing the action of a secret-shared group element requires high round complexity. Specifically round-robin protocols, where parties interact sequentially, are currently the only black-box option. We show that such limitation is inherent. Specifically, any protocol (black-box in the underlying group action) computing the action of a shared secret, requires a number of rounds linear in the privacy threshold.

Third, in the fifth chapter we investigate black-box realizations of Anamorphic Encryption (AE). AE is a new encryption paradigm, guaranteeing privacy even against a dictator who can enforce the usage of an encryption scheme of their choice and observe all user's secret keys. AE is defined with respect to a specific public key encryption (PKE) scheme, providing an alternative, indistinguishable, encryption mode. Generic constructions that are black-box with respect to the underlying PKE, and thus apply to any PKE, are known. However, these only allow communicating few bits per ciphertext, specifically logarithmically many in the

security parameter. We prove this to be the best any black-box construction can achieve. Furthermore, we show how black-box constructions cannot satisfy stronger security notions, such as Fully Asymmetric AE, recently introduced by Catalano *et al.* (Eurocrypt '24).

Resumen

La criptografía es la ciencia de la comunicación segura. Originada como una disciplina esotérica basada en heurísticas, experimentó un cambio de paradigma significativo en el último siglo. Los esquemas criptográficos modernos son, de hecho, formalmente demostrados como seguros mediante reducciones. Este enfoque permite argumentar que la seguridad de un esquema se mantiene siempre que sus primitivas base sean seguras. Un enfoque popular para lograr esto es a través de construcciones "black-box", que informalmente dependen de las primitivas subyacentes solo a través de su interfaz prevista y por tanto no utilizan ninguna propiedad matemática adicional de una implementación concreta de estas primitivas.

Ser black-box es una característica deseable para muchas construcciones. Como principio de diseño, permite una mayor modularidad y un nivel más alto de abstracción. Para la eficiencia y seguridad concretas, significa que no se han implementado técnicas costosas que no son black-box. Estas últimas, de hecho, a menudo afectan significativamente el rendimiento, como en el caso de "garbling schemes" o "indistinguishability obfuscation", o requieren suposiciones fuertes, como en el caso de los SNARKS. Aunque deseable, una gran cantidad de investigaciones, comenzando con los resultados de Impagliazzo y Rudich (STOC '89), han mostrado que en algunos casos las construcciones black-box no son posibles o presentan limitaciones.

En esta tesis, avanzamos en nuestra comprensión de las limitaciones de las construcciones black-box. Lo hacemos proporcionando nuevos resultados negativos y límites inferiores para varias primitivas de objetos criptográficos estándar.

Primero, nos centramos en primitivas que pueden realizarse a partir de un grupo criptográfico de orden primo, es decir, donde el problema del logaritmo discreto es difícil. Dichos grupos son la piedra angular de la criptografía de clave pública moderna, y se sabe que muchas primitivas existen basadas en ellos. En los tres primeros capítulos, estudiamos si las firmas digitales, los vector commitment y pruebas de conocimiento cero no interactivo (NIZK) podrían realizarse a partir de un grupo black-box. Encontramos que, en este contexto, las firmas solo existen para un espacio de mensajes polinomialmente pequeño, los vector commitments no pueden ser sucintos y las NIZK no existen para muchos lenguajes útiles.

En segundo lugar, en el cuarto capítulo dirigimos nuestra atención a las acciones de grupo criptográficas. Estas representan una de las pocas fuentes de problemas difíciles en el contexto post-cuántico actualmente disponibles. Un problema de muchos esquemas de umbral en este contexto es que calcular la acción de un elemento de grupo compartido de forma secreta requiere una alta complejidad de rondas. Específicamente, los protocolos de ronda en cadena, donde las partes interactúan secuencialmente, son actualmente la única opción black-box. Mostramos que tal limitación es inherente. Específicamente, cualquier protocolo (black-box en la acción del grupo subyacente) que calcule la acción de un secreto compartido, requiere un número de rondas lineal en el umbral de privacidad.

En tercer lugar, en el quinto capítulo investigamos las realizaciones black-box de la Cifrado Anamórfico (AE). AE es un nuevo paradigma de cifrado que garantiza privacidad incluso contra un dictador que puede imponer el uso de un esquema de cifrado de su elección y

observar todas las claves secretas de los usuarios. AE se define con respecto a un esquema específico de cifrado de clave pública (PKE), proporcionando un modo de cifrado alternativo e indistinguible. Se conocen construcciones genéricas que son black-box con respecto a la PKE subyacente, y que por lo tanto se aplican a cualquier PKE. Sin embargo, estas solo permiten comunicar un número limitado de bits por cada texto cifrado. Probamos que este es el mejor rendimiento que cualquier construcción black-box puede lograr. Además, mostramos cómo las construcciones black-box no pueden satisfacer nociones de seguridad más fuertes, como la "Fully Asymmetric AE", introducida recientemente por Catalano *et al.* (Eurocrypt '24).

Table of Contents

List of Figures

List of Publications

This thesis stems from the results obtained in the following 4 papers, published in peer-reviewed academic conferences.

- [32] **On the Impossibility of Algebraic Vector Commitments in Pairing-Free Groups.**
  D. Catalano, R. Gennaro, D. Fiore, E. Giunta – TCC 2022.
- [63] **On the Impossibility of Algebraic NIZK In Pairing-Free Groups.**
  E. Giunta – Crypto 2023.
- [42] **Round-Robin is Optimal: Lower Bounds for Group Action Based Protocols.**
  D. Cozzo, E. Giunta – TCC 2023.
- [38] **Limits of Black-Box Anamorphic Encryption.**
  D. Catalano, E. Giunta, F. Migliaro – Crypto 2024.

Other papers co-authored during my PhD include:

- [28] **On Interactive Oracle Proofs for Boolean R1CS Statements.**
  I. Cascudo, E. Giunta – FC 2022.
- [34] **Efficient and Universally Composable Single Secret Leader Election from Pairings.**
  D. Catalano, D. Fiore, E. Giunta – PKC 2023.
- [33] **Adaptively Secure Single Secret Leader Election from DDH.**
  D. Catalano, D. Fiore, E. Giunta – PODC 2022.
- [35] **Anamorphic Encryption: New Constructions and Homomorphic Realizations.**
  D. Catalano, E. Giunta, F. Migliaro – Eurocrypt 2024.
- [64] **Unbiasable Verifiable Random Functions.**
  E. Giunta, A. Stewart – Eurocrypt 2024.
- [27] **Verifiable Secret Sharing from Symmetric Key Cryptography with Improved Optimistic Complexity.**
  I. Cascudo, D. Cozzo, E. Giunta – Asiacrypt 2024.
- [36] **Generic Anamorphic Encryption, Revisited: New Limitations and Con-**

structions.
D. Catalano, E. Giunta, F. Migliaro – Eurocrypt 2025.

- [26] **Anamorphic Resistant Encryption: the Good, the Bad and the Ugly.**
D. Carnemolla, D. Catalano, E. Giunta, F. Migliaro – in submission.
- [8] **The Malice of ELFs: Practical Anamorphic-Resistant Encryption without Random Oracles.**
G. Avitabile, V. Botta, E. Giunta, M. Mielniczuk, F. Migliaro – in submission.

Part I

Introduction and Background

Chapter 1

Introduction

1.1 Historical Remarks

The ability to establish private means of communication has been a fundamental problem accompanying mankind throughout history. Early records of such techniques were already present in ancient Egyptian and Greek civilizations. The popular *Caesar Cipher* was used by the roman commanders to exchange sensitive information with their generals.

For centuries thereafter, developments of new schemes closely followed the progress of known attacks. Examples includes Alberti's cipher, presented in his work *de componendis cifris*, and Vigenére's one. Both stemmed indeed from the need to make attacks for substitution ciphers *harder* to carry out. The connection between security and computational power became evident in the context of World War 2. Indeed, the efforts to break enigma-based codes eventually succeeded also thanks to the realization of *the bombe* and *colossus*, impressive set of computing machines.

Up until this point, cryptography was mostly deemed as an art: known techniques were indeed based on heuristics, and security heavily relied on the secrecy of those techniques. The second approach is now addressed as "security through obscurity". This was arguably among the main factors that delayed the development of cryptography, along with several bans on *strong* cryptographic schemes in the decades after WWII.

Provable security. The major revolution that occurred in cryptography over the last half century has been a methodological shift: rejecting security through obscurity, and rather designing schemes that can be *proven secure* (and thus publicly disclosed). A milestone in this sense was laid by the formalization of many fundamental primitives, including semantic security for encryption [67] and unforgeability for signatures [68]. Agreeing on meaningful definitions was indeed the first step toward the realization of schemes eventually achieving them.

Unfortunately, however, *unconditional* security of virtually any cryptographic primitive would have tremendous impact on fundamental problems in complexity theory. Definitive (positive) answers are extremely hard to obtain based on our current knowledge. Hence, security is

typically argued through a *reduction*. Informally, assuming a given problem to be hard, a proof by reduction shows that any attack breaking the target security notion can be used to also solve the hard problem.

Thanks to security reductions, reliance on heuristic assumptions can be effectively mitigated. Indeed, this allows for quickly trusting new schemes whose security reduces to long-standing problems, even though the scheme itself did not receive extensive cryptanalytic attention. Moreover, reductions further allow to focus the cryptanalytic efforts only on a handful of core problems. To this day indeed most cryptographic applications are based only on the hardness of the discrete logarithm (and related problems), factoring, lattice problems (notably LWE), codes, and more recently on isogenies.

Modular constructions. Another positive aspect of security reductions is that they incentivize *modularity*. This often comes in terms of so called *black-box* constructions, i.e. that rely on some base primitive only through its interface. Indeed, it is simpler to reduce the security of a complex protocol to that of its components, rather than reducing it to the base hard problems.

Black-box constructions are desirable for a few reasons. First, they grant freedom to instantiate the base component arbitrarily in order to achieve different properties or efficiency trade-offs. Moreover, should a specific realization of a component be proven insecure, this can easily be replaced in black-box constructions with a different one (providing the same functionality).

Black-box realizations also tend to be more efficient than non-black-box (yet generic in the underlying primitive) ones. A reason this occurs is that currently known non-black-box techniques add significant overhead. This is for instance the case of garbling [110], fully homomorphic encryption (e.g. [59]) or obfuscation [**?**].

Given the robustness, flexibility and maintainability black-box constructions come with, it is usually interesting to study whether a given primitive can be black-box realized from other simpler ones. As it will be discussed in the next section, unfortunately the answer does not always turns out to be positive.

1.2 Separations

One of the weakest cryptographic primitives known to date are *one-way functions* (OWF), that are functions easy to compute but hard to invert on average. If OWF exist, then many generic constructions are known to realize most *symmetric key* primitives, notably pseudo-random generators [73], pseudo-random functions [65] and thus symmetric key encryption, (interactive) commitments [92] as well as (surprisingly!) digital signatures [99]. However, the relation between OWF and public-key encryption (PKE), or stronger primitives, is less clear. To this day indeed it is not known whether the existence of OWFs is enough to imply any PKE scheme.

Relativizing reductions. The first indication of a negative answer to the last point came in the groundbreaking result of Impagliazzo and Rudich [74]. Their work ruled out black-box

(formally *relativizing*) constructions of key-agreement from one-way permutations. Apart from the result itself, the key innovation of [74] lies in the technique. To show their separation they prove the existence of an oracle relative to which one-way functions exist, but key agreement is insecure. This in particular excludes black-box constructions (with a black-box security reduction to the underlying primitive), as such construction should remain secure relative to their oracle.

Following their results many subsequent works adapted their techniques to separate other cryptographic primitives. Some examples include the separations of collision-resistant hash functions [104] and one way permutations [76] from OWF, of public-key encryption from oblivious transfer [61] and of trapdoor functions from PKEs [62].

1.3 Main Contributions

In this thesis we push forward our understanding of the intrinsic limitations of black-box constructions. While all our results are further motivated and detailed in the following sections, here we provide a high level overview. Throughout this thesis we mainly focus on the black-box realization of five different primitives from a given base object of interest. Specifically we investigate: signatures, vector commitments, and non-interactive zero-knowledge from black-box prime order groups (without pairing and of known order); distributed computation of the action of a secretly shared group element over a black-box group action; anamorphic encryption [96] compilers for every PKE through black-box access.

In most cases we do not provide full separations in the sense of Impagliazzo-Rudich [74]. Instead, assuming a black-box construction in the given setting exists, we then derive efficiency bounds. Those are informally summarized below for each case:

1. Signatures in Maurer's GGM [88]: We show the same bound proved in [53] to hold more generally in Maurer's GGM, removing the *linear verification* constraint[1]. Such bound informally implies that any signature scheme unconditionally secure in Maurer's GGM (an idealized model for group computation), whose key is composed of n groups elements, can support at most n messages.

2. Vector Commitments (VC) in Maurer's GGM: we show that when committing to a vector of length n, either commitment or opening size must be $\Omega(\sqrt{n})$. This is further strengthened if the VC is *hiding*, in which case the commitment must have size $\Omega(n)$.

3. Non-Interactive Zero-Knowledge (NIZK) in Maurer's GGM: we show that, for many relativized[2] relations, NIZK cannot exist. More precisely, NIZK of knowledge are impossible for the preimage relation of OWF, which covers the discrete logarithm relation. Moreover NIZK are impossible for hard subset membership languages such as Diffie-Hellman.

4. Action of a shared secret in the GAM: we prove that computing $s \star E$ for s a secret group element t-out-of-n shared, either involves at least t rounds of interaction or t users

[1] i.e. a signature is valid iff a specified system of linear equations involving group elements is satisfied.
[2] i.e. defined relative to the group operation's oracle

can jointly retrieve s. This matches the known round-robin protocol round complexity, proposed for threshold signatures from group actions.

5. Anamorphic Encryption (AE) for any PKE: we show that any realization of AE that is black-box in the underlying PKEs (and thus would apply to *any* secure and correct scheme) can only support polynomially bounded message space. Moreover, in the same setting, stronger notions such as fully-asymmetric AE [35] are unachievable.

1.3.1 Improved Results for Algebraic Signatures

As explained in the introduction of [53], there informally appears to be a mismatch between our theoretical and practical understanding of public key encryption and signature schemes. On the one hand signatures are theoretically weaker objects, being equivalent to OWF. In contrast, public key encryption is known to be black-box separated from many symmetric key primitives, as discussed in Section 1.2. On the other hand, concrete PKEs are easier to realize from simple assumptions, while efficient signatures often rely on the random oracle model (ROM) [12] or require stronger assumptions [45, 108].

To address this gap, [53] focuses on signatures in (variants of) Maurer's GGM, modeling a prime order group of known order. In such model they show that any *unconditionally secure* signatures scheme whose verification key consisting of n group elements can support message space of size at most n. Their result however holds restricting the verification procedure to only check a system of linear equations. Specifically, given verification key $\mathbf{X}$ and signature $(\mathbf{Y}, t)$ for a message m, where $\mathbf{X}, \mathbf{Y}$ are vectors of group elements, the signature verification computes two matrices $A = A(m, t)$ and $B = B(m, t)$, and accept iff $A\mathbf{X} = B\mathbf{Y}$.

Such requirement appears to be natural in the GGM, where constraints of the idealized model only allow to compute linear combinations of given group elements and testing equality among them. However this condition does affect generality, as also acknowledged in [53]. Examples of signatures in the GGM that do not satisfy the given condition are provided in [32]. The first result presented in this thesis is thus a generalization of [53] in two directions.

First of all, we study schemes with *strictly linear verifications*. This is a strict sub-class of the *algebraic* schemes defined in [53], constrained as follows: the signature is assumed to be $(\mathbf{Y}, \mathbf{z})$ with $\mathbf{z}$ a vector of field elements, and verification is limited to compute $A = A(\mathbf{z}, m)$ (with A depending linearly on $\mathbf{z}$), $B = B(m)$ and check $A \cdot \mathbf{X} = B \cdot \mathbf{Y}$. For this class of schemes we show the same strategy of [53] yields an efficient adversary, breaking security *unconditionally*. In the full version of [32] it is also shown that assuming a dependency on $\mathbf{z}$ either quadratic for A or linear for B instead yields to secure signature. This is done through standard arithmetization techniques, and this entirely bypasses usage of the group.

Next, we show their bound to hold for *any* signature scheme that is unconditionally secure in Maurer's GGM. Our results also hold in the following generalized settings (which are instrumental for later results): First, we assume the public key $\mathbf{X}$ to be split in two vector $\mathbf{X}_1, \mathbf{X}_2$, the first produced by a trusted setup (and thus no secret information about it is available to the signer), and the second one by the signer. Next, we show that if $\mathbf{X}_2$ contains n_2 group elements, and a given signature scheme allegedly supports $n_2 + \vartheta$ messages, there

exists an adversary producing ϑ valid forgeries (breaking what we call ϑ-*unforgeability*).

Techniques. Both results are fundamentally based on the strategy introduced by [53], where the adversary initially maintains a linear space L of all possible exponents the verification key $\mathbf{X}$ could have, and at each step either finds a forgery or a non-trivial relation on $\mathbf{X}$ which reduces the dimension of L by 1. Our second result however required novel techniques to overcome new problems.

The first problem we face is how to extract a set of linear equation of the form $A\mathbf{X} = B\mathbf{Y}$ from a generic verification algorithm. The natural approach is through equality queries, provided by Maurer's GGM oracles. Each query indeed can always be interpreted as a linear check $\mathbf{a}^\top\mathbf{X} = \mathbf{b}^\top\mathbf{Y}$. However, while the coefficients of the first equation are independent on $\mathbf{X}$ and $\mathbf{Y}$, subsequent queries may depend on the outcome of previous checks, eventually breaking the linearity such verification in $\mathbf{X}$ and $\mathbf{Y}$.

We solve the issue through brute-force. Specifically, we let our adversary, trying to forge m, run the verifier $\mathsf{S.Vfy}^{\mathcal{O}^0_{\mathsf{eq}},\mathcal{O}_{\mathsf{add}}}$ *guessing* the replies of $\mathcal{O}^0_{\mathsf{eq}}$, that are χ-many bits $\beta_1, \ldots, \beta_\chi$. This eventually yields two a set of linear constrains $\mathbf{a}^\top\mathbf{X} = \mathbf{b}^\top\mathbf{Y}$ that should be satisfied (those with $\beta_i = 1$), and a set of constrains that should not be (those with $\beta_i = 0$). If the resulting system is satisfiable[3] accepting and can be solved knowing $\mathbf{X}$ has exponent in L, our adversary finds a forgery. Conversely, if the above condition does not hold for the (exponentially many) possible reply choices for $\beta_1, \ldots, \beta_\chi$, our adversary queries a signature and learns a new linear relation among the elements in $\mathbf{X}$.

Our second technical challenge is that the attack above might find, in the worst case, all linear relations among the elements in $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2)$, thus requiring $n_1 + n_2$ queries (as opposed to only n_2). If $\mathbf{X}_1$, the CRS, were to be composed of random group elements we could argue that finding linear relations among them is infeasible. However such argument would fail for a possibly structured $\mathbf{X}_1$.

Our approach is to rely on a preprocessing phase through simulation. Here, the adversary simulates the whole attack described above with keys $\mathsf{sk}^*, \mathbf{X}_2^*$ generated by him (as opposed to use $\mathbf{X}_2$ provided by the challenger) several times. If the simulated adversary "fails" by finding a new non-trivial relation on $\mathbf{X}_1$, the whole simulation is restarted. Note that finding such query did cost polynomially many queries to the GGM, but required no (much more precious!) signature query. Conversely, if the simulated adversary finds no new relations sufficiently many time, our higher-level adversary runs it one last time with $\mathbf{X}_2$ provided by the challenger, and answering signature queries forwarding them to the challenger. The probability then that our subroutine finds the required forgeries without finding new relations on $\mathbf{X}_1$ (and in particular does so in at most n_2 signature queries) is then $1 - 1/\mathsf{poly}(\lambda)$.

1.3.2 Bounds for Algebraic Vector Commitments

Vector commitments (VC), introduced in [84, 31], generalize the notion of commitment scheme. They allow committing to a long list of message through a compact digest, and later open

[3] i.e. no contradictory replies where given, for instance $X = 0$ and later $X = 1$.

the vector in required positions in a publicly verifiable and succinct way.

To this day two classes of constructions exists: Tree-based ones, the most notable of which are Merkle-trees [90], allows for realizations from simple primitives such as hash functions at the cost of logarithmic opening proofs size. Alternatively, *algebraic* constructions from more structured hypothesis exist. This includes pairing [77, 84, 31], RSA and groups of unknown order [31, 19, 5] and lattices [94, 95]. Advantages of such constructions is that they achieve constant commitment and opening proofs size. Moreover these often feature extra properties e.g. (linear) homomorphism, batching/aggregation [23], sub-vector opening [19, 79]. Some of the above further supports more expressive functionalities than position opening, including the evaluations of polynomials in arbitrary points (resulting in so called *polynomial commitments*) or general inner products evaluations.

In the context of simple (known) prime order groups without pairing not many constructions are known. The current best (folklore) construction is obtained combining Pedersen commitments with inner product arguments [21, 22]. This would however come with $\Omega(\log n)$ opening proof size, and would heavily depend on hashing group elements, which breaks the algebraic structure, preventing to easily achieve the desirable extra properties mentioned above.

The second result presented in this Thesis attempts to explain the current state of affairs. We show that in Maurer's GGM, where hashing group elements (or more generally using group element's representation) is disallowed, succinct VC cannot be realized. More specifically we show tight lower bounds on the commitment and opening proof size, proving that any VC in Maurer's GGM with commitment and opening size ℓ_c and ℓ_π respectively when committing to vectors of length n, must satisfy $\ell_c \cdot \ell_\pi = \Omega(n)$. In particular either $\ell_c = \Omega(\sqrt{n})$ or $\ell_\pi = \Omega(\sqrt{n})$. Furthermore, if the VC is *hiding*, i.e. no information is ever leaked about unopened entries, a stronger bound holds. Namely, the commitment must contain $\Omega(n)$ group elements. Our results are tight in the sense that constructions achieving them exist [11].

Techniques. Our main technique is to connect (algebraic) VC to (algebraic) signatures in a way that allows us to use the previously discussed lower bounds. To do so, we show a generic compiler that turns any VC into a signature with polynomially bounded message space. The idea is simply to initially commit to a vector of n random messages $(m_1, \dots, m_n)$. Then, to sign a message i in $\{1, \dots, n\}$ we open the VC in position i.

Assuming the messages to all have high min-entropy and the VC to be hiding, it is intuitively easy to show such transformation to yield an unforgeable signature. Indeed, even after many signing queries, no information is leaked about the remaining messages. Thus forging a signature for an unopened index i either requires to guess a message with high min-entropy (information-theoretically hard), or to find a new opening in position i for the VC to a different message. The latter results in a break of *position hiding*, which for a secure VC is computationally hard.

Extending the result to non-hiding VC requires a different strategy. In this case unforgeability cannot be argued, as information on opened messages might be leaked in previous opening. However, if commitment and proofs are succinct, such leaked information cannot be too much. Informally, about $\ell_c + Q \cdot \ell_\pi$ bits of the unopened messages are revealed at most after Q queries.

If each message consists of 1 bit, forging more than $\ell_c + Q \cdot \ell_\pi + \lambda$ positions should be as hard as guessing λ random bits. Hence the scheme is ϑ-unforgeable secure with $\vartheta \approx \ell_c + Q \cdot \ell_\pi$. Precise and more general bounds are given in Theorem 4.2.1.

Finally, the attack described to break a signature scheme with verification key consisting of ν_2 group elements, and message space of size n, can produce $n - \nu_2$ forgeries in ν_2 queries (crucially, the CRS size has no impact). Here ν_2 is the number of group elements contained in the commitment (due to our VC to signature compiler), and this $\nu_2 \leq \ell_c$. As a consequence the scheme is ϑ-unforgeable only if $n - \nu_2 \leq \vartheta = \Theta(\ell_c \cdot \ell_\pi)$. Hence $\ell_c \cdot \ell_\pi = \Omega(n)$.

1.3.3 Impossibility of Algebraic NIZK

Zero Knowledge proofs and their non-interactive counterpart (NIZK) [17] are an essential tool in modern cryptography. As they allow convincing a verifier about the validity of a statement without leaking further information, they are naturally deployed to lift semi-honest secure protocols (with no active attacks) to actively secure ones (where deviations from the protocols are allowed). Due to their importance a vast literature exists about NIZK, with constructions either in the random oracle model (ROM) or in the common reference string (CRS) one.

While most NIZKs target NP-complete relations, and can thus prove any NP statement through Karp reductions, we will focus on *specialized* NIZK for group-theoretic relations such as discrete logarithm or Diffie-Hellman. In this contexts three families of constructions exist.

In the ROM, the celebrated Schnorr proofs [100], later generalized in [39, 44, 89], allow showing knowledge of a discrete logarithm (or more generally of the preimage of a linear map). Subsequent improvements eventually led to logarithmic proof length [21, 22, 6]. All such constructions are obtained compiling interactive sigma protocols (or multi-round) through the Fiat-Shamir transform [56]. Notably, security in the multi-round setting has been open for several year until proven in [7, 58, 46].

In the CRS model instead, a different line of work investigated NIZK from pairing groups [69, 70, 71] eventually leading to the popular Groth-Sahai proofs [72]. These constructions, besides removing the ROM, feature *statical soundness* and homomorphic properties. Still in the pairing setting, [40] proposed a different framework to compile some of the sigma protocols above without ROM. All these construction notably only make black-box usage of the bilinear group.

A third way rejecting both pairings and the ROM was recently put forward through correlation intractable hash (CIH). The approach is again to instantiate the Fiat-Smamir transform with standard model tools as initially proposed in [25]. The work of Jain and Jin [75], which instantiated CIH from sub-exponential DDH in pairing-free groups, eventually led to NIZK from the same assumptions. On the one hand this improved our understanding of the gap between pairing-free group and bilinear ones, proving it to be thinner than anticipated. However, the usage of CIH, as well as their realizations, requires heavy non-black box usage of the group. This is undesirable as the resulting schemes do not enjoy most of the useful properties of Groth-Sahai.

The third result discussed in this thesis explains why best of both worlds solutions are unlikely

to exist. Formally, we show that in Maurer's GGM two distinct classes of (unconditionally secure) NIZK are impossible to realize for group-related relations. First, NIZK-Argument of Knowledge, are impossible for the preimage relation $\mathcal{R} = \{(x, y) \ : \ f(x) = y\}$ for any function f that is one-way in the GGM. This includes the discrete logarithm relations. Second, NIZK are impossible for hard subset membership problems. These are languages where two efficiently sampleable distributions exist on the set of valid and invalid instances respectively, and distinguishing among the two is computationally hard. The simplest example is arguably the DDH language, but virtually any decisional assumption on groups gives rise to an hard subset membership instance.

Note the two results are incomparable. Arguments of knowledge are a subclass of NIZKs, meaning the first one cannot imply the second. On the other hand, some preimage relations, notably the discrete logarithm one, have a trivial language. Hence the second result would not hold for them and cannot in particular imply the first one.

Techniques. Different techniques are deployed for the two cases. Regarding the impossibility of NIZK-Argument of Knowledge we proceed by contradiction. If such primitive exists, we can then build hiding VC whose commitment contains $O(1)$ group elements, violating previously discussed results. Note that to reach a contradiction the opening size is not relevant.

To build intuition, let us focus here on NIZK for discrete logarithm. Given $G_1, \dots, G_n$ random CRS elements we start with a Pedersen commitment $c = m_1G_1 + \dots + m_nG_n$. In order to open m_i while hiding the other messages a first approach is to give m_jG_j, i.e. m_j in the exponent, while also providing a proof of knowledge π_j for m_j. This preserves position binding. Indeed, as π_j are proof of knowledge, an extractor could obtain the exponents of two openings and break discrete logarithm as in the security proof for Pedersen commitments. Unfortunately though this is not hiding, as m_jG_j does not hide m_j.

To circumvent such issue we use an hard-core predicate ℓ for discrete logarithm, e.g. [18, 85] (for more general OWF we will use the Goldreich-Levin predicate [66]). With it the scheme above can be adapted to commit to a vector of bits $b_1, \dots, b_n$ by first sampling m_i such that $\ell(m_i) = b_i$ and then committing to $c = m_1G_1 + \dots + m_nG_n$ as before. Opening to b_i can also be done as before: provided m_i, $H_j = m_jG_j$ and π_j proofs of knowledge for m_j for all $j \neq i$ a verifier can check $c = m_iG_i + \sum_{j \neq i} H_j$, the validity of all π_j and that $\ell(m_i) = b_i$. This eventually contradicts the results in Chapter 4 as the resulting VC is hiding, but the commitment only contains $O(1)$ group elements.

Our impossibility results for NIZK of hard subset membership languages instead follows a different path. Here we build directly on top of the adversary breaking signatures in the GGM described in Chapter 3. To start with a simpler case, assume the NIZK to be *simulation-sound*, meaning that producing new proofs for false statement is hard even when access to a NIZK simulator oracle is provided. This can be compiled to a signature scheme with polynomially bounded message space. To do so, the signature CRS consists of n false statements $x_1, \dots, x_n$. The verification key vk is the NIZK crs in simulation mode, and a signature for message i is a (simulated) proof for x_i. Unforgeability crucially depends on simulation soundness, which is broken by the attack in Chapter 3 when $n > |\mathsf{crs}|$.

The same strategy for plain NIZK however fails. Indeed simulating even one statement might grant the ability to prove *any* false one. To handle such constrained setting we leverage the power of hard subset membership problems. Our reduction $\mathcal{R}$, playing against a NIZK with honest setup crs, will each round toss a coin and simulate the signature adversary $\mathcal{A}$ in one out of two worlds:

1. In the first, it samples a true statement x and witness w, setting x as the CRS of a 1-message signature. If $\mathcal{A}$ queries a signature for the only message x, $\mathcal{R}$ honestly computes π using its witness and extracts from $\mathcal{A}$ a linear combination for the verification key elements (i.e. its crs).
2. In the second, it samples a false statement x, and sets it as before as the CRS of a 1-message signature. If the adversary $\mathcal{A}$ returns a forgery, this will be a proof π for x, which breaks the NIZK soundness.

As the two worlds are indistinguishable, as distinguishing true and false statement is hard, $\mathcal{A}$ has no information on the world it is executed in. With sufficiently many repetitions, either a forgery or a linear relation over the elements of crs can be then extracted. Thus $\mathcal{R}$ eventually breaks soundness.

1.3.4 Bounds for Distributed Group Action Computation

Hard homogeneous spaces (HHS) [41] represents one of the few sources of supposedly *post-quantum* problems along with lattices and symmetric key primitives. An HHS is informally defined by a group action $\star : \mathbb{G} \times \mathcal{E} \to \mathcal{E}$ (with $(\mathbb{G}, +)$ being a group) which is hard to invert. A simple example of group action, to build intuition, is the scalar multiplication map $\cdot : \mathbb{F}_q \times \mathbb{G} \to \mathbb{G}$ mapping $(a, G) \mapsto a \cdot G$. However, while in this example the set also features a group structure, in general elements in $\mathcal{E}$ cannot be added in a way that respects the action.

Such limited structure is the main strength of HHS. Indeed quantum algorithms to break discrete logarithm [102, 97] crucially rely on the periodic group structure. Note that current instantiations of HHS are based on super singular isogenies, in particular from CSIDH. While a recent line of work broke the (classical) hardness of SIDH [29, 86, 98], CSIDH was not affected.

Compared to symmetric key primitives such as hash functions, the (limited) homomorphic properties have proven useful to build a variety of cryptographic primitives. Notably PKE [105, 91], signatures [47, 16, 55, 2], identification schemes [9], identity-based signatures [101, 1], adaptor signatures [106], verifiable random functions [80], oblivious transfer [81], ring signatures [15], group signatures [14], and most importantly threshold schemes [48, 43, 13, 24, 2, 3, 4].

However, compared to lattices or prime order groups, HHS are significantly less flexible, especially for distributed protocols. A common problem that arises in the context of threshold signatures or distributed key generation is to compute the action $s \star E$ for a public E and a secretly shared s. Over groups this is easily solved: if the sharing is additive, i.e. each party has s_i so that $s = s_1 + \ldots + s_n$, then parties broadcast $s_i \cdot G$ and locally multiply these group elements to get $s \cdot G$. Group actions however do not admit such efficient aggregation. To the best of our knowledge, only two solutions are known to date.

A first one relies on so called *round-robin protocols*. The main idea is that parties are linearly ordered, and then operate sequentially, i.e. at round i, the i-th party performs some computation and return its output to the $(i+1)$-th user. Concretely, reconstructing $s \star E$ when s is additively shared is performed computing in the first round $E_1 = s_1 \star E$, in the second $E_2 = s_2 \star E_1$ and so on until $E_n = s_n \star E_{n-1}$. Assuming semi-honest behavior, eventually $E_n = s \star E$. This results in a computation and communication efficient protocol, using the action in a black-box way, but its round complexity scales linearly in the number of users, severely affecting scalability.

A second, always viable, approach is to rely on generic multi-party computation (MPC). While this effectively solves the $\Omega(n)$ round complexity, it also introduces new issues. In particular, generic MPC techniques requires an explicit description of the circuit evaluating the action in order to compute it gate-by-gate. However, currently known circuits for HHS ([30]) are not quite MPC-friendly. As high circuit size/depth affect either round or communication complexity, this results in currently impractical protocols.

The forth result discussed in this thesis aims at arguing that such dichotomy of approaches is inherent. More specifically, we show that any protocol computing $s \star E$ (or, more generally a function of s acting on E), with a t out of n secretly shared[4] value s, either

- Depends on the group action circuit (formally, cannot be instantiated in the generic action model).
- Has round complexity $\Omega(t)$

Remarkably, the attack we provide when both conditions do not hold is efficient: hence relying on external assumptions is not enough to bypass our result as opposed to those presented in Chapters 3-5.

Moreover, we show tight efficiency bound for so called *fair round-robin* protocols, where all users receive their output in the last round (even if a single corruption occurs). An example of such protocol was proposed in [48] with communication and computation complexity $O(n \log_2 n)$. We prove this to be optimal when aiming for optimal round complexity when n is a power of 2.

Techniques. The techniques we develop and refine to get our results are all graph-theoretic. Following [20], the idea is to associate to any generic group action computation a directed graph, where vertices are the observed set elements, and an edge between E_1 and E_2 exists if $\mathcal{O}_{\mathsf{act}}(a, E_1) = E_2$ was queried[5]. A useful observation is that, given such (labeled) graph, if D and E are connected it is possible to recover a such that $\mathcal{O}_{\mathsf{act}}(a, D) = E$.

For the sake of generality, we prove our results in (a relaxation of) the *dense* Generic Action Model of [54], where oblivious sampling of set element is efficient. A first step is then to show oblivious sampling to be useless in reconstruction protocols. Specifically we show that, for any $D \in \mathcal{E}$ and machine $\mathcal{A}(D)$ returning (a, E) such that $E = a \star D$, then D and E are connected in the associated graph. We call this the *sequentiality lemma*, as it shows that $a \star D$ can only

[4] i.e. so that $t+1$ parties can recover s, but any set of t users cannot jointly recover any information on it.
[5] to avoid cycles we also require E_1 to be observed *before* E_2

be computed through sequential applications of the action.

To obtain our first result we refine the lemma above in the interactive setting. Precisely, fix a k round protocol computing $E_{\mathsf{out}} = s \star E_0$ given E_0. Then among all paths connecting E_0 to E_{out} there exists one whose associated queries[6] are such that no two were performed in the same round by two distinct players. This implies that the path involves queries of at most k users, and in particular those k users involved can jointly recover the path and so the secret s.

Regarding our second result, techniques are more involved. To recap, our hypothesis are that, for t-out-of-n shared secret s, the protocol takes $t+1$ rounds (for simplicity let us assume $t = n-1$) and that each user outputs $E^i_{\mathsf{out}} = s \star E_0$ in the last round. A first approach to show efficiency bounds could be to consider all paths connecting E_0 to E^i_{out}. Indeed, taking a single path, it is easy to show this must involve $\Omega(n)$ communication and computation, or else some user along the path could have been skipped, leading to an attack as before. Considering multiple paths however is harder, as those may share a (large) fraction of associated queries.

We address such issue with the following abstraction: to each path we associate the sequence of indices $\pi(1), \ldots, \pi(n)$ of users involved in its computation. Since each user must appear at least once (or else $n-1$ users were involved, leading to attacks), π are permutations over $\{1, \ldots, n\}$. A key technical observation is then that two paths (with associated permutations π_i, π_j) can share the same queries/set elements at round r *only if* π_i, π_j share the same prefix, i.e. $\pi_i(1) = \pi_j(1), \ldots, \pi_i(r) = \pi_j(r)$.

Such observation allows us to link communication and computation costs to the number of edges in the *prefix tree* of $\pi_1, \ldots, \pi_n$. We can then conclude our argument in a purely graph-theoretic way. First, what we call the *tall sub-tree property* (TSP) is introduced. Informally, a tree is *tall* if all leaves have the same distance from the root and the height is higher than the number of leaves. A tree has the TSP if all its sub-trees are tall. Leveraging the fact that $\pi_1, \ldots, \pi_n$ are permutations, it is easy to show that their prefix tree satisfies the TSP. Finally the argument is completed showing by induction that a tree with the TSP and n leaves has $\Omega(n \log n)$ edges. This concludes the proof of our bound on fair protocols.

1.3.5 Bounds for Black-Box Anamorphic Encryption

Anamorphic Encryption (AE) [96] is a novel paradigm to provide private communication against a dictator exercising significant control over users. This includes observing user's secret key (violating *receiver privacy*) and influencing sent messages (violating *sender freedom*). Such level of control is indeed plausible in contexts where strong censorship measures are in place.

To address this, [96] put forward the notion of Anamorphic Encryption. In the so called *receiver* AE setting, a given public key encryption scheme can be deployed in one out of two modes. In regular mode keys and ciphertexts are computed honestly according to the provided PKE. In anamorphic mode, an extra key dk is generated and shared between sender and receiver. This allows to encrypt both a regular message m, recoverable with the scheme's secret key, along with a *covert message* $\widehat{m}$ which can only be retrieved using dk. An AE

[6] i.e. those queries to $\mathcal{O}_{\mathsf{act}}$ that allowed us to place the edges forming such path.

scheme is secure if the two modes are indistinguishable for a dictator holding both pk and sk.

Over the last years several constructions have been proposed [96, 78, 10, 107, 35]. Most of them exploit specific properties of the underlying PKE. Notable exceptions are the rejection sampling scheme in [96], as well as the first robust construction in [10], both generic in the underlying PKE. The flexibility of being able to apply those construction to any PKE is a desirable property in contexts where the PKE is chosen by the dictator. In such case indeed, encryption schemes admitting efficient *ad hoc* anamorphism may simply be banned.

Currently, however, the price to pay for generic AE constructions is to only achieve low communication rate. Indeed, both solutions mentioned above support anamorphic message space of size at most polynomial in the security parameter. On the contrary most specialized constructions do achieve exponential message space size. Another limitation is that stronger notions of security were not known to be achievable through generic AE. This notably includes *Asymmetric AE*, where in anamorphic mode two extra keys dk and tk are generated, acting respectively as public encryption and private decryption key for anamorphic messages.

The fifth and final result presented in this thesis shows such limitations are inherent for the class of black-box constructions. More in detail, we focus on AE schemes that only access the underlying PKE through oracle calls – where security holds as long as the PKE is correct and IND-CPA. For this class of schemes, assuming correctness *on average* (i.e. that decryption error is negligible averaging on $m, \widehat{m}$), we prove:

- The anamorphic message space must be bounded by $|\widehat{M}| \leq \mathsf{poly}(\lambda)$.
- (A weakening of) Asymmetric AE is impossible, even with small message space.

As per other results targeting black-box constructions, the above limitations above could be bypassed via non black-box techniques. However, subsequent work [37] showed that common tools such as NIZK [17], Garbling [109] and indistinguishability obfuscation [**?**] do not suffice. Dodis and Goldin [50] recently described a PKE in the ROM with backdoored public parameters such that any AE for it must match our bound. Yet, whether *for every* PKE (in the standard model) there exists a secure AE[7] extending it with super-polynomial anamorphic message space size, or not, remains open.

Techniques. The starting point of both results is a characterization of how AE black-box PKE. We call it the *ciphertext selection lemma*. Roughly it states that $\mathsf{AT.Enc}$, the anamorphic encryption procedure, when encrypting a regular m and anamorphic $\widehat{m}$, must return a ciphertext c obtained as $\mathsf{E.Enc}(\mathsf{pk}, m)$. This is shown instantiating the PKE via truly random functions (with carefully chosen parameters), referred to in the following as an *ideal PKE*. In such case a value c not previously observed to be a valid encryption of m will result in a decryption error with overwhelming probability, which the dictator can detect.

Given this powerful lemma, we prove the first bound on the message space as follows. First, an information-theoretical game called a *Random Oracle Channel* is defined where two unbounded players attempt to communicate a message under some restrictions (more on this later). Next, through a compression argument, we bound the message space size to be at most $\mathsf{poly}(\lambda)$.

[7] at least *semi-adaptively* secure as defined in [37].

Finally, we compile any AE into a ROC where the anamorphic message $\widehat{m}$ is exchanged, meaning that $|\widehat{M}| \leq \mathsf{poly}(\lambda)$.

More in details, two players in a RO-channel abide to the following rules: both have access to a common random oracle H, can make at most $\mathsf{poly}(\lambda)$ many queries to H, the sender $\mathcal{S}^{\mathsf{H}}(m) \to y$ can only return one of the (polynomially many) RO values it observed, while the receiver $\mathcal{R}^{\mathsf{H}}(y)$ has to guess m. Constructing a RO-channel from black-box AE is then achieved by replacing the ideal PKE's encryption procedure with $\mathsf{H}(\mathsf{pk}, m, r)$. $\mathsf{AT.Enc}$ on input $(m, \widehat{m})$ is then a valid sender as, by the ciphertext selection lemma, it only returns a value it obtained from H. Correctness further implies that $\mathsf{AT.Dec}$ is a valid receiver, as it successfully recovers $\widehat{m}$.

Our second result instead proves the impossibility of building black-box asymmetric AE. The added security requirement in this setting is that on input dk it is possible to produce anamorphic ciphertexts for $(m, \widehat{m})$, but not to decrypt them. We will however show that if such a scheme exists, then $\mathsf{AT.Enc}$ (which only requires dk) can be exploited to decrypt ciphertexts. The main idea is that, by the ciphertext selection lemma, $\mathsf{AT.Enc}$ encrypts $(m, \widehat{m})$ by choosing one of those ciphertexts obtained through queries to $\mathsf{E.Enc}(\mathsf{pk}, m)$. In particular, given a list of ciphertexts $\mathsf{AT.Enc}$ must be able to understand which one encrypts $\widehat{m}$.

Formally the attack works as follows: Our adversary initially obtains a challenge c^* encrypting either $(m, \widehat{m}_0)$ or $(m, \widehat{m}_1)$. Then it locally runs $\mathsf{AT.Enc}$ on input $(m, \widehat{m}_0)$ and replace the reply of a randomly chosen query to $\mathsf{E.Enc}(\mathsf{pk}, m)$ with c^*. Ideally, if $\mathsf{AT.Enc}$ cannot distinguish c^* from fresh ciphertexts, correctness guarantees that c^* is returned with probability $\geq 1/\mathsf{poly}(\lambda)$ if c^* encrypts $\widehat{m}_0$, and negligible otherwise.

Unfortunately however, $\mathsf{AT.Enc}$ *can* easily distinguish c^* from fresh ciphertexts. This is the case as c^* was obtained through $\mathsf{AT.Enc}$. While on the one hand we know c^* is chosen from a set of correctly distributed ciphertexts, on the other hand such choice introduces a (possibly strong) bias[8] in c^*. To save our argument we show that, although c^* is distinguishable from the other $c_1, \ldots, c_q$ observed by $\mathsf{AT.Enc}$, when it encrypts the right message it is still chosen with probability $\approx 1/q$. This unexpected result follows by a more general observation on what we call *symmetric choice function*.

A symmetric choice function is any f which returns one of its arguments and does not depend on the argument's order. Our key observation is that such functions are *consistent* in their choices. More precisely, given freshly sampled $u_1, \ldots, u_n, v_2, \ldots, v_n$, let $v_1 = f(u_1, \ldots, u_n)$. Then this element is chosen again, i.e. $f(v_1, v_2, \ldots, v_n) = v_1$, with probability $\approx 1/n$. Carefully applying this elementary lemma (as $\mathsf{AT.Enc}$ behaves as a symmetric choice functions acting on the ciphertext it gets from $\mathsf{E.Enc}$), the attack above can eventually be shown to succeed with significant probability.

[8] In the same way $b_1 \wedge \ldots \wedge b_n$ is not uniformly distributed for freshly sampled bits $b_1, \ldots, b_n$

Chapter 2

Background

2.1 Hard Problems and Idealized Models

2.1.1 Prime Order Groups

A significant portion of modern public-key cryptography is based on hard problems defined over a prime order group. In this setting we assume the representation of a group $(\mathbb{G}, +)$ along with a generator $G \in \mathbb{G}$ is provided, where $\mathbb{G}$ has prime order q. Moreover, membership in $\mathbb{G}$, sampling from $\mathbb{G}$, additions and inversion should all be efficiently computable. Instantiations of such includes multiplicative subgroups of $\mathbb{Z}_p^*$ or the group structure of an elliptic curve over a finite field.

The most studied problems over such groups includes the discrete logarithm problem (DLP), stating that scalar multiplication is a one-way function, and computational (or decisional) Diffie-Hellman assumption [49]. These are formally defined as follows.

Definition 2.1.1 (DLP)**.** *For any* PPT *adversary* $\mathcal{A}$ *there exists a negligible* ε *such that given* $G \in \mathbb{G}$ *and* $x \xleftarrow{\$} \mathbb{F}_q$

$$\mathsf{Adv}(\mathcal{A}) \coloneqq \Pr\left[\mathcal{A}(G, xG) \to x\right] \leq \varepsilon(\lambda).$$

Definition 2.1.2 (CDH)**.** *For any* PPT *adversary* $\mathcal{A}$ *there exists a negligible* ε *such that given* $G \in \mathbb{G}$ *and* $a, b \xleftarrow{\$} \mathbb{F}_q$

$$\mathsf{Adv}(\mathcal{A}) \coloneqq \Pr\left[\mathcal{A}(G, aG, bG) \to ab \cdot G\right] \leq \varepsilon(\lambda).$$

Definition 2.1.3 (DDH)**.** *For any* PPT *adversary* $\mathcal{A}$ *there exists a negligible* ε *such that, given* $G \in \mathbb{G}$ *and* $a, b, c \xleftarrow{\$} \mathbb{F}_q$

$$\mathsf{Adv}(\mathcal{A}) \coloneqq \left|\Pr\left[\mathcal{A}(G, aG, bG, abG) \to 1\right] - \Pr\left[\mathcal{A}(G, aG, bG, cG) \to 1\right]\right| \leq \varepsilon(\lambda).$$

2.1.2 Generic Group Models

In spite of its simplicity, analyzing in general the hardness of group-theoretic problem such as the DLP or CHD is hard, as potential attacks may heavily rely on the group representation.

This is for instance the case of number-field sieve attacks, also affecting subgroups of $\mathbb{Z}_p^*$. This raised the question of what is the best security one could hope from a general, if no specialized attack can be carried out. The Generic Group Model (GGM) was then introduced in [103] and [88] to address such question. In this section we recall the two models (respectively called Shoup's GGM and Maurer's GGM)and their differences.

Shoup's GGM. The model proposed by Victor Shoup removed the help provided by a given group representation through a truly random labeling injective function $\sigma : \mathbb{F}_q \to \{0,1\}^\ell$ for some $\ell \geq \log_2 q$. A machine in Shoup's GGM has then access to two oracles: $\mathcal{O}_{\mathsf{gen}}, \mathcal{O}_{\mathsf{add}}$. The first one always returns $\sigma(0)$, i.e. the canonical generator, while $\mathcal{O}_{\mathsf{add}}(\sigma(n), \sigma(m))$ returns $\sigma(n+m)$. Note that inversion and scalar multiplication can be efficiently computed through repeated queries to $\mathcal{O}_{\mathsf{add}}$. Membership instead can be either tested though an oracle returning whether $X \in \sigma(\mathbb{F}_q)$ or by making $\mathcal{O}_{\mathsf{add}}$ return $\perp$ if one of its entries is not a valid label.

The length of ℓ plays an important role, easily overlooked, in this model. Indeed if $\ell = \log_2 q + O(\log \lambda)$ sampling random group elements of unknown discrete logarithm, or hashing into the group, is feasible. Conversely, setting $\ell = \log_2 q + \omega(\log \lambda)$ up to negligible probability all computed group elements can be *explained* as a linear combination of previously observed ones (as in the Algebraic Group Model [57]).

Maurer's GGM. We now discuss Maurer's Generic Group Model [88], later revised by Maurer, Portmann and Zhu [87]. Although the original definition was given in higher generality, below we formally present it for the specific case of prime order groups.

Definition 2.1.4. *Maurer's Generic Group of prime order q is an interactive stateful Turing machine $\mathcal{B}$ which at any step stores a list of elements $L = (V_1, \dots, V_m)$ with $V_i \in \mathbb{F}_q$. Initially L is empty and $m = 0$. Furthermore, upon receiving:*

- (gen)*: Sets $V_{m+1} \leftarrow 1$, appends V_{m+1} to L and increments $m \leftarrow m+1$.*
- (add, i, j)*: if $i, j \in [m]$, sets $V_{m+1} \leftarrow V_i + V_j$, appends V_{m+1} to L and sets $m \leftarrow m+1$.*
- (eq, i, j)*: if $i, j \in [m]$, computes b as the bit $V_i == V_j$ and returns b.*

A remarkable difference between this model and Shoup's one is that a machine interacting with $\mathcal{B}$ has no access to a representation of group elements it computed, and has to instead *remember* their indices as it queries them. In particular, computation that depends on the representation[1] is not allowed in this model. For such reason, any machine in Maurer's GGM can be compiled into a functionally equivalent one in Shoup's GGM, but the opposite is not true.

To keep notation simpler we may assume without loss of generality that after (gen) or (add, i, j), $\mathcal{B}$ returns $m+1$, the newly generated element's index. Furthermore, for notational simplicity, we may describe usage of the generic group through three oracles $\mathcal{O}_{\mathsf{gen}}$, $\mathcal{O}_{\mathsf{add}}$, $\mathcal{O}_{\mathsf{eq}}$ such that

- $\mathcal{O}_{\mathsf{gen}}()$ queries (gen) to $\mathcal{B}$ and return the new group element's index G.
- $\mathcal{O}_{\mathsf{add}}(X, Y)$, queries (add, X, Y) to $\mathcal{B}$ and return the new group element's index Z

[1] e.g. hashing/mapping a group element to scalar.

- $\mathcal{O}_{\mathsf{eq}}(X, Y)$, queries (eq, X, Y) to $\mathcal{B}$ and return the bit b it receives back from $\mathcal{B}$.

Noticeably, without loss of generality, one may assume that $\mathcal{O}_{\mathsf{gen}}$ is queried only once at the beginning of any algorithm's execution – or equivalently that the list L maintained by $\mathcal{B}$ is initialized with 1 in its first entry. Similarly one may also assume that, up to performing at most $2 \log q$ queries to $\mathcal{O}_{\mathsf{add}}$, an index for the group identity 0 is known - which could be obtained by computing $q \cdot G$. Analogously, given the index of an element X, its inverse can be computed in at most $2 \log q$ queries to $\mathcal{O}_{\mathsf{add}}$ by computing $(q-1)X$. Finally we observe that, since inverses can be computed efficiently, for algorithms that are bounded to perform at most a polynomial number of queries to $\mathcal{B}$, it is enough to provide access to the identity equality test oracle $\mathcal{O}^0_{\mathsf{eq}}$ defined as

- $\mathcal{O}^0_{\mathsf{eq}}(X)$: Given a precomputed index for 0, it queries $(\mathsf{eq}, X, 0)$ to $\mathcal{B}$ and returns the bit b it receives back from $\mathcal{B}$.

Given access to this oracle $\mathcal{O}_{\mathsf{eq}}(X, Y)$ can then be simulated querying $\mathcal{O}^0_{\mathsf{eq}}(X - Y)$.

We finally conclude this section explicitly defining an important class of adversaries often used to prove information-theoretic results in the GGM.

Definition 2.1.5. GPPT *is the class of all (unbounded) probabilistic Turing Machines with access to* $\mathcal{O}_{\mathsf{add}}, \mathcal{O}_{\mathsf{eq}}$ *whose number of oracle queries is polynomially bounded in their input length.*

In Chapters 3, 4 and 5, we often consider adversaries from this large class. We do so to limit the source of hardness to group-theoretic problems.

2.1.3 Hard Homogeneous Spaces

The main long-term threat to cryptography based on prime order groups is posed by the feasibility of breaking the DLP in quantum polynomial time [102]. For this reason new problems, plausibly post-quantum, have been studied. In this section we recall the notion of Hard Homogeneous Space [41]. These, along lattices, represent one the few sources of problems believed to be quantum resistant.

First, we start with the mathematical definition of group action.

Definition 2.1.6. *An action of a finite group* $(\mathbb{G}, +)$ *on a set* $\mathcal{E}$ *is given by a map* $\star : \mathbb{G} \times \mathcal{E} \to \mathcal{E}$ *satisfying the following properties:*

1. *Identity:* $0 \star E = E$ *for all* $E \in \mathcal{E}$.
2. *Associativity:* $(h + g) \star E = h \star (g \star E)$ *for every* $h, g \in \mathbb{G}$ *and* $E \in \mathcal{E}$.

The action is transitive if for all $E_1, E_2 \in \mathcal{E}$ *there exists* $g \in \mathbb{G}$ *such that* $E_2 = g \star E_1$. *If* g *is unique the action is called free.*

For a group action to be *effectively computable* we further assume that deciding membership in $\mathbb{G}$, $\mathcal{E}$ and equality between elements in $\mathbb{G}$ and $\mathcal{E}$ is efficient. Moreover sampling uniformly from both sets, computing the operations and inverses in $\mathbb{G}$, and the group action $\star$ is also efficient.

A set $\mathcal{E}$ equipped with an effective group action $\star$ is called *Homogeneous Space* if $\star$ is transitive and free. In the rest of this thesis however we will extend this notion to actions that are not necessarily free. This, in line with the approach of [20], allows for instance encoding quadratic twists for CSIDH as the action of specific group elements.

An homogeneous space is finally called *hard* if the vectorization and parallelization problems, defined below, are hard. Those can be seen as the generalization of the DLP and CDH to the group action case.

Definition 2.1.7. *The Vectorization problem is hard for a given homogeneous space if for any* PPT *adversary* $\mathcal{A}$ *there exists a negligible* ε *such that, setting* $E \in \mathcal{E}$ *and* $a \leftarrow^{\$} \mathbb{G}$

$$\mathsf{Adv}(\mathcal{A}) \;=\; \Pr\left[\mathcal{A}(E, a \star E) \to a\right] \;\leq\; \varepsilon(\lambda).$$

Definition 2.1.8. *The Parallelization problem is hard for a given homogeneous space if, for any* PPT *adversary* $\mathcal{A}$ *there exists a negligible* ε *such that, setting* $E \in \mathcal{E}$ *and* $a, b \leftarrow^{\$} \mathbb{G}$

$$\mathsf{Adv}(\mathcal{A}) \;=\; \Pr\left[\mathcal{A}(E, a \star E, b \star E) \to (a+b) \star E\right] \;\leq\; \varepsilon(\lambda).$$

2.1.4 Generic Action Model

In complete analogy to the GGM it is possible to define idealized model capturing generic use of an HHS. Although both Shoup's and Maurer's model could be extended to this setting, we only present here a generalization of Shoup's GGM, which will be called the Generic Action Model (GAM) throughout this thesis.

In the GAM, the group action $\star : \mathbb{G} \times \mathcal{E} \to \mathcal{E}$ is modeled through an oracle $\mathcal{O}_{\mathsf{act}}$. Initially a random injective labeling function $\sigma : \mathcal{E} \to \{0,1\}^{\mu}$ is sampled and users receive $E_0 = \sigma(E'_0)$ the encoding of an element in $\mathcal{E}$. Action queries are then replied to with

$$\mathcal{O}_{\mathsf{act}}(a, E) \;=\; \begin{cases} \sigma(a \star E') & \text{If } E = \sigma(E') \\ \perp & \text{If } E \notin \operatorname{Im} \sigma \end{cases}.$$

We do not provide an oracle to test membership in $\sigma(\mathcal{E})$ as this can be checked querying $\mathcal{O}_{\mathsf{act}}(0, E)$, which returns E if $E \in \sigma(\mathcal{E})$ and $\perp$ if $E \notin \sigma(\mathcal{E})$. As in the case of Shoup's GGM, if $\mu = \log_2 |\mathcal{E}| + O(\log \lambda)$ the model allows sampling random elements of unknown "discrete logarithm" in base E_0. Conversely if $\mu = \log_2 |\mathcal{E}| + \omega(\log \lambda)$ sampling random elements is computationally hard.

In relation to previously proposed models [54, 20] for generic group actions, ours allows parties to have an explicit representation for set elements as done in [54] and as opposed to [20]. However, as in [20] we allow non-free and non-commutative group actions in order to encode external operations such as quadratic twists as particular action evaluations. In this sense we do not need a separate oracle to capture twists as done in [54].

2.2 Cryptographic Primitives

2.2.1 Public Key Encryption and Signatures

We recall the notation and syntax for public key encryption and signature schemes.

Definition 2.2.1. *A Public Key Encryption scheme (PKE) is a tuple of* PPT *algorithms* $(\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ *and a message space set* M *such that*

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(1^\lambda)$ *generates the public encryption and secret decryption keys.*
- $c \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{pk}, m)$ *returns an encryption of* m
- $m \leftarrow \mathsf{E.Dec}(\mathsf{sk}, c)$ *returns a decryption of* c

A public key encryption scheme is *perfectly correct* if for any $(\mathsf{pk}, \mathsf{sk})$ in the support of $\mathsf{E.Gen}$, and message m, setting $c \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{pk}, m)$ then $\Pr[m \leftarrow \mathsf{E.Dec}(\mathsf{sk}, c)] = 1$. The standard security notion for PKE schemes is IND-CPA.

Definition 2.2.2. *A signature scheme is a tuple of* PPT *algorithms* $(\mathsf{S.Setup}, \mathsf{S.Sign}, \mathsf{S.Vfy})$ *and a message space set* $\mathsf{S.M}$ *such that*

- $\mathsf{S.Setup}(1^\lambda) \xrightarrow{\$} (\mathsf{sk}, \mathsf{vk})$ *generates the secret and verification keys*
- $\mathsf{S.Sign}(\mathsf{sk}, m) \xrightarrow{\$} \sigma$ *returns the signature of a message* $m \in \mathsf{S.M}$
- $\mathsf{S.Vfy}(\mathsf{vk}, m, \sigma) \to 0/1$ *verifies the signature* σ *for a message* $m \in \mathsf{S.M}$

We further require a signature scheme to satisfy *perfect correctness*, meaning that if $(\mathsf{sk}, \mathsf{vk}) \leftarrow^{\$} \mathsf{S.Setup}(1^\lambda)$ and $\sigma \leftarrow^{\$} \mathsf{S.Sign}(\mathsf{sk}, m)$ for any $m \in \mathsf{S.M}$ then the verification algorithm accepts always, i.e.

$$\Pr[\mathsf{S.Vfy}(\mathsf{vk}, m, \sigma) \to 1] = 1.$$

2.2.2 Vector Commitments

We recall the definition of vector commitments from [31].

Definition 2.2.3 (VC)**.** *A Vector Commitment scheme is a tuple of algorithms* $(\mathsf{VC.Setup}, \mathsf{VC.Com}, \mathsf{VC.Open}, \mathsf{VC.Vfy})$ *and a message space* $\mathsf{VC.M}$ *such that*

- $\mathsf{VC.Setup}(1^\lambda) \xrightarrow{\$} \mathsf{pp}$ *generates the public parameters.*
- $\mathsf{VC.Com}(\mathsf{pp}, m_1, \ldots, m_n) \xrightarrow{\$} c, \mathsf{aux}$ *produce a commitment to* $m_1, \ldots, m_n \in \mathsf{VC.M}$ *together with some auxiliary information.*
- $\mathsf{VC.Open}(\mathsf{pp}, m, i, \mathsf{aux}) \xrightarrow{\$} \pi$ *return an opening proof that the* i*-th entry of a given commitment is* m_i*.*
- $\mathsf{VC.Vfy}(\mathsf{pp}, c, m, i, \pi) \to 0/1$ *verifies the opening proof's correctness.*

We require a vector commitment scheme to satisfy *perfect correctness*, that is, given public parameters $\mathsf{pp} \leftarrow^{\$} \mathsf{VC.Setup}(1^\lambda)$, commitment $c, \mathsf{aux} \leftarrow^{\$} \mathsf{VC.Com}(\mathsf{pp}, m_1, \ldots, m_n)$ for any

$m_i \in \mathsf{VC.M}$, and opening $\pi \leftarrow^{\$} \mathsf{VC.Open}(\mathsf{pp}, m_i, i, \mathsf{aux})$, it holds

$$\Pr\left[\mathsf{VC.Vfy}(\mathsf{pp}, c, m, i, \pi) \to 1\right] = 1$$

Moreover, to avoid trivial cases, in this thesis we assume $|\mathsf{VC.M}| \geq 2$.

The main security property for a vector commitments is the so called *position binding*, which informally states that no adversary can open the same position of a given commitment to two different values. Formally

Definition 2.2.4 (Position binding)**.** *A vector commitment scheme satisfies position binding if for any* PPT *adversary* $\mathcal{A}$ *there exists a negligible function* $\varepsilon(\lambda)$ *such that*

$$\Pr\left[\begin{array}{l}\mathsf{VC.Vfy}(\mathsf{pp}, c, m, i, \pi) \to 1 \\ \mathsf{VC.Vfy}(\mathsf{pp}, c, m', i, \pi') \to 1 \\ m \neq m'\end{array} \middle| \begin{array}{l}\mathsf{pp} \leftarrow^{\$} \mathsf{VC.Setup}(1^\lambda) \\ \mathcal{A}(\mathsf{pp}) \to (c, m, m', i, \pi, \pi')\end{array}\right] \leq \varepsilon(\lambda).$$

The property that distinguishes VCs from classical binding commitments is *succinctness* Following [84, 31], a VC scheme is said succinct if there is a fixed $p(\lambda) = \mathsf{poly}(\lambda)$ such that for any n the size of honestly generated commitments and openings is bounded by $p(\lambda)$. One may also consider weaker notions where the size may be bounded by $p(\lambda)\log n$ or $p(\lambda, \log n)$.

Since in this thesis we are interested in understanding the feasibility of VCs based on their level of succinctness, we consider a parametric notion. We say that a VC has succinctness (ℓ_c, ℓ_π) if for any $m_1, \ldots, m_n \in \mathsf{VC.M}$, commitment $c, \mathsf{aux} \leftarrow^{\$} \mathsf{VC.Com}(\mathsf{pp}, m_1, \ldots, m_n)$ and opening $\pi \leftarrow^{\$} \mathsf{VC.Open}(\mathsf{pp}, m_i, i, \mathsf{aux})$ for any $i \in [n]$, we have that c (resp. π) has bit-length $\ell_c(\lambda, n)$ (resp. $\ell_\pi(\lambda, n)$).

2.2.3 Non-Interactive Zero-Knowledge Arguments

A Non-Interactive Zero-Knowledge argument (NIZK) for a relation $\mathcal{R}$ is a tuple of three algorithms $(\mathsf{G}, \mathsf{P}, \mathsf{V})$ that allow a prover to convince a verifier about the validity of a statement without leaking any other information. Given $\mathsf{crs} \leftarrow \mathsf{G}(1^\lambda)$ and $(x, w) \in \mathcal{R}$ a valid statement, the prover can compute a proof running $\pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w)$ which can later be verified by $b \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi)$. The proof is accepted if $b = 1$, or rejected otherwise. Below we revise formally the main properties NIZKs can satify.

- **Completeness**: $\forall (x, w) \in \mathcal{R}$

$$\Pr\left[1 \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi) \,\middle|\, \mathsf{crs} \leftarrow \mathsf{G}(1^\lambda),\ \pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w)\right] \ = \ 1.$$

- **Soundness**: $\exists \varepsilon$ negligible such that $\forall x \ : \nexists w \ : \ (x, w) \in \mathcal{R}$ and $\forall \mathcal{A}$ PPT

$$\Pr\left[1 \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi) \,\middle|\, \mathsf{crs} \leftarrow \mathsf{G}(1^\lambda),\ \pi \leftarrow \mathcal{A}(\mathsf{crs}, x)\right] \ \leq \ \varepsilon(\lambda).$$

- **Argument of Knowledge**: For any PPT adversary $\mathcal{A}$ there exists a PPT extractor E such that

1. $\exists \varepsilon$ negligible such that $\forall \mathcal{D}$ PPT, given $\mathsf{crs}_0, \mathsf{td} \leftarrow \mathsf{E}(1^\lambda),\ \mathsf{crs}_1 \leftarrow \mathsf{G}(1^\lambda)$

$$|\Pr[1 \leftarrow \mathcal{D}(\mathsf{crs}_0)] - \Pr[1 \leftarrow \mathcal{D}(\mathsf{crs}_1)]| \leq \varepsilon(\lambda).$$

2. There exists a negligible function ε such that

$$\Pr\left[\begin{matrix}\mathsf{V}(\mathsf{crs}, x, \pi) \to 1 \\ (x, w) \notin \mathcal{R}\end{matrix} \;\middle|\; \begin{matrix}\mathsf{crs}, \mathsf{td} \leftarrow \mathsf{E}(1^\lambda),\ (x, \pi) \leftarrow \mathcal{A}(\mathsf{crs}) \\ w \leftarrow \mathsf{E}(\mathsf{td}, x, \pi)\end{matrix}\right] \leq \varepsilon(\lambda).$$

- **Zero-Knowledge**: There exists a PPT simulator S such that, up to negligible probability ε, for all $(x, w) \in \mathcal{R}$ and PPT adversary $\mathcal{A}$, given

$$\mathsf{crs}_0, \mathsf{td} \leftarrow \mathsf{S}(1^\lambda), \quad \pi_0 \leftarrow \mathsf{S}(\mathsf{td}, x), \quad \mathsf{crs}_1 \leftarrow \mathsf{G}(1^\lambda), \quad \pi_1 \leftarrow \mathsf{P}(\mathsf{crs}_1, x, w)$$

$$\Rightarrow \quad |\Pr[1 \leftarrow \mathcal{A}(\mathsf{crs}_0, \pi_0)] - \Pr[1 \leftarrow \mathcal{A}(\mathsf{crs}_1, \pi_1)]| \leq \varepsilon(1^\lambda).$$

Looking ahead, as we will study NIZK in the GGM, we will call a NIZK *algebraic* if soundness and zero-knowledge hold against any GPPT adversary, i.e. with unbounded computational power but limited to perform a polynomially bounded number of queries to the GGM oracles. Analogously an Algebraic NIZK-AoK is an argument of knowledge against GPPT adversaries.

Part II

Main Results

Chapter 3

Signatures in Generic Groups

Chapter Overview

In this chapter we investigate signature schemes in Maurer's Generic Group Model (see Section 2.1.2). The main theorems we prove, Theorem 3.2.1 and 3.3.1, provide two attacks against such schemes that are either *efficient* but only addresses signatures with linear verification equations or GPPT (i.e. efficient only in terms of group operations) but covers all schemes in the GGM. In summary, for a scheme with verification key vk containing n_2 group elements, and with message space $\mathsf{S.M}$, both attack allows finding $n_2 - |\mathsf{S.M}|$ forgeries in at most n_2 queries. Results in this section appeared in [32] and extend some of the results in [53].

3.1 Definitions

3.1.1 ϑ-Unforgeability

We begin presenting a new notion of security for signatures which weakens the standard Unforgeability one [68]. This property, which we call ϑ-unforgeability, informally requires the existence of an adversary capable of producing more than ϑ forgeries, with ϑ being a function of the public parameters and the number of signature queries performed. We do so for the sake of generality as the attacks presented later in this chapter affects such weaker notion. Moreover this higher level of generality will prove useful in Chapter 4 to conduct our study of Vector Commitments in Generic Groups. A formal definition follows.

Definition 3.1.1 (ϑ-UF)**.** *Given a function $\vartheta : \{0,1\}^* \to \mathbb{N}$ and a signature scheme we define the ϑ-Unforgeability Experiment as in Fig. 3.1. The advantage of an adversary $\mathcal{A}$ is defined as*

$$\mathsf{Adv}^{\vartheta\text{-}\mathsf{UF}}(\mathcal{A}) \;=\; \Pr\left[\mathsf{Exp}^{\vartheta\text{-}\mathsf{UF}}_{\mathcal{A}} = 1\right].$$

A scheme is ϑ-Unforgeable if any PPT adversary has negligible advantage.

To provide more intuition about this notion we observe that setting $\vartheta = 0$ yields the classic unforgeability under chosen message attacks (UF-CMA) [68] security definition. For higher

$\mathsf{Exp}^{\vartheta\text{-}\mathsf{UF}}_{\mathcal{A}}$ **with adversary** $\mathcal{A}$:

1 : Initialize $Q \leftarrow \varnothing$, generate $\mathsf{sk}, \mathsf{vk} \leftarrow^{\$} \mathsf{S.Setup}(1^\lambda)$ and send $\mathcal{A} \leftarrow \mathsf{vk}$
2 : **When** $\mathcal{A} \to m \in \mathsf{S.M}$:
3 : Sign $\sigma \leftarrow^{\$} \mathsf{S.Sign}(\mathsf{sk}, m)$, store $Q \leftarrow Q \cup (m, \sigma)$ and send $\mathcal{A} \leftarrow \sigma$
4 : **When** $\mathcal{A} \to F$:
5 : Return 1 if the following conditions are satisfied:
6 : For all $(m, \sigma) \in F$, the signature is correct, i.e. $\mathsf{S.Vfy}(\mathsf{vk}, m, \sigma) \to 1$
7 : Messages in F were not queried, i.e. $(m, \sigma) \in F \;\Rightarrow\; (m, \cdot) \notin Q$
8 : $|\{m \,:\, (m, \cdot) \in F\}| > \vartheta(\mathsf{vk}, Q)$
9 : Else return 0

Figure 3.1: ϑ-Unforgeability Experiment for a given signature scheme

values of ϑ we obtain progressively weaker definitions until $\vartheta(\mathsf{vk}, Q) = |\mathsf{S.M}|$, which is trivially true for any scheme. The notion of t-time security (e.g. [82]) is also captured by our definition setting

$$\vartheta(\mathsf{vk}, Q) \;=\; \begin{cases} 0 & \text{If } |Q| \leq t \\ |\mathsf{S.M}| & \text{If } |Q| > t. \end{cases}$$

3.1.2 Strictly Linear and Generic Verification

In order to study signatures relying on a "black-box" prime order group we introduce (an equivalent version of) *algebraic signatures* [53] where the verification procedure is only allowed to test a system of linear equations. Our definition contains, with respect to the original, a minor syntactical addition: we split $\mathsf{S.Setup}$ in a CRS-generator $\mathsf{S.SetupCRS}$ which returns the public parameters (a list of group elements $\mathbf{X}_1$) and the actual key generation algorithm $\mathsf{S.SetupKey}(\mathbf{X}_1)$ which produces vk and sk. Note there is no loss of generality assuming this structure as $\mathsf{S.SetupCRS}$ may return an empty vector which could then be ignored by $\mathsf{S.SetupKey}$.

Definition 3.1.2. *A signature scheme* $(\mathsf{S.Setup}, \mathsf{S.Sign}, \mathsf{S.Vfy})$ *is said to be algebraic with linear verification if*

- $\mathsf{S.Setup}$ *is divided into two algorithms* $\mathsf{S.SetupCRS}$ *and* $\mathsf{S.SetupKey}$ *respectively returning* $\mathsf{S.SetupCRS}(1^\lambda) \xrightarrow{\$} (\mathbf{X}_1, s_1) \in \mathbb{G}^{n_1} \times \{0,1\}^*$ *and* $\mathsf{S.SetupKey}(1^\lambda, \mathbf{X}_1, s_1) \xrightarrow{\$} (\mathsf{sk}, \mathsf{vk})$ *with*

$$\mathsf{vk} = (\mathbf{X}, s) \in \mathbb{G}^n \times \{0,1\}^* \quad : \quad \mathbf{X} = \mathbf{X}_1 || \mathbf{X}_2, \qquad \mathbf{X}_2 \in \mathbb{G}^{n_2}, \qquad s = s_1 || s_2.$$

- $\mathsf{S.Sign}(\mathsf{sk}, m) \xrightarrow{\$} \sigma$ *where* $\sigma = (\mathbf{Y}, \mathbf{z})$ *with* $\mathbf{Y} \in \mathbb{G}^k$ *and* $\mathbf{z} \in \mathbb{F}_q^h$.
- *There exist* $A : \mathbb{F}_q^h \times \mathsf{S.M} \times \{0,1\}^* \to \mathbb{F}_q^{\ell,n}$ *and* $B : \mathbb{F}_q^h \times \mathsf{S.M} \times \{0,1\}^* \to \mathbb{F}_q^{\ell,k}$ *matrices such that* $\mathsf{S.Vfy}(\mathsf{vk}, m, \sigma) \to 1$ *if and only if* $\sigma = (\mathbf{z}, \mathbf{Y})$ *and*

$$A(\mathbf{z}, m, s) \cdot \mathbf{X} \;=\; B(\mathbf{z}, m, s) \cdot \mathbf{Y}.$$

Furthermore the scheme is said to have strictly linear verification if $A(\mathbf{z}, m, s)$ is an affine function of $\mathbf{z}$ and $B(m, s)$ does not depend on $\mathbf{z}$.

When clear from the context we will omit for clarity the argument s in the matrices A, B above. Next we provide an analogous for algebraic vector commitments with generic verification. As in the previous definition we split the setup algorithm into a procedure that prepares the CRS and another one that uses the CRS, oblivious to any trapdoor information about it, to compute the secret and verification keys.

Definition 3.1.3. *A signature scheme* $(\mathsf{S.Setup}, \mathsf{S.Sign}, \mathsf{S.Vfy})$ *is said to be algebraic with generic verification if, in the GGM, all algorithms have access to* $\mathcal{O}_{\mathsf{add}}$ *and* $\mathcal{O}^0_{\mathsf{eq}}$*. Furthermore we require* $\mathsf{S.Setup}$ *to be divided into two algorithms* $\mathsf{S.SetupCRS}$ *and* $\mathsf{S.SetupKey}$ *such that* $\mathsf{S.SetupCRS}(1^\lambda) \xrightarrow{\$} (\mathbf{X}_1, s_1) \in \mathbb{G}^{n_1} \times \{0,1\}^*$ *and* $\mathsf{S.SetupKey}(1^\lambda, \mathbf{X}_1, s_1) \xrightarrow{\$} \mathsf{sk}, \mathsf{vk}$ *with*

$$\mathsf{vk} = (\mathbf{X}, s) \in \mathbb{G}^n \times \{0,1\}^* \quad : \quad \mathbf{X} = \mathbf{X}_1 || \mathbf{X}_2, \qquad \mathbf{X}_2 \in \mathbb{G}^{n_2}, \qquad s = s_1 || s_2.$$

3.2 Attack against Strictly Linear Verification

3.2.1 Attack Description

We now provide an *efficient* attack for algebraic signatures with strictly linear verification. Note that, as opposed to the attack presented in the following section, which will be efficient only in terms of generic group operations, the one presented here effectively runs in probabilistic polynomial time. The same notation of Definition 3.1.2 will be used below without further reference.

Theorem 3.2.1. *Given a signature scheme with strictly linear verification, for any ϑ polynomially bounded such that $n_2 + \vartheta \leq |\mathsf{S.M}|$ there exists a* PPT *algorithm $\mathcal{A}$ that in the unforgeability experiment in Fig. 3.1 performs at most n_2 queries and produces ϑ distinct forgeries with significant probability.*

For the sake of presentation we build $\mathcal{A}$ describing first a subroutine $\mathcal{B}$ which breaks security by doing potentially more signing queries that n_2. Next, we show how $\mathcal{A}$ can use $\mathcal{B}$ in a black-box way to realize the full attack with at most n_2 queries.

Similarly to the attack described in [53], upon receiving the verification key $(\mathbf{X}, s)$, the subroutine $\mathcal{B}$ (described formally in Figure 3.2) keeps track of all possible exponents of $\mathbf{X}$ in an affine space $L \subseteq \mathbb{F}_q^n$. Then for each message m_i either a forgery can be produced or a new linear relation on $\mathbf{X}$ is found, thus decreasing $\dim L$, at the cost of a signature query. This is done by checking if the system $A(\mathbf{z}, m)\mathbf{x} = B(m)\mathbf{y}$ can be solved for a given $\mathbf{z} \in \mathbb{F}_q^h$ and all $\mathbf{x} \in L$. More specifically we define $S(L, m)$, the *solutions* set, as the collection of all those $\mathbf{z}$ for which any $\mathbf{x} \in L$ makes the systems solvable, formally

$$S(L, m) \;=\; \{\mathbf{z} \in \mathbb{F}_q^h \;:\; A(\mathbf{z}, m) \cdot L \subseteq \operatorname{Im} B(m)\}.$$

If $S(L, m)$ is easy to compute, a strategy for $\mathcal{B}$ is to check whether $S(L, m) \neq \varnothing$ and in this case to get any $\mathbf{z} \in S(L, m)$ and find, using pseudo-inverses or Gaussian elimination, a vector $\mathbf{Y} \in \mathbb{G}^k$ such that $A(\mathbf{z}, m)\mathbf{X} = B(m)\mathbf{Y}$. Conversely, if $S(L, m) = \varnothing$, $\mathcal{B}$ may

request a signature $(\mathbf{Y}, \mathbf{z})$, which implies that the exponent $\mathbf{x}$ of $\mathbf{X}$ satisfies the condition $A(\mathbf{z}, m)\mathbf{x} \in \operatorname{Im} B(m)$. Notice that, unlike the attack presented in [53], $\mathcal{B}$ is required to be PPT and thus computing $S(L, m)$ efficiently is essential in our argument. This will follow as we assumed the verification to be strictly linear, implying that $S(L, m)$ is an affine space.

Although $\mathcal{B}$ effectively breaks security, we can only upper bound the number of signatures queried by $n_1 + n_2$, i.e. one for each group element in the CRS $\mathbf{X}_1$ and verification key $\mathbf{X}_2$, since initially $L = \mathbb{F}_q^{n_1+n_2}$ with dimension $n_1 + n_2$. In order to reduce the requested signatures to be at most n_2 we introduce a preprocessing phase to find as many linear relations among group elements of the CRS as possibile and then run $\mathcal{B}$ providing as input a refined space L. Informally, if $\mathcal{B}$ is unable to find new relations among the elements of $\mathbf{X}_1$, then $\dim L$ can at most decrease by n_2, yielding the desired upper bound.

To conclude we then need to describe how the preprocessing is carried out: The core idea is to initialize the set of possible exponents $V = \mathbb{F}_q^{n_1}$ and execute several times $\mathcal{B}(\mathsf{vk}^*, V)$ replying to signing queries with $\mathsf{S.Sign}(\mathsf{sk}^*, \cdot)$ where $\mathsf{vk}^*, \mathsf{sk}^* \leftarrow^{\$} \mathsf{S.SetupKey}(1^\lambda, \mathbf{X}_1, s_1)$ is freshly sampled each time. If in some of those executions $\mathcal{B}$ is able to find a new relation among the group elements, then V is updated accordingly (lowering its dimension by at least 1), and a new round of simulations is run. Conversely if $\mathcal{B}(\mathsf{vk}^*, V)$ fails to find new relations several times in this simulated environment, then it is executed one last time with the real verification key vk and signing oracle. If no new relation is found in this last execution, $\mathcal{A}$ concludes by returning the forgeries found by $\mathcal{B}$. Otherwise $\mathcal{A}$ aborts.

Informally $\mathcal{A}$ aborts with low probability since the simulated and real executions are identically distributed from $\mathcal{B}$ perspective and in particular since no relation is found among the many simulated executions, it is unlikely this will happen in the real one. Finally we remark that simulating the signature challenger in this preprocessing phase is crucial. In this way the only signature queries performed by $\mathcal{A}$ are those requested by the last execution of $\mathcal{B}$.

Adversary $\mathcal{B}(\mathsf{vk}, V)$:

1 : Set $L \leftarrow V \times \mathbb{F}_q^{n_2} \subseteq \mathbb{F}_q^{n_1+n_2}$
2 : Initialize the set of forgeries $F \leftarrow \varnothing$ and call $\theta \leftarrow n_2 + \vartheta$
3 : Sample $m_1, \ldots, m_\theta \leftarrow^{\$} \mathsf{S.M}$ distinct messages
4 : **For** $i \in \{1, \ldots, \theta\}$:
5 : **If** $S(L, m_i) \neq \varnothing$:
6 : Get a vector $\mathbf{z} \in S(L, m_i)$
7 : Find a solution $\mathbf{Y} \in \mathbb{G}^k$ such that $A(\mathbf{z}, m_i)\mathbf{X} = B(m_i)\mathbf{Y}$
8 : Set $\sigma \leftarrow (\mathbf{Y}, \mathbf{z})$ and store $F \leftarrow F \cup \{(m_i, \sigma)\}$
9 : **Else**:
10 : Query m_i to the challenger and get $\sigma = (\mathbf{Y}, \mathbf{z})$
11 : Update $L \leftarrow L \cap \{\mathbf{x} \in \mathbb{F}_q^n \ : \ A(\mathbf{z}, m_i)\mathbf{x} \in \operatorname{Im} B(m_i)\}$
12 : Return F, L

Figure 3.2: $\mathcal{B}$ breaking ϑ-UF of an algebraic signature with strictly linear verification.

Adversary $\mathcal{A}^{\mathsf{S.Sign}(\mathsf{sk},\cdot)}(\mathsf{vk})$:

1 : Parse $\mathsf{vk} = (\mathbf{X}, s)$ with $\mathbf{X} = \mathbf{X}_1||\mathbf{X}_2$ and $s = s_1||s_2$
2 : Initialize $V \leftarrow \mathbb{F}_q^{n_1}$ the space of potential exponents of $\mathbf{X}_1$
3 : **Do**:
4 : **For** $2n_1 + 1$ times:
5 : $\mathsf{vk}^*, \mathsf{sk}^* \leftarrow^{\$} \mathsf{S.SetupKey}(1^\lambda, \mathbf{X}_1, s_1)$
6 : Execute $F^*, L^* \leftarrow^{\$} \mathcal{B}^{\mathsf{S.Sign}(\mathsf{sk}^*,\cdot)}(\mathsf{vk}^*, V)$
7 : Set $V^* \leftarrow \{\mathbf{x}_1 : \exists \mathbf{x}_2 : \mathbf{x}_1||\mathbf{x}_2 \in L^*\}$ the projection of L^* on $\mathbb{F}_q^{n_1}$
8 : **If** $V^* \neq V$:
9 : Update $V \leftarrow V^*$, **break**
10 : **Until** the for-cycle ends without interruptions
11 : Execute $F, L \leftarrow^{\$} \mathcal{B}^{\mathsf{S.Sign}(\mathsf{sk},\cdot)}(\mathsf{vk}, V)$
12 : Compute V^* as the projection of L on $\mathbb{F}_q^{n_1}$
13 : **If** $V^* \neq V$: Return fail
14 : **Else**: Return F

Figure 3.3: $\mathcal{A}$ breaking the ϑ-UF of an algebraic signature using as subroutine an algorithm $\mathcal{B}$, which is that of Fig. 3.2 in the case of schemes with strictly linear verification, or that of Fig. 3.4 in the case of schemes with generic verification.

3.2.2 Proof

Proof of Theorem 3.2.1. Having provided the intuition behind the attacker $\mathcal{A}$ built on top of $\mathcal{B}$, we now proceed to prove the theorem through a sequence of claims. We begin by stating the following properties about $\mathcal{B}(\mathsf{vk}, V)$ where we denote $\mathsf{vk} = (\mathbf{X}, s)$ with $\mathbf{X} = \mathbf{X}_1||\mathbf{X}_2$, $\mathbf{x}_1$ the discrete logarithm of $\mathbf{X}_1$ and $\mathbf{x}$ the discrete logarithm of $\mathbf{X}$. Finally we denote $\pi : \mathbb{F}_q^{n_1} \times \mathbb{F}_q^{n_2} \to \mathbb{F}_q^{n_1}$ the projection on first component, i.e. $\pi(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1$.

Claim 3.2.1. *If L is an affine space, $S(L, m)$ is an affine space. Moreover an affine base for $S(L, m)$ can be computed in polynomial time.*

Claim 3.2.2. *If $\mathbf{x}_1 \in V$ then at any step of $\mathcal{B}(\mathsf{vk}, V)$, $\mathbf{x} \in L$.*

Claim 3.2.3. *If $\mathbf{x}_1 \in V$, $\mathcal{B}$ is* PPT *and upon returning (F, L), F is a set of valid forgeries.*

Claim 3.2.4. *For a given m_i, if the condition at step 5 is not satisfied, i.e. $S(L, m_i) = \varnothing$, then after step 11 the dimension of L decreases strictly.*

Claim 3.2.5. *After the execution of line 1, Fig 3.2,* $\dim L = n_2 + \dim V$ *and if $\mathcal{B}(\mathsf{vk}, V)$ returns (F, L) with $\pi(L) = V$ then* $\dim L \geq \dim V$.

Next we state the following properties about $\mathcal{A}$

Claim 3.2.6. *$\mathcal{A}$ is* PPT.

Claim 3.2.7. *At any step of $\mathcal{A}$ execution,* $\mathbf{x}_1 \in V$.

Claim 3.2.8. *$\mathcal{A}$ fails with probability* $\Pr[\mathcal{A}(\mathsf{vk}) \to \mathsf{fail}] \leq 1/2$.

First we observe these claims imply the thesis. Indeed by Claim 3.2.8, with probability greater than $1/2$, $\mathcal{A}$ does not return fail. By construction, this implies that in the last execution $\mathcal{B}(\mathsf{vk}, V)$ returns (F, L) with $\pi(L) = V$. Thus by Claim 3.2.5 $n_2 + \dim V \geq L \geq \dim V$ at any step of $\mathcal{B}$ during its last execution. As a consequence $\dim L$ can decrease at most n_2 times. Applying Claim 3.2.4 we conclude that $S(L, m_i) = \varnothing$ can happen at most n_2 times because each time this occurs, $\dim L$ decreases. It follows then that for at least $\theta - n_2 = \vartheta$ messages, the condition $S(L, m_i) \neq \varnothing$ is satisfied, meaning that $\mathcal{B}$ adds a new signature to the set F, which in the end will have cardinality $|F| \geq \vartheta$. Finally, since $\mathbf{x} \in V$ by Claim 3.2.7, we can apply Claim 3.2.3 to conclude that F is a valid set of forgeries, implying that $\mathcal{A}$ breaks ϑ-UF.

Next, we provide a proof for each of these claims:

Proof of Claim 3.2.1. We start observing that if L is any set and $\mathbf{x}_1, \dots, \mathbf{x}_d \in L$ is a base for the linear span of L then $S(L, m) = \bigcap_{i=1}^{d} S(\mathbf{x}_i, m)$. By construction, $\mathbf{x}_i \in L$ implies $S(L, m) \subseteq S(\mathbf{x}_i, m)$, and in particular $S(L, m) \subseteq \cap_{i=1}^{d} S(\mathbf{x}_i, m)$. Conversely let $\mathbf{z}$ be a vector in the intersection of all $S(\mathbf{x}_i, m)$. We can find vectors $\mathbf{u}_i \in \mathbb{F}_q^k$ such that $A(\mathbf{z}, m)\mathbf{x}_i = B(m)\mathbf{u}_i$. Since $\mathbf{x}_1, \dots, \mathbf{x}_d$ is a base for the linear span of L, for any $\mathbf{x} \in L$ we can express it as a linear combination $\alpha_1 \mathbf{x}_1 + \ldots + \alpha_d \mathbf{x}_d$. In conclusion

$$A(\mathbf{z}, m)\mathbf{x} \;=\; \sum_{i=1}^{d} \alpha_i A(\mathbf{z}, m)\mathbf{x}_i \;=\; \sum_{i=1}^{d} \alpha_i B(m)\mathbf{u}_i \;=\; B(m) \sum_{i=1}^{d} \alpha_i \mathbf{u}_i.$$

Thus $A(\mathbf{z}, m)\mathbf{x} \in \operatorname{Im} B(m)$ and in particular $\mathbf{z} \in S(L, m)$.

In order to show that $S(L, m)$ is efficiently computable it suffices to show that $S(\mathbf{x}, m)$ can be computed in polynomial time for any point $\mathbf{x}$. To this aim let $f_{\mathbf{x}} : \mathbb{F}_q^h \to \mathbb{F}_q^\ell$ be such that $f(\mathbf{z}) = A(\mathbf{z}, m)\mathbf{x}$. Since the scheme has strictly linear verification (Definition 3.1.2) $A(\,\cdot\,, m)$ is an affine map and so is f. Furthermore by construction $S(\mathbf{x}, m) = f_{\mathbf{x}}^{-1}(\operatorname{Im} B(m))$ since $\mathbf{z} \in S(\mathbf{x}, m)$ if and only if $A(\mathbf{z}, m)\mathbf{x} \in \operatorname{Im} B(m)$. This concludes the argument as the preimage through an affine map of a linear space is an affine space which can be computed in polynomial time. □

Proof of Claim 3.2.2. If $\mathbf{x}_1 \in V$ then $\mathbf{x} = \mathbf{x}_1 || \mathbf{x}_2 \in V \times \mathbb{F}_q^{n_2}$ which by construction implies that, when L is initialized, $\mathbf{x} \in L$. Next assume by induction $\mathbf{x} \in L$ in all previous steps. The only instruction in $\mathcal{B}$ that may modify L is in step 11 and when this is executed, since $\sigma = (\mathbf{Y}, \mathbf{z})$ is a valid signature by perfect correctness, we have

$$A(\mathbf{z}, m_i)\mathbf{X} = B(m_i)\mathbf{Y} \quad \Rightarrow \quad A(\mathbf{z}, m_i)\mathbf{x} \in \operatorname{Im} B(m_i).$$

□

Proof of Claim 3.2.3. To prove that $\mathcal{B}$ is a PPT algorithm, observe that the for-loop is executed $\theta = n_2 + \vartheta$, that is polynomially bounded, times. Inside the loop, checking $S(L, m_i) \neq \varnothing$ and

possibly computing a $\mathbf{z} \in S(L, m_i)$ can be done efficiently from Claim 3.2.1 by computing a base for it. Next, calling $\mathbf{x}$ the discrete logarithm of $\mathbf{X}$, we have that $A(\mathbf{z}, m_i)\mathbf{x} \in \operatorname{Im} B(m_i)$ because

$$\mathbf{z} \in S(L, m_i) \quad \Rightarrow \quad A(\mathbf{z}, m_i) \cdot L \subseteq \operatorname{Im} B(m_i) \quad \Rightarrow \quad A(\mathbf{z}, m_i)\mathbf{x} \in \operatorname{Im} B(m_i)$$

where the last implication follows as $\mathbf{x} \in L$ by Claim 3.2.2 and the assumption $\mathbf{x}_1 \in V$. Thus, calling H a weak-inverse[1] of $B(m_i)$, which can be computed efficiently, the vector $\mathbf{Y}$ can be set as $H \cdot A(\mathbf{z}, m_i)\mathbf{X}$. Indeed, as $A(\mathbf{z}, m_i)\mathbf{X} \in \operatorname{Im} B(m_i)$ there exists a vector $\mathbf{Z} \in \mathbb{G}^k$ such that $A(\mathbf{z}, m_i)\mathbf{X} = B(m_i)\mathbf{Z}$ and in particular

$$B(m_i)\mathbf{Y} \;=\; B(m_i)HA(\mathbf{z}, m_i)\mathbf{X} \;=\; B(m_i)HB(m_i)\mathbf{Z} \;=\; B(m_i)\mathbf{Z} \;=\; A(\mathbf{z}, m_i)\mathbf{X}.$$

Finally, given the bases of two affine spaces, a base of their intersection can be computed efficiently. This conclude the proof that $\mathcal{B}$ is PPT.

For the second part, by construction each entry in F is of the form $(m_i, \mathbf{Y}, \mathbf{z})$ such that

$$A(\mathbf{z}, m_i)\mathbf{X} = B(m_i)\mathbf{Y}.$$

Therefore, by our definition of signatures with linear verification scheme, the verifier accepts $(m_i, \mathbf{Y}, \mathbf{z})$. The claim is thus proven. □

Proof of Claim 3.2.4. Since the condition at step 5 is not satisfied, $S(L, m_i) = \varnothing$ and in particular $\mathbf{z} \notin S(L, m_i)$ implying that $A(\mathbf{z}, m_i)\mathbf{x} \notin \operatorname{Im} B(m_i)$ for some $\mathbf{x} \in L$. Therefore L is not contained in the space of all $\mathbf{x}$ such that $A(\mathbf{z}, m_i)\mathbf{x} \in \operatorname{Im} B(m_i)$ and in particular its dimension decreases after the execution of step 11 □

Proof of Claim 3.2.5. The first part follow as L is initially $V \times \mathbb{F}_q^{n_2}$ of dimension $\dim V + n_2$. The second part follows by linear algebra since $\dim L \geq \dim \pi(L) = \dim V$. □

Proof of Claim 3.2.6. Since $\mathsf{S.SetupKey}, \mathsf{S.Sign}$ and $\mathcal{B}$ are PPT algorithm, by Claim 3.2.3 in the last case, each step in the loop can be computed efficiently. In particular, as $2n_1 + 1$ is polynomially bounded, each for-loop in $\mathcal{A}$ can be performed efficiently.

Next we show that the procedure inside the Do-Until loop is repeated at most n_1+1 times. The key observation is that during the execution of $\mathcal{B}$, the space L forms a monotone decreasing sequence, implying that when $\mathcal{B}(\mathsf{vk}^*, V) \to (F^*, L^*)$ then $L^* \subseteq V \times \mathbb{F}_q^{n_2}$. In particular this implies that $\pi(L^*) \subseteq \pi(V \times \mathbb{F}_q^{n_2}) = V$. Thus if at any point the for-loop is halted, $\pi(L^*) = V^* \neq V$ implies $V^* \subseteq V$. Hence the dimension of V strictly decreases, and since initially $\dim(V) = n_1$, the foor-loop can be halted at most n_1 times.

Finally, using again that $\mathcal{B}$ is an efficient algorithm, computing F, L can be done in polynomial time. It follows that $\mathcal{A}$ is PPT. □

Proof of Claim 3.2.7. We proceed by induction. Initially $V = \mathbb{F}_q^{n_1}$ implies $\mathbf{x}_1 \in V$. Next we observe that the value of V is only changed if, within the for-loop, $V^* \neq V$ (see step 8,

[1] H is the weak-inverse of A if $A \cdot H \cdot A = A$

Fig. 3.3). Assume by induction that before this step is executed $\mathbf{x}_1 \in V$. Then, when this happens, $\mathcal{B}(\mathsf{vk}^*, V) \to (F^*, L^*)$ had been executed with $\mathbf{x}_1 \in V$. By Claim 3.2.2 this implies that $\mathbf{x} \in L^*$ and in particular $\mathbf{x}_1 = \pi(\mathbf{x}) \in \pi(L^*) = V^*$. Thus when $\mathcal{A}$ sets $V \leftarrow V^*$, $\mathbf{x}_1 \in V$. □

Proof of Claim 3.2.8. Define the following events:

- $\mathcal{E}_{i,j}$ = "During the i-th iteration of the Do-Until loop, and the j-th iteration of the for loop, $\mathcal{B}^{\mathsf{S.Sign}(\mathsf{sk}^*,\cdot)}(\mathsf{vk}^*, V)$ returns (F^*, L^*) such that $\pi(L^*) = V$".
- $\mathcal{E}_{\mathsf{last}}$ = "$\mathcal{B}^{\mathsf{S.Sign}(\mathsf{sk},\cdot)}(\mathsf{vk}, V)$ returns F, L with $\pi(L) = V$".

Furthermore let $I \sim \{1, \dots, n_1 + 1\}$ be the random variable such that $\mathcal{A}$ terminates the Do-Until loop after the I-th execution. Then we observe that, conditioned on $\mathbf{X}_1, s_1$ and the V at iteration i, the event $\mathcal{E}_{i,j}$ depends only on the random coins used for $\mathcal{B}$, $\mathsf{S.SetupKey}$ and $\mathsf{S.Sign}$ which are chosen independently at each execution of $\mathcal{B}$. In particular, for a fixed i, the events $\{\mathcal{E}_{i,j}\}_j$ are independent and, since for $\mathcal{E}_{i,j}, \mathcal{E}_{i,k}$ with $j \neq k$ the procedure $\mathcal{B}$ is invoked with the same input, $\Pr[\mathcal{E}_{i,j}] = \Pr[\mathcal{E}_{i,k}]$.

We may therefore define $p_i = \Pr[\mathcal{E}_{i,1}]$ as the success probability of each execution of $\mathcal{B}$ during the i-th loop. Similarly, if $I = i$, the vector space V given in input to $\mathcal{B}$ is by construction equal to the one used during the i-th execution of the Do-Until loop. In particular

$$p_i = \Pr[\mathcal{E}_{\mathsf{last}} | I = i].$$

To conclude we show that

$$\begin{aligned}
\Pr[\mathcal{A} \to \mathsf{fail}] &= \Pr[\neg\mathcal{E}_{\mathsf{last}}] = \textstyle\sum_{i=1}^{n_1+1} \Pr[\neg\mathcal{E}_{\mathsf{last}} | I = i] \cdot \Pr[I = i] \\
&\leq \textstyle\sum_{i=1}^{n_1+1} \Pr[\neg\mathcal{E}_{\mathsf{last}} | I = i] \cdot \Pr[\mathcal{E}_{i,1} \wedge \ldots \wedge \mathcal{E}_{i,2n_1+1}] \\
&= \textstyle\sum_{i=1}^{n_1+1} \Pr[\neg\mathcal{E}_{\mathsf{last}} | I = i] \cdot \prod_{j=1}^{2n_1+1} \Pr[\mathcal{E}_{i,j}] \\
&= \textstyle\sum_{i=1}^{n_1+1} (1 - p_i) \cdot p_i^{2n_1+1} \\
&\leq \textstyle\sum_{i=1}^{n_1+1} \frac{1}{2n_1+2} = \frac{n_1+1}{2n_1+2} = \frac{1}{2}.
\end{aligned}$$

where the first inequality comes from the fact that $I = i$ implies $\mathcal{E}_{i,j}$ for all $j \in \{1, \dots, 2n_1+1\}$, while the second inequality comes from the fact that the function $f_t(x) = (1-x)x^t$ is upper bounded by $1/(t+1)$ when $x \in [0,1]$. Indeed $f_t(0) = f_t(1) = 0$ and its derivative vanishes only at $t/(t+1)$, which has to be the maximum point, implying that

$$(1-x) \cdot x^t \leq \left(1 - \frac{t}{t+1}\right) \cdot \left(\frac{t}{t+1}\right)^t \leq \frac{1}{t+1}.$$ □

This completes the proof. □

3.3 Attack against Generic Verification

3.3.1 Attack Description

Theorem 3.3.1. *Given an algebraic signature scheme with generic verification, for any ϑ such that $n_2 + \vartheta \leq |\mathsf{S.M}|$ there exists an adversary $\mathcal{A}$ that in the unforgeability experiment in Fig. 3.1 performs at most n_2 signature queries and produces ϑ distinct forgeries.*

Moreover, calling κ an upper bound on the signature bit-length, and χ an upper bound on the number of queries $\mathsf{S.Vfy}$ performs to $\mathcal{O}^0_{\mathsf{eq}}$, then $\mathcal{A}$ runs in time $O(\vartheta \cdot 2^\kappa \cdot 2^\chi \cdot \mathsf{poly}(\lambda))$ and performs $O(\vartheta \cdot \mathsf{poly}(\lambda))$ queries to $\mathcal{O}_{\mathsf{add}}$ and $\mathcal{O}^0_{\mathsf{eq}}$.

As done in Section 3.2 we begin by providing an attack $\mathcal{B}$ which breaks the scheme but performs potentially $n_1 + n_2$ signature queries. At a high level $\mathcal{B}$, given the verification key $\mathsf{vk} = (\mathbf{X}, s)$, will keep track of all possible exponents of $\mathbf{X}$ in a set L and for each message m either the dimension of L decreases by one or $\mathcal{B}$ finds a forgery. Assume without loss of generality that signatures are of the form $(\mathbf{Y}', t')$ with $\mathbf{Y}' \in \mathbb{G}^k$ and $t' \in \{0,1\}^\kappa$.

For any m, our adversary attempts to produce a forgery as follows: For all possible $t \in \{0,1\}^\kappa$, it executes the verification algorithm by simulating a generic group $\widetilde{\mathbb{G}}$ with oracles $\widetilde{\mathcal{O}}_{\mathsf{add}}$ and $\widetilde{\mathcal{O}}^0_{\mathsf{eq}}$. More specifically, since $\mathsf{S.Vfy}$ requires as input the verification key $(\mathbf{X}, s)$, the message m and the signatures $(\mathbf{Y}, t)$, $\mathcal{B}$ reproduces all the group elements involved by assigning dummy indexes for $\widetilde{\mathbf{X}}$, $\widetilde{\mathbf{Y}}$ and runs $\mathsf{S.Vfy}((\widetilde{\mathbf{X}}, s), m, (\widetilde{\mathbf{Y}}, t))$. During the execution, each query to $\widetilde{\mathcal{O}}_{\mathsf{add}}$ is emulated by simply returning new incremental indexes, while to emulate $\widetilde{\mathcal{O}}^0_{\mathsf{eq}}$, χ bits $\beta_1, \ldots, \beta_\chi$ are chosen at the beginning of the execution so that the answer to the i-th query will be β_i. Note that each element T_i the verifier queries to $\widetilde{\mathcal{O}}^0_{\mathsf{eq}}$ has to be a linear combination of the initial group elements he received, i.e. $T_i = \mathbf{a}_i^\top \widetilde{\mathbf{X}} - \mathbf{b}_i^\top \widetilde{\mathbf{Y}} - c_i \cdot \widetilde{G}$ obtained though $\widetilde{\mathcal{O}}_{\mathsf{add}}$, and $\mathcal{B}$ can extract these coefficients.

Repeating the execution of $\mathsf{S.Vfy}$ for different values of $\beta_1, \ldots, \beta_\chi$ implicitly defines a tree of height χ in which paths are determined by the replies $\mathcal{B}$ gave at the i-th query to $\widetilde{\mathcal{O}}^0_{\mathsf{eq}}$. If at some point a path $\beta_1, \ldots, \beta_\chi$ that makes the verifier accept is found, $\mathcal{B}$ can try to find a vector $\mathbf{Y}$ in the real GGM, such that the i-query $\mathsf{S.Vfy}$ would do to $\mathcal{O}^0_{\mathsf{eq}}$ will be answered with β_i. If such a $\mathbf{Y}$ is found, then $(\mathbf{Y}, t)$ will be a valid forgery for m.

Recalling that the i-th query has the form $T_i = \mathbf{a}_i \widetilde{\mathbf{X}} - \mathbf{b}_i \widetilde{\mathbf{Y}} - c_i \cdot G$, then $\mathcal{B}$ needs to find a vector $\mathbf{Y}$ such that for all $i \in \{1, \ldots, \chi\}$

$$\mathbf{a}_i^\top \mathbf{X} = \mathbf{b}_i^\top \mathbf{Y} + c_i \cdot G \text{ when } \beta_i = 1, \qquad \mathbf{a}_i^\top \mathbf{X} \neq \mathbf{b}_i^\top \mathbf{Y} + c_i \cdot G \text{ when } \beta_i = 0$$

Regarding the equations on the left side, they can be packed up into a system $A\mathbf{X} = B\mathbf{Y} + \mathbf{c} \cdot G$. Through pseudo-inverses or Gaussian elimination is easy to check if solutions exists for all $\mathbf{x} \in L$ (as in the proof of Theorem 3.2.1). If this is not the case $\mathcal{B}$ simply discards this path and continues its brute-force search. However, even if the previous condition is satisfied, for some of the points $\mathbf{x}$ in L it may be the case that any vector $\mathbf{y}$ satisfying $A\mathbf{x} = B\mathbf{y} + \mathbf{c}$ fails to satisfy some of the inequalities above $\mathbf{a}_i^\top \mathbf{x} \neq \mathbf{b}_i^\top \mathbf{y} + c_i$, implying that no solution $\mathbf{Y} \in \mathbb{G}^k$ can be found if $\mathbf{x}$ is the discrete logarithm of $\mathbf{X}$. We call these points $\mathbf{x} \in L$ *faulty* and, more

specifically, the set of faulty points is defined as

$$\mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a},\mathbf{b},c} = \{\mathbf{x} : A\mathbf{x} \in \operatorname{Im} B + \mathbf{c}, \quad \forall \mathbf{y} \in \mathbb{F}_q^m \quad A\mathbf{x} = B\mathbf{y} + \mathbf{c} \Rightarrow \mathbf{a}^\top \mathbf{x} = \mathbf{b}^\top \mathbf{y} + c\}.$$

Three possible cases may occur now:

- If all points in L are faulty with respect to some inequality constraint, then $\mathcal{B}$ gives up on the path as the solution $\mathbf{Y}$ does not exist.
- If not all points are faulty $\mathcal{B}$ attempts to solve the system, which requires expensive queries to $\mathcal{O}_{\mathsf{add}}, \mathcal{O}^0_{\mathsf{eq}}$: if a solution $\mathbf{Y}$ satisfying all constraints is found, this is a valid forgery.
- If not all points are faulty, but no solution can be found, it means that $\mathbf{x}$, the discrete log of $\mathbf{X}$, has to be a faulty point. This information reduces the dimension of L as not all points in L are faulty.

Finally, if no solution can be found for any $t \in \{0,1\}^\kappa$ and path $\beta_1, \ldots, \beta_\chi$, $\mathcal{B}$ queries a signature for m and uses this information to reduce the dimension of L. As for the proof of Theorem 3.2.1, $\mathcal{B}$ might overall query $n_1 + n_2$ signatures (as opposed to the desired n_1) since initially it has no information on the exponents of $\mathbf{X}$, i.e. $\dim L = n_1 + n_2$, and each signature query may reveal only one new linear combination among these group elements. To address this issue we use the same strategy presented in Theorem 3.2.1, that is, we use $\mathcal{B}$ in a black-box way inside the algorithm $\mathcal{A}$, formally described in Fig. 3.3. The main idea is again that $\mathcal{A}$ initially extracts linear combinations among CRS elements that could be found by $\mathcal{B}$, and finally executes $\mathcal{B}$ providing the retrieved information as input. In this way $\mathcal{B}$ will, with significant probability, only find relations among elements of $\mathbf{X}_2$, thus requesting at most n_2 signatures.

A detailed description of $\mathcal{A}$ appears in Fig. 3.4, while a proof of the Theorem is presented in the next section.

3.3.2 Proof

Proof of Theorem 3.3.1. Having provided a description of the adversary $\mathcal{A}$, which uses in a block-box way the procedure $\mathcal{B}$ described in Fig. 3.4, we now show that this adversary satisfies the requirements of the Theorem. As in the previous section we break down the proof into a sequence of Claims which imply the thesis, beginning with a list of properties related to $\mathcal{B}$. In the following we denote $\mathsf{vk} = (\mathbf{X}, s)$ with $\mathbf{X} = \mathbf{X}_1 || \mathbf{X}_2$, $\mathbf{x}_1$ the discrete logarithm of $\mathbf{X}_1$ and $\mathbf{x}$ the discrete logarithm of $\mathbf{X}$. Moreover we call again $\pi : \mathbb{F}_q^{n_1} \times \mathbb{F}_q^{n_2}$ the projection on the first component, i.e. $\pi(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1$.

Claim 3.3.1. *For all $B, \mathbf{c}$ and $\mathbf{a}, \mathbf{b}, c$, the set $\mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a},\mathbf{b},c}$ is an affine space.*

Claim 3.3.2. *An affine base for $\mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a},\mathbf{b},c}$ is efficiently computable.*

Claim 3.3.3. *If $\mathbf{x}_i \in V$ then at any step of $\mathcal{B}(\mathsf{vk}, V)$, $L \subseteq \mathbb{F}_q^n$ is an affine subspace and $\mathbf{x} \in L$.*

Adversary $\mathcal{B}(\mathsf{vk}, V)$:

1 : Initialize $F \leftarrow \varnothing$ the set of forgeries
2 : Call $L = V \times \mathbb{F}_q^{n_2}$ the set of possible exponents of $\mathbf{X}$
3 : Call $\theta = n + \vartheta$ and sample $m_1, \ldots, m_\theta \leftarrow^{\$} \mathsf{S.M}$ distinct messages
4 : **For** $m \in \{m_1, \ldots, m_\theta\}$:
5 : **For** $t \in \{0,1\}^\kappa$ and $(\beta_1, \ldots, \beta_\chi) \in \{0,1\}^\chi$:
6 : Simulate a Generic Group $\widetilde{\mathbb{G}}$ with generator $\widetilde{G}$ and oracles $\widetilde{\mathcal{O}}_{\mathsf{add}}$ and $\widetilde{\mathcal{O}}^0_{\mathsf{eq}}$
7 : Assign indices for two vectors $\widetilde{\mathbf{X}} \in \widetilde{\mathbb{G}}^n$ and $\widetilde{\mathbf{Y}} \in \widetilde{\mathbb{G}}^k$
8 : Run $\mathsf{S.Vfy}((\widetilde{\mathbf{X}}, s), m, (\widetilde{\mathbf{Y}}, t))$ using $\widetilde{\mathbb{G}}$
9 : **When** $\mathsf{S.Vfy}$ queries $\widetilde{\mathcal{O}}_{\mathsf{add}}(T, S)$:
10 : Store a way to express $T + S$ as a linear combination of $\widetilde{\mathbf{X}}$, $\widetilde{\mathbf{Y}}$ and $\widetilde{G}$
11 : Return to $\mathsf{S.Vfy}$ a label for $T + S$
12 : **When** $\mathsf{S.Vfy}$ queries $\widetilde{\mathcal{O}}^0_{\mathsf{eq}}(T_i)$ the i-th time:
13 : Store $\mathbf{a}_i \in \mathbb{F}_q^n$, $\mathbf{b}_i \in \mathbb{F}_q^k$ and $c_i \in \mathbb{F}_q$ such that $T_i = \mathbf{a}_i^\top \widetilde{\mathbf{X}} - \mathbf{b}_i^\top \widetilde{\mathbf{Y}} - c_i \cdot \widetilde{G}$
14 : Return β_i to $\mathsf{S.Vfy}$
15 : **When** $\mathsf{S.Vfy}$ halts and returns $b \in \{0,1\}$:
16 : Let $A = (\mathbf{a}_i \ : \ \beta_i = 1)$, $B = (\mathbf{b}_i \ : \ \beta_i = 1)$ and $\mathbf{c} = (c_i \ : \ \beta_i = 1)$
17 : **If** $b = 0$:
18 : **Continue** cycle in line 5
19 : **Elif** $A \cdot L \nsubseteq \operatorname{Im} B + \mathbf{c}$:
20 : **Continue** cycle in line 5
21 : **Elif** $\exists i \ : \ \beta_i = 0$ and $L \subseteq \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i}$:
22 : **Continue** cycle in line 5
23 : **Elif** $\exists i \ : \ \beta_i = 0$ and $\mathbf{X} \in \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i} \cdot G$:
24 : Update $L \leftarrow L \cap \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i}$
25 : **Break** cycle in line 5
26 : **Else**:
27 : Find $\mathbf{Y} \in \mathbb{G}^k$ s.t. $A\mathbf{X} = B\mathbf{Y} + \mathbf{c}G$ and $\mathbf{a}_i^\top \mathbf{X} \neq \mathbf{b}_i^\top \mathbf{Y} + c_i G$ for $\beta_i = 0$
28 : Store $\sigma \leftarrow (\mathbf{Y}, t)$ and $F \leftarrow F \cup \{(m, \sigma)\}$
29 : **Break** cycle in line 5
30 : **If** the cycle ended without interruptions:
31 : Query a signature for m and wait for $(\mathbf{Y}, t)$
32 : Reconstruct $A, B, \mathbf{c}$ as in step 16 using $(\mathbf{X}, s, m, \mathbf{Y}, t)$ and the group $\mathbb{G}$
33 : Update $L \leftarrow L \cap \{\mathbf{x} \in \mathbb{F}_q^n \ : \ A\mathbf{x} \in \operatorname{Im} B + \mathbf{c}\}$
34 : Return F, L

Figure 3.4: $\mathcal{B}$ breaking security of an algebraic signature scheme with generic verification.

Claim 3.3.4. *If $\mathbf{x}_i \in V$, then $\mathcal{B}(\mathsf{vk}, V)$ performs a polynomially bounded number of queries to the GGM oracles.*

Claim 3.3.5. *Every time either condition at step 23 or 30 is satisfied, when step 24 or 33 respectively are executed, the dimension of L strictly decreases.*

Claim 3.3.6. *When step 27 is executed, then $(\mathbf{Y}, t)$ is a correct signature for m.*

Claim 3.3.7. *After step 2, $\dim L = n_2 + \dim V$. Moreover, when $\mathcal{B}(\mathsf{vk}, V)$ returns (F, L) with $\pi(L) = V$, then $\dim L \geq \dim V$.*

Next we state three claims about the adversary $\mathcal{A}$, when executed with $\mathcal{B}$ as in Fig. 3.4.

Claim 3.3.8. *At any given step of $\mathcal{A}(\mathsf{vk})$, $\mathbf{x}_1 \in V$.*

Claim 3.3.9. *$\mathcal{A}$ performs a polynomially bounded number of queries to the GGM oracles.*

Claim 3.3.10. *$\mathcal{A}(\mathsf{vk})$ fails with probability $\Pr[\mathcal{A}(\mathsf{vk}) \to \mathsf{fail}] \leq 1/2$.*

These completes the proof as in the case of Theorem 3.2.1. □

Proof of Claim 3.3.1. Given $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a},\mathbf{b},c}$ we should show that $\mathbf{w} + (\mathbf{u} - \mathbf{v}) \in \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a},\mathbf{b},c}$. Let $\mathbf{d} = \mathbf{u} - \mathbf{v}$. Then $A\mathbf{u}, A\mathbf{v} \in \operatorname{Im} B + \mathbf{c}$ implies the existence of $\mathbf{r}, \mathbf{s} \in \mathbb{F}_q^m$ such that $A\mathbf{u} = B\mathbf{r} + \mathbf{c}$ and $A\mathbf{v} = B\mathbf{s} + \mathbf{c}$, therefore $A\mathbf{d} = B(\mathbf{r} - \mathbf{s})$. Moreover by construction

$$\mathbf{a}^\top \mathbf{u} = \mathbf{b}^\top \mathbf{r} + \mathbf{c}, \qquad \mathbf{a}^\top \mathbf{v} = \mathbf{b}^\top \mathbf{s} + \mathbf{c} \quad \Rightarrow \quad \mathbf{a}^\top \mathbf{d} = \mathbf{b}^\top (\mathbf{r} - \mathbf{s}).$$

Turning our attention to the vector $\mathbf{w} + \mathbf{d}$, its image through A lies in $\operatorname{Im} B + \mathbf{c}$ because $A\mathbf{d} \in \operatorname{Im} B$. Furthermore for all $\mathbf{y} \in \mathbb{F}_q^m$ such that $B\mathbf{y} + \mathbf{c} = A(\mathbf{w} + \mathbf{d})$

$$\begin{aligned} B(\mathbf{y} - (\mathbf{r} - \mathbf{s})) = A\mathbf{w} \quad &\Rightarrow \quad \mathbf{a}^\top \mathbf{w} = \mathbf{b}^\top (\mathbf{y} - (\mathbf{r} - \mathbf{s})) + c \\ &\Rightarrow \quad \mathbf{a}^\top \mathbf{w} + \mathbf{a}^\top \mathbf{d} = \mathbf{b}^\top \mathbf{y} + c \\ &\Rightarrow \quad \mathbf{a}^\top (\mathbf{w} + \mathbf{d}) = \mathbf{b}^\top \mathbf{y} + c. \end{aligned}$$

Thus $\mathbf{w} + \mathbf{d} \in \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a},\mathbf{b},c}$ proving the claim. □

Proof of Claim 3.3.2. Let H a weak left inverse of B, then we can rewrite $\mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a},\mathbf{b},c}$ as the set of all $\mathbf{x}$ such that $A\mathbf{x} \in \operatorname{Im} B + c$ and

$$\forall \mathbf{w} \qquad \mathbf{a}^\top \mathbf{x} = \mathbf{b}^\top (HA\mathbf{x} + (I - BH)\mathbf{w}) - \mathbf{b}^\top H\mathbf{c} + c$$

using the fact that all solutions to $B\mathbf{y} + \mathbf{c} = \mathbf{t}$ are of the form $H(\mathbf{t} - \mathbf{c}) + (I - BH)\mathbf{w}$ for arbitrary $\mathbf{w}$. Next we introduce an auxiliary space $\mathcal{V}_{\mathbf{r},\mathbf{s},t}$ defined as

$$\mathcal{V}_{\mathbf{r},\mathbf{s},t} \;=\; \{(\mathbf{x}, \mathbf{z}) \in \mathbb{F}_q^n \times \mathbb{F}_q^n \;:\; \mathbf{r}^\top \mathbf{x} = \mathbf{s}^\top \mathbf{z} + t\}$$

Calling $\pi : \mathcal{V}_{\mathbf{r},\mathbf{s},t} \to \mathbb{F}_q^n$ the projection on the first component $(\mathbf{x}, \mathbf{w}) \mapsto \mathbf{x}$ then it's easy to prove that, letting $\mathbf{r} = \mathbf{a} - A^\top H^\top \mathbf{b}$, $\mathbf{s} = (I - BH)^\top \mathbf{b}$ and $t = -\mathbf{b}^\top H \mathbf{c} + c$

$$\begin{aligned} \mathbf{0} \times \mathbb{F}_q^n \not\subseteq \dir(\mathcal{V}_{\mathbf{r},\mathbf{s},t}) \quad &\Rightarrow \quad \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a},\mathbf{b},c} = \varnothing \\ \mathbf{0} \times \mathbb{F}_q^n \subseteq \dir(\mathcal{V}_{\mathbf{r},\mathbf{s},t}) \quad &\Rightarrow \quad \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a},\mathbf{b},c} = \pi(\mathcal{V}_{\mathbf{r},\mathbf{s},t}) \cap \{\mathbf{x} : A\mathbf{x} \in \operatorname{Im} B + \mathbf{c}\} \end{aligned}$$

Since $\mathcal{V}_{\mathbf{r},\mathbf{s},t}$ is an affine hyper-plane, a base of $\operatorname{Im} B + \mathbf{c}$ is efficiently computable, and so is its preimage through A. Thus the thesis follows. □

Proof of Claim 3.3.3. L is an affine subspace since initially $L = V \times \mathbb{F}_q^{n_2}$. By induction, assume that, at a given step, L is an affine subspace. Then it is only updated at step 24 or 33. In both cases, by Claim 3.3.1, we have that L is the intersection of two affine subspaces.

For the second part, initially $\mathbf{x}_1 \in V$ implies $\mathbf{x} = \mathbf{x}_1 || \mathbf{x}_2 \in V \times \mathbb{F}_q^n = L$. Next assume by induction $\mathbf{x} \in L$. If step 24 is executed then the condition 23 is true, meaning that $\mathbf{x} \in \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i}$. Therefore $\mathbf{x}$ still lies in L. Conversely, if step 33 is executed then by construction the signature satisfies $A\mathbf{X} = B\mathbf{Y} + \mathbf{c}$ meaning that $A\mathbf{x} \in \operatorname{Im} B + \mathbf{x}$. Thus again $\mathbf{x} \in L$ after step 33. □

Proof of Claim 3.3.4. The only instructions involving generic group operation (excluding those using the simulated oracles) in the description of $\mathcal{B}$ (Fig. 3.4) are in line 23 to check if $\mathbf{X} \in \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i} \cdot G$ and in line 27 to compute $\mathbf{Y}$.

For each message m_i, within the cycle starting at step 5, these instructions are executed at most once. Indeed if the condition in line 23 is satisfied then the cycle is halted. Conversely if the condition is not satisfied, line 27 is executed and subsequently the cycle is once again halted. Thus it suffices to show that both these operations can be computed efficiently.

To check that $\mathbf{X} \in \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i} \cdot G$, given a base of this vector space (which can be computed efficiently from Claim 3.3.2) it suffices to verify that for each $\mathbf{v}$ in a base of the dual, it holds $\mathbf{v}^\top \mathbf{X} = 1$. Since the dual has dimension at most $n = n_1 + n_2$, this step can be performed using at most n^2 external multiplications and $n(n-1)$ additions.

Regarding the second instruction, when it is executed we have that $\mathbf{x} \in L$ by Claim 3.3.3 and $A \cdot L \subseteq \operatorname{Im} B + \mathbf{c}$ since the check at step 19 had to fail. In particular $\mathbf{Ax} \in \operatorname{Im} B + \mathbf{c}$. As shown in the proof of Claim 3.2.3, one can efficiently compute a vector $\mathbf{Y}_0$ such that $A\mathbf{X} + B\mathbf{Y}_0 = \mathbf{c} \cdot G$, and in particular this requires only polynomially many generic group operations.

To conclude we need to improve this solution in such a way that for all $i \in \{1, \ldots, \chi\}$ such that $\beta_i = 0$, $\mathbf{a}_i^\top \mathbf{X} + \mathbf{b}_i^\top \mathbf{Y}_0 \neq c_i$. To this aim call $I = \{i \leq \chi \, : \, \beta_i = 0\}$ and for all $i \in I$

$$W_i \; = \; \{\mathbf{y} \, : \, \mathbf{a}_i^\top \mathbf{x} + \mathbf{b_i}^\top = c_i\}.$$

First we observe that the solution for the system $A\mathbf{X} + B\mathbf{Y} = \mathbf{c} \cdot G$ is $\mathbf{Y}_0 + (\operatorname{Ker} B) \cdot G$. Indeed for any vector $\mathbf{Y}$, using the fact that $\mathbf{Y}_0$ is a solution too,

$$A\mathbf{X} + B\mathbf{Y} = \mathbf{c} \cdot G \quad \Leftrightarrow \quad B(\mathbf{Y} - \mathbf{Y}_0) = \mathbf{0} \quad \Leftrightarrow \quad \mathbf{Y} \in \mathbf{Y}_0 + (\operatorname{Ker} B) \cdot G.$$

Next, calling $\mathbf{y}_0$ the discrete logarithm of $\mathbf{Y}_0$, i.e. such that $\mathbf{Y}_0 = \mathbf{y}_0 \cdot G$, then for all $i \in I$ we prove that $\mathbf{y}_0 + \operatorname{Ker} B \not\subseteq W_i$. Assuming by contradiction that for some i this is not true,

then, since $\mathbf{y}_0 + \operatorname{Ker} B$ is the set of all solutions to the linear system defined by $A, B, \mathbf{c}$, we would have that for all $\mathbf{y}$

$$A\mathbf{x} + B\mathbf{y} = \mathbf{c} \quad \Rightarrow \quad \mathbf{y} \in W_i \quad \Rightarrow \quad \mathbf{a}_i^\top \mathbf{x} + \mathbf{b}_i^\top \mathbf{y} = c_i.$$

In particular $\mathbf{x} \in \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i}$ which implies $\mathbf{X} \in \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i} \cdot G$, which is a contradiction.

Observing now that W_i are hyperplanes and that $\operatorname{dir} W_i = \{\mathbf{y} \,:\, \mathbf{b}_i^\top \mathbf{y} = 0\}$ we have that either $W_i \cap (\mathbf{y}_0 + \operatorname{Ker} B) = \varnothing$ or $\operatorname{Ker} B \nsubseteq \operatorname{dir} W_i$. Let now $J \subseteq I$ be the set of indices i of those spaces such that $\mathbf{Y}_0 \in W_i$. If J were empty, $\mathbf{Y}_0$ would be the desired solution. Otherwise, for each of these spaces W_i, we can efficiently find $\mathbf{u}_i$ such that $\mathbf{u}_i \in \operatorname{Ker} B \setminus \operatorname{dim} W_i$, i.e. $\mathbf{u}_i \in \operatorname{Ker} B$ and $\mathbf{b}_i^\top \mathbf{u}_i \neq 0$.

We claim, but not prove immediately, that a vector $\mathbf{v}$ that is not orthogonal to any $\mathbf{b}_i$ for $i \in J$ and that lies in the span of $\mathbf{u}_{ii\in J}$ can be computed in polynomial time. Assuming for the moment that the latter is true, then we can conclude that the desired solution is one of the vectors in

$$\{\mathbf{Y}_0 + (\alpha \cdot \mathbf{v}) \cdot G \,:\, \alpha \in \{1, \dots, \chi + 1\}\}.$$

Indeed all these points (all distinct because $\chi - 1 < q$, the order of $\mathbb{G}$) lie on an affine line passing through $\mathbf{Y}_0$ and with direction $\mathbf{v}$. Since $\mathbf{u}_i \in \operatorname{Ker} B$, then also the vector $\mathbf{v} \in \operatorname{Ker} B$ as it belongs to the span of $\{\mathbf{u}_i\}_{i\in J}$, implying that the line (and in particular the set described above) is contained in $(\mathbf{y}_0 + \operatorname{Ker} B) \cdot G$. Next, by construction $\mathbf{Y}_0 \in W_i$ for $i \in J$ but $\mathbf{v} \notin \operatorname{dir} W_i$, we have that these hyperplanes intersect the line only in $\mathbf{Y}_0$. Conversely the hyperplanes W_i with $i \notin J$ by definition do not contain the point $\mathbf{Y}_0$ and in particular can intersect the line in at most 1 point. As the number of these spaces is at most χ, by the pigeonhole principle at least one among $\chi + 1$ points on the line does not belong in any of them. As checking membership in W_i can be done with polynomially many group operations as shown before, and χ is polynomially bounded, we can find a point $\mathbf{Y} \in \mathbf{Y}_0 + (\operatorname{Ker} B) \cdot G$ such that $\mathbf{Y} \notin W_i \cdot G$ for all $i \in I$. From the way we defined W_i we thus proved that

$$\begin{aligned} \mathbf{Y} \in \mathbf{Y}_0 + (\operatorname{Ker} B) \cdot G \quad &\Rightarrow \quad A\mathbf{X} + B\mathbf{Y} = \mathbf{c} \\ \mathbf{Y} \notin W_i \cdot G \quad &\Rightarrow \quad \mathbf{a}_i^\top \mathbf{X} + \mathbf{b}_i^\top \mathbf{Y} = c_i \cdot G. \end{aligned}$$

That is a solution to the given system.

Before concluding the proof of the claim we are left with showing an algorithm for computing $\mathbf{v}$ efficiently. This is presented if Figure 3.5. We show correctness by induction. If $n = 1$, $\mathbf{v} = \mathbf{u}_1$ is not orthogonal by hypothesis to $\mathbf{b}_1$ and trivially lies on the span of $\mathbf{u}_1$. Assuming that correctness holds for $n - 1$, then the intermediate vector $\mathbf{v}$ computed is not orthogonal to any $\mathbf{b}_1, \dots, \mathbf{b}_{n-1}$ and is a linear combination of $\mathbf{u}_1, \dots, \mathbf{u}_{n-1}$. If $\mathbf{v}$ is also non orthogonal to $\mathbf{b}_n$, then it satisfies the desired property. Conversely, we have that for all $\alpha \in \{0, 1\}$

$$(\alpha\mathbf{v} + \mathbf{u}_n)^\top \mathbf{b}_n \;=\; \mathbf{u}_n^\top \mathbf{b}_n \;\neq\; 0.$$

Regarding the other vector observe that for each i there can exists at most one $\alpha \in \mathbb{F}_q$ such that $(\alpha\mathbf{v} + \mathbf{u}_n)^\top \mathbf{b}_i = 0$, that is

$$\alpha \;=\; \frac{\mathbf{u}_n^\top \mathbf{b}_i}{\mathbf{v}^\top \mathbf{b}_i}.$$

$\mathsf{ExtPoint}(\mathbf{b}_1, \ldots, \mathbf{b}_n, \mathbf{u}_1, \ldots, \mathbf{u}_n)$:

1 : **If** $n = 1$: Return u_1
2 : **Else**:
3 : $\quad \mathbf{v} \leftarrow \mathsf{ExtPoint}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}, \mathbf{u}_1, \ldots, \mathbf{u}_n)$
4 : $\quad$ **If** $\mathbf{v}^\top \mathbf{u} \neq 0$: Return $\mathbf{v}$
5 : $\quad$ **Else**:
6 : $\quad\quad$ **For** $\alpha \in \{1, \ldots, n\}$:
7 : $\quad\quad\quad$ **If** $(\alpha \mathbf{v} + \mathbf{u}_n)^\perp \mathbf{b}_i = 0$ for all $i \in \{1, \ldots, n-1\}$:
8 : $\quad\quad\quad\quad$ Return $\alpha \mathbf{v} + \mathbf{u}_n$

Figure 3.5: ExtPoint, given $\mathbf{b}_i^\top \mathbf{u}_i \neq 0$ returns $\mathbf{v}$ a linear combination of $\mathbf{u}_i$ s.t. $\mathbf{b}_i^\top \mathbf{v} \neq 0$.

Since $n < q$ the values of α we consider are all distinct and by the pigeonhole there has then to exists an α such that $(\alpha \mathbf{v}^\top + \mathbf{u}_n)$ is not orthogonal to any $\mathbf{b}_i$ for $i < n$. In conclusion $(\alpha \mathbf{v}^\top + \mathbf{u}_n)$ is not orthogonal to any $\mathbf{b}_i$ and it is, by the inductive hypothesis, a linear combination of $\mathbf{u}_1, \ldots, \mathbf{u}_n$. This concludes the proof. □

Proof of Claim 3.3.5. First of all, if condition at step 23 is satisfied, then by construction the condition at step 21 is not (or step 23 is not executed). Hence $L \cap \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i}$ is a proper subspace of L, which implies that after step 24 the dimension of L decreases strictly.

If instead the condition at step 33 is satisfied, as done for the proof of Claim 3.3.6, we can assume with loss of generality that $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$ share the same labels of $\mathbf{X}$ and $\mathbf{Y}$. Executing $\mathsf{S.Vfy}$ on input $(\mathbf{X}, s, m, \mathbf{Y}, t)$ and calling $\beta_1, \ldots, \beta_\chi$ the bits returned by $\mathcal{O}^0_{\mathsf{eq}}$ invoked during this execution, we have that at least one of the conditions on steps 17, 19 or 21 is satisfied (since the other conditions would break the cycle). However

- If the check on step 17 is satisfied then $\mathsf{S.Vfy}$ on input $(\mathbf{X}, s, m, \mathbf{Y}, t)$ would output 0, meaning that the queried signature is rejected. This contradicts correctness of the underlying scheme.
- If the test on step 21 is satisfied, by Claim 3.3.3 $\mathbf{x} \in L$ and in particular $\mathbf{X} \in \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i}$ for some i such that $\beta_i = 0$. By construction this means that for any $\mathbf{Y}'$ that satisfies $A\mathbf{X} = B\mathbf{Y} + \mathbf{c} \cdot G$, then $\mathbf{a}_i^\top \mathbf{X} = \mathbf{b}_i^\top \mathbf{Y}' + c_i \cdot G$.

  As $A, B, \mathbf{c}$ consist of the linear constraints tested by $\mathsf{S.Vfy}$ that $\mathbf{X}$, $\mathbf{Y}$ and G verifies, then $A\mathbf{X} = B\mathbf{Y} + \mathbf{c} \cdot G$. In particular $\mathbf{a}_i^\top \mathbf{X} = \mathbf{b}_i^\top \mathbf{Y} + c_i \cdot G$, which contradicts the hypothesis that $\mathcal{O}^0_{\mathsf{eq}}(\,\cdot\,)$ returns $\beta_i = 0$ for the i-th query.

In conclusion we obtain that $A \cdot L \nsubseteq \operatorname{Im} B + \mathbf{c}$, implying that $\{\mathbf{x} \in L \,:\, A\mathbf{x} \in \operatorname{Im} B + \mathbf{c}\}$ is a proper subspace of L. Hence after step 33 the dimension of L strictly decreases. □

Proof of Claim 3.3.6. First of all we remark that step 27 can be efficiently computed since $A \cdot L \subseteq \operatorname{Im} B + \mathbf{c}$, which by Claim 3.3.3 implies that, calling $\mathbf{x}$ the discrete logarithm of $\mathbf{X}$, $A\mathbf{x} \in \operatorname{Im} B + \mathbf{c}$, and $\mathbf{x} \notin \mathcal{F}^{A,B,\mathbf{c}}_{\mathbf{a}_i,\mathbf{b}_i,c_i}$. In particular, calling V the affine space of solutions $\mathbf{y} \in \mathbb{F}_q^k$ such that $A\mathbf{x} = B\mathbf{y} + \mathbf{c}$, and V_i the subspace of V such that $\mathbf{a}_i^\top \mathbf{x} = \mathbf{b}_i^\top \mathbf{y} + c_i$ we have two

possible cases:

- $\dim V = 0$: then since condition on step 23 is not satisfied, $V_i \subsetneq V$, which means that $V_i = \varnothing$ for all i such that $\beta_i = 0$. In particular $\mathbf{Y}$ is the only solution to the system $A\mathbf{X} = B\mathbf{Y} + \mathbf{c} \cdot G$, which is obtained by

$$\mathbf{Y} = H \cdot (A\mathbf{X} - \mathbf{c} \cdot G)$$

  with H a left weak inverse of B, and it automatically satisfies all the conditions of the form $\mathbf{a}_i^\top \mathbf{X} \neq \mathbf{b}^\top \mathbf{Y} + c_i \cdot G$.

- $\dim V \geq 1$: then since condition on step 23 is not satisfied, $V_i \subsetneq V$ is a proper subspace of V. It follows that $\bigcup_{i:\beta_i=0} V_i$ contains at most $\chi \cdot q^{\dim V - 1}$ points. Since χ is polynomially bounded (or the verification algorithm would not be efficient) $\chi < q$ and in particular the set $V \setminus \bigcup_{i:\beta_i=0} V_i$ is not empty and a vector $\mathbf{Y}$ satisfying the above conditions can be obtained for instance with overwhelming probability greater than $1 - \chi \cdot q^{-1}$ by setting

$$\mathbf{Y} = H(A\mathbf{X} - \mathbf{c} \cdot G) + (I - HB)\mathbf{w} \cdot G$$

  with H a weak left inverse of B and $\mathbf{w} \leftarrow^{\$} \mathbb{F}_q^n$.

Finally to show that $(\mathbf{Y}, t)$ is a valid signature we first observe that without loss of generality we may assume that $\mathbf{X}$ and $\mathbf{Y}$ have the same labels of $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$. Indeed $\mathcal{A}$ can set $\widetilde{\mathbf{X}}$ with the same labels of $\mathbf{X}$ and $\widetilde{\mathbf{Y}}$ with large enough labels so that, for a given $\mathbf{Y} \in \mathbb{G}^k$, up to performing dummy operations (adding zeroes for instance), it is possible to obtain a new vector $\mathbf{Y}'$ representing the same group elements of $\mathbf{Y}$ and with the same labels of $\mathbf{Y}$.

As the inputs $(\mathbf{X}, s, m, \mathbf{Y}, t)$ and $(\widetilde{\mathbf{X}}, s, m, \widetilde{\mathbf{Y}}, t)$ are equal, the deterministic algorithm $\mathsf{S.Vfy}$ performs the same queries to $\mathcal{O}_{\mathsf{add}}$ and $\mathcal{O}^0_{\mathsf{eq}}$. In particular, for the i-th query, if $\beta_i = 0$ then $\mathbf{a}_i^\top \mathbf{X} \neq \mathbf{b}_i^\top \mathbf{Y} + c_i \cdot G$, meaning that $\mathcal{O}^0_{\mathsf{eq}}(\,\cdot\,)$ returns $0 = \beta_i$. Conversely if $\beta_i = 1$, then $\mathbf{a}^\top \mathbf{X} = \mathbf{b}^\top \mathbf{Y} + c_i \cdot G$, meaning that $\mathcal{O}^0_{\mathsf{eq}}(\,\cdot\,)$ returns $1 = \beta_i$. Hence the execution of $\mathsf{S.Vfy}$ produces the same output b it returns when executed with the simulated group $\widetilde{\mathbb{G}}$. As condition on step 17 is not satisfied, $b = 1$, meaning that the signature $(\mathbf{Y}, t)$ is valid for m. □

Proof of Claim 3.3.7. Initially $L = V \times \mathbb{F}_q^{n_2}$ implies $\dim L = n_2 + \dim V$. If $\mathcal{B}(\mathsf{vk}, V) \to (F, L)$ with $\pi(L) = V$ then $\dim L \geq \dim \pi(L) \geq \dim V$. □

Proof of Claim 3.3.8. Analogous to the proof of Claim 3.2.7. □

Proof of Claim 3.3.9. As in the proof of Claim 3.2.6, it can be shown that $\mathcal{A}$ executes its subroutine a polynomial number of times. By Claim 3.3.4 each execution requires a polynomial number of queries to the GGM and not other query is performed explicitly by $\mathcal{A}$. The thesis follows. □

Proof of Claim 3.3.10. Analogous to the proof of Claim 3.2.8. □

Chapter 4

Vector Commitments in Generic Groups

Chapter Overview

In this chapter we investigate Vector Commitments over Maurer's GGM. The main results are Corollary 4.3.3 and Theorem 4.3.4, proving lower bounds for the commitment and opening length of certain classes of VC in the GGM. More specifically, let ℓ_c and ℓ_π the commitment and maximum opening bit length respectively. The first result states that VC with generic verification (or *strictly linear*) satisfies $\ell_c \cdot \ell_\pi = \Omega(n)$ with n being the number of supported entries in the commitment. Conversely, for *hiding* VC (see Section 4.1.1), the stronger bound $\ell_c = \Omega(n)$ must hold. These results follow from the attacks developed in Chapter 3, and appeared in [32] and [63].

4.1 Definitions

4.1.1 Hiding Vector Commitments

In this section we provide a game-based definition for the *hiding* property of Vector Commitments. The approach we take is inspired by IND security for functional encryption schemes: for any two vectors $\mathbf{x}^0$, $\mathbf{x}^1$ provided by the adversary, we ask that guessing which one was committed is hard even when those positions in which $\mathbf{x}^0$ and $\mathbf{x}^1$ match are opened.

Definition 4.1.1. *Given a Vector Commitment and an adversary $\mathcal{A}$ we define its advantage at the hiding game, described in Fig. 4.1, as*

$$\mathsf{Adv}(\mathcal{A}) \;=\; \left|\frac{1}{2} - \Pr\left[\mathsf{ExpHide}^{\mathcal{A}}(1^\lambda) = 1\right]\right|.$$

A VC (resp. Algebraic VC) is hiding if there exists ε negligible such that for all PPT *(resp.* GPPT*) adversaries $\mathcal{A}$, $\mathsf{Adv}(\mathcal{A}) \leq \varepsilon(\lambda)$.*

As a sanity check we observe that combining any (non necessarily algebraic) VC with a commitment scheme yields an hiding VC as informally stated in [31]. We further notice that,

$\mathsf{ExpHide}^{\mathcal{A}}(1^\lambda)$

1 : $\mathsf{pp} \leftarrow \mathsf{VC.Setup}(1^\lambda), \quad \beta \leftarrow^{\$} \{0,1\}$
2 : $(\mathbf{x}^0, \mathbf{x}^1) \leftarrow \mathcal{A}(\mathsf{pp})$ such that $\mathbf{x}^0, \mathbf{x}^1 \in (\mathsf{VC.M})^n$
3 : $c, \mathsf{aux} \leftarrow \mathsf{VC.Com}(\mathsf{pp}, \mathbf{x}^\beta), \quad \mathcal{A} \leftarrow c$
4 : **When** $\mathcal{A}$ queries $i \in \{1, \dots, n\}$:
5 : **If** $x_i^0 = x_i^1$: $\Lambda_i \leftarrow \mathsf{VC.Open}(\mathsf{pp}, i, \mathsf{aux}), \quad \mathcal{A} \leftarrow \Lambda_i$
6 : **When** $\beta' \leftarrow \mathcal{A}$:
7 : Return $\beta' == \beta$

Figure 4.1: Vector Commitment's hiding game with adversary $\mathcal{A}$

viewing VCs as a special class of Functional Commitments [83], the game in Fig. 4.1.1 could be rephrased for this general primitive by letting $\mathcal{A}$ query functions f and receive an opening for f only if $f(\mathbf{x}^0) = f(\mathbf{x}^1)$.

While the above definition is given in a way that can be easily generalized, when applied to VC it becomes equivalent to a simpler notion, given through the game described in Fig. 4.2. The two main differences are that $\mathbf{x}^0$ and $\mathbf{x}^1$ are allowed to differ in at most one position, and that opening proofs for all other positions are given directly without oracle queries.

$\mathsf{ExpHideVC}^{\mathcal{A}}(1^\lambda)$

1 : $\mathsf{pp} \leftarrow \mathsf{VC.Setup}(1^\lambda), \quad \beta \leftarrow^{\$} \{0,1\}$
2 : $(\mathbf{x}^0, \mathbf{x}^1) \leftarrow \mathcal{A}(\mathsf{pp})$ with $\mathbf{x}^0, \mathbf{x}^1$ differing only in position i
3 : $c, \mathsf{aux} \leftarrow \mathsf{VC.Com}(\mathsf{pp}, \mathbf{x}^\beta), \quad \Lambda_j \leftarrow \mathsf{VC.Open}(\mathsf{pp}, j, \mathsf{aux})$ for all $j \neq i$.
4 : **When** $\beta' \leftarrow \mathcal{A}(c, (\Lambda_j)_{j \neq i})$
5 : Return $\beta' == \beta$

Figure 4.2: Simpler Vector Commitment's hiding game with adversary $\mathcal{A}$

Proposition 4.1.1. *A (resp. algebraic) VC is hiding if and only if there exists a negligible ε such that for each* PPT *(resp.* GPPT*) adversary $\mathcal{A}$, its advantage in the game described in Fig. 4.2 is*

$$\mathsf{Adv}(\mathcal{A}) \;=\; \left|\frac{1}{2} - \Pr\left[\mathsf{ExpHideVC}^{\mathcal{A}}(1^\lambda) = 1\right]\right| \;\leq\; \varepsilon(\lambda).$$

A proof of this Proposition appears in the full version of [63].

4.1.2 Strictly Linear and Generic Verification

As done for signatures over pairing-free groups, we study two variants of algebraic Vector Commitments. Namely those with *strictly linear* verification, for which looking ahead we will be able to provide an unconditional impossibility results, and those with *generic verification*, where all procedures are simply asked to be compatible with the GGM interface. We start by introducing a notion of algebraic vector commitments where the verification algorithm only consists of a system of linear equations.

Definition 4.1.2 (Algebraic VCs with linear verification)**.** *A vector commitment scheme is said to be* algebraic with linear verification *if the message space is* $\mathsf{VC.M} = \mathbb{F}_q$ *and*

- $\mathsf{VC.Setup}(1^\lambda) \xrightarrow{\$} \mathsf{pp}$ *such that* $\mathsf{pp} = (\mathbf{X}_1, s_1) \in \mathbb{G}^{\nu_1} \times \{0,1\}^*$.
- $\mathsf{VC.Com}(\mathsf{pp}, m_1, \ldots, m_n) \xrightarrow{\$} c, \mathsf{aux}$ *such that* $c = (\mathbf{X}_2, s_2) \in \mathbb{G}^{\nu_2} \times \{0,1\}^*$.
- $\mathsf{VC.Open}(\mathsf{pp}, m, i, \mathsf{aux}) \to \pi$ *such that* $\pi = (\mathbf{Y}, \mathbf{z})$ *with* $\mathbf{Y} \in \mathbb{G}^k$ *and* $\mathbf{z} \in \mathbb{F}_q^h$.
- *There exist* $A : \mathbb{F}_q^{h+1} \times [n] \times \{0,1\}^* \to \mathbb{F}_q^{\ell,n}$ *and* $B : \mathbb{F}_q^{h+1} \times [n] \times \{0,1\}^* \to \mathbb{F}_q^{\ell,k}$ *matrices such that* $\mathsf{VC.Vfy}(\mathsf{pp}, c, m, i, \pi) \to 1$ *if and only if, calling* $\mathbf{X} = \mathbf{X}_1 || \mathbf{X}_2$ *and* $s = s_1 || s_2$

$$A(\mathbf{z}, m, i, s) \cdot \mathbf{X} = B(\mathbf{z}, m, i, s) \cdot \mathbf{Y}.$$

For the ease of presentation we will omit s in A and B when clear from the context. Notice that the definition imposes linearity only with respect to group elements while it allows procedures A, B to depend non-linearly on the field vector element $\mathbf{z}$.

As we shall see, our first impossibility result states that whenever A is an affine function of $\mathbf{z}, m$ and B does not depends on $\mathbf{z}, m$, then the resulting scheme cannot be both "succinct" and position binding. We call these schemes *strictly* linear since their verification equations depend linearly both in $\mathbf{z}$ and $\mathbf{Y}$.

Definition 4.1.3 (Algebraic VCs with strictly linear verification)**.** *A vector commitment is said to be algebraic with strictly linear verification if it satisfies Definition 4.1.2,* $A(\mathbf{z}, m, i)$ *is an affine function*[1] *of* $\mathbf{z}, m$ *and* $B(i)$ *does not depends on* $\mathbf{z}, m$.

However, if we allow A to depend quadratically, or B linearly, on $\mathbf{z}, m$ then we could use arithmetization techniques, such as R1CS, to encode a circuit representing for example a Merkle tree verification into the verification equation of Definition 4.1.2. This means that we can construct algebraic VC schemes with linear verification that are succinct and position binding. Explicit examples of such schemes are provided in the full version of [32].

This technique however either bypasses the underlying group and may reduce security to external problems, or rely on non-black-box usage of the group. An example of the latter comes by encoding a Merkle tree built using an hash function whose collision resistance is based on discrete logarithm over the same group $\mathbb{G}$, such as Pedersen hash. Note that this construction would not retain algebraic properties from the underlying group. For this reason, following an approach similar to [93, 53], we study whether in the Generic Group Model (GGM) the security of a VC can be reduced to hard problems on the underlying group. To this aim we provide the following more general definition.

Definition 4.1.4 (Algebraic VCs with generic verification)**.** *A vector commitment scheme is said to be* algebraic with generic verification *if, in the GGM, the algorithms* $\mathsf{VC.Setup}, \mathsf{VC.Com}, \mathsf{VC.Open}, \mathsf{VC.Vfy}$ *are oracle machines with access to* $\mathcal{O}_{\mathsf{add}}$ *and* $\mathcal{O}^0_{\mathsf{eq}}$.

[1] i.e. $A(\mathbf{z}, m, i) = A_0(i) + z_1 A_1(i) + \ldots + z_h A_h(i) + m A_{h+1}(i)$

4.2 From Algebraic VC to Algebraic Signatures

4.2.1 Compiler

The strategy we adopt to show our impossibility results is to establish a connection between vector commitments and signatures, providing a way to construct the latter from the former generically. This way we will be able to bridge extensions of the impossibility results in [53] for algebraic signatures to algebraic vector commitments.

More specifically, for a given VC (not necessarily algebraic) our transformation produces a signature scheme with polynomially bounded message space $\{1, \dots, n\}$. The high-level idea is to compute a commitment c to random messages $m_1, \dots, m_n$, and use (pp, c) as the verification key and the auxiliary information aux as the secret key. In order to sign a message $i \in \{1, \dots, n\}$, the signer returns m_i and π, the message and opening proof for the i-th position, while verification is performed by checking the correctness of π. A formal description of the transformation is presented in Fig. 4.3.

$\mathsf{S}_{\mathsf{VC}}.\mathsf{Setup}(1^\lambda)$:

1 : $\mathsf{VC.Setup}(1^\lambda) \to \mathsf{pp}$
2 : $m_1, \dots, m_n \leftarrow^{\$} \mathsf{VC.M}$
3 : $c, \mathsf{aux} \leftarrow^{\$} \mathsf{VC.Com}(\mathsf{pp}, m_1, \dots, m_n)$
4 : $\mathsf{vk} \leftarrow (\mathsf{pp}, c)$ $\mathsf{sk} \leftarrow (\mathsf{aux}, \{m_i\}_{i=1}^n)$
5 : Return vk, sk

$\mathsf{S}_{\mathsf{VC}}.\mathsf{Sign}(\mathsf{sk}, i)$:

1 : Parse $\mathsf{sk} = (\mathsf{aux}, \{m_i\}_{i=1}^n)$
2 : $\pi \leftarrow \mathsf{VC.Open}(\mathsf{pp}, m_i, i, \mathsf{aux})$
3 : $\sigma \leftarrow (m_i, \pi)$
4 : Return σ

$\mathsf{S}_{\mathsf{VC}}.\mathsf{Vfy}(\mathsf{vk}, i, \sigma)$:

1 : Parse $\mathsf{vk} = (\mathsf{pp}, c)$ and $\sigma = (m_i, \pi)$. Return $\mathsf{VC.Vfy}(\mathsf{pp}, c, m_i, i, \pi)$

Figure 4.3: Generic transformation from VCs to signature schemes

In the rest of this section we will argue security according to the properties satisfied by the underlying VC. In particular we will show that succinct VC yields ϑ-unforgeable signatures (for a non-trivial function ϑ), while *hiding* VC yields unforgeable signatures.

4.2.2 ϑ-Unforgeability from Succinct VC

We now show that if the VC is somewhat succinct, then our compiler yields a ϑ-unforgeable signature. The high-level intuition is that openings (obtained through signing queries) may leak information on the committed values in other locations. However, the amount of information leaked cannot exceed the opening bit length. Hence, an adversary breaking unforgeability for *many* messages, either breaks position binding, or correctly guesses many information-theoretically hidden bits contained in the messages m_i in the forged positions i.

Theorem 4.2.1. *Given a Vector Commitment with commitments of bit-length $\ell_c = \ell_c(n, \lambda)$ and opening proofs of bit-length $\ell_\pi = \ell_\pi(n, \lambda)$, then there exists a* PPT *black box reduction $\mathcal{R}$ of ϑ-UF for the derived signature scheme described in Fig. 4.3 to the position binding property,*

where

$$\vartheta(\mathsf{vk}, Q) \;=\; \frac{\lambda + \ell_c + |Q| \cdot (\ell_\pi + \log|\mathsf{VC.M}|)}{\log|\mathsf{VC.M}|}.$$

In particular for any position binding VC, the resulting signature is ϑ-UF with ϑ as specified above.

Proof. Given an adversary $\mathcal{A}$ breaking ϑ-UF for this signature scheme, we provide a PPT algorithm $\mathcal{R}$ that breaks the underlying VC's position binding using $\mathcal{A}$ in a black-box way. For notational convenience let us denote $Q = \{(j, m_j, \pi_j)\}_{j \in S}$ the set of all queries recorded in the experiment 3.1 with S being the set of queried positions.

Adversary $\mathcal{R}(\mathsf{pp})$:

1 : Samples $m_1, \ldots, m_n \leftarrow^{\$} \mathsf{VC.M}$ and get $c, \mathsf{aux} \leftarrow^{\$} \mathsf{VC.Com}(\mathsf{pp}, m_1, \ldots, m_n)$
2 : Set $\mathsf{vk} \leftarrow (\mathsf{pp}, c)$, $S \leftarrow \varnothing$ and send $\mathcal{A} \leftarrow \mathsf{vk}$
3 : **When** $\mathcal{A}$ queries $j \in \{1, \ldots, n\}$:
4 : Compute the opening $\pi \leftarrow^{\$} \mathsf{VC.Open}(\mathsf{pp}, m_j, j, \mathsf{aux})$ and set $\sigma \leftarrow (m_j, \pi)$
5 : Update $S \leftarrow S \cup \{j\}$ and send $\mathcal{A} \leftarrow \sigma$
6 : **When** $\mathcal{A}$ returns a set F:
7 : **If** $\exists (i, \sigma) \in F \;:\; i \notin S \;\wedge\; \sigma = (\widehat{m}, \widehat{\pi}) \;\wedge\; \widehat{m} \neq m_i$:
8 : Compute $\pi_i \leftarrow \mathsf{VC.Open}(\mathsf{pp}, m_i, i, \mathsf{aux})$ and return $(c, m_i, \widehat{m}, i, \pi_i, \widehat{\pi})$
9 : **Else**: Return $\perp$

Figure 4.4: Reduction $\mathcal{R}$ breaking position binding.

As a first step we argue that with overwhelming probability, if $\mathcal{A}$ produces more than ϑ forgeries, then at least one them contains an opening to a message that differs from the committed one.

Claim 4.2.1. *Let $F = \{(i, \widehat{m}_i, \widehat{\pi}_i)\}_{i \in I}$ be the set of forgeries $\mathcal{A}$ produces. If $|I| \geq \vartheta(\mathsf{vk}, Q)$ and $I \cap S = \varnothing$, calling* bad *the event $\widehat{m}_i = m_i$, $\forall i \in I$, then* $\Pr[\mathsf{bad}] \leq 2^{-\lambda}$.

Proof of Claim 4.2.1. $\{m_i\}_{i \in I}$ are uniformly distributed and independent from pp, but may not be independent from c and $\{m_j, \pi_j\}_{j \in S}$. Notice that, by the way we defined ℓ_c and ℓ_π, we have that

$$(c, \{\pi_j\}_{j \in S}, \{m_j\}_{j \in S}) \in \{0,1\}^{\ell_c + |S|\ell_\pi} \times \mathsf{VC.M}^{|S|}.$$

We can then lower bound the conditional min-entropy as $\mathrm{H}_\infty(\{m_i\}_{i \in I} \,|\, c,\, \{m_j, \pi_j\}_{j \in S}) \geq$

$$\begin{aligned}
&\geq\; \mathrm{H}_\infty(\{m_i\}_{i \in I}) - \ell_c - |S| \cdot \ell_\pi - |S| \cdot \log|\mathsf{VC.M}| \\
&=\; |I| \cdot \log|\mathsf{VC.M}| - \ell_c - |S| \cdot (\ell_\pi + \log|\mathsf{VC.M}|) \\
&\geq\; \vartheta(\mathsf{vk}, Q) \cdot \log|\mathsf{VC.M}| - \ell_c - |Q| \cdot (\ell_\pi + \log|\mathsf{VC.M}|) \\
&=\; \lambda
\end{aligned}$$

where the first equality follows as, without conditioning on c, π_j, m_j, $m_{i i \in I}$ and I are independent, the second inequality uses $|I| \geq \vartheta(\mathsf{vk}, Q)$ and $|S| = |Q|$ while the last equality applies

our assumption on $\vartheta(\mathsf{vk}, Q)$. In conclusion, the probability of correctly guessing $\{m_i\}_{i\in I}$ given c and $\{m_j, \pi_j\}_{j\in S}$ is smaller than $2^{-\lambda}$, proving the claim. □

To conclude the Theorem's proof, if $\mathcal{A}$ wins in game 3.1 then all the forgeries returned are correct signatures of non-queried messages. In particular if bad does not occur, among the forgeries returned by $\mathcal{A}$, $\mathcal{R}$ will find $\widehat{m}_i \neq m_i$ and $\widehat{\pi}_i$ such that

$$\mathsf{VC.Vfy}(\mathsf{pp}, c, m_i, i, \pi_i) \to 1 \qquad \mathsf{VC.Vfy}(\mathsf{pp}, c, \widehat{m}_i, i, \widehat{\pi}_i) \to 1.$$

which breaks the position binding. Therefore $\mathsf{Adv}(\mathcal{A}) \leq \mathsf{Adv}(\mathcal{R}) + 2^{-\lambda}$ which is negligible. □

4.2.3 Unforgeability from Hiding VC

We now turn our attention to *hiding* VC schemes. The key idea is that the information-theoretic argument in the previous section could have been avoided if commitments and openings do not leak information on the committed values. This suggests the following (informal) proof strategy for unforgeability: Assume an adversary forges a signature for message i, that is an opening of c at position i to some message m'_i, with c being a commitment to $m_1, \ldots, m_n$. Then $m'_i \neq m_i$ only with negligible probability, or else $\mathcal{A}$ would break position binding. Therefore $\mathcal{A}$ can be used to break the hiding property. This is done by guessing the position i it will forge, querying in the hiding game, Figure 4.2, two random vectors differing in that position, answering signature queries with the received opening values and finally returning the bit corresponding to the vector containing m'_i in position i.

Proposition 4.2.2. *Given a position-binding and hiding (Algebraic) Vector Commitment, the (Algebraic) Signature scheme in Fig. 4.3 is unforgeable.*

Proof of Proposition 4.2.2. Given an adversary $\mathcal{A}$ breaking unforgeability of the signature scheme described in Fig. 4.3, we build in Figure 4.5 an adversary $\mathcal{B}$ playing against the (simpler) hiding game for Vector Commitment described in Fig. 4.2.

We first state the following claim, where b is the challenge bit chosen by $\mathcal{B}$'s challenger and forge the event $1 \leftarrow \mathsf{VC.Vfy}(\mathsf{pp}, c, x^*_k, k, \Lambda^*_k)$.

Claim 4.2.2. *Exists ε negligible such that* $\Pr[\mathsf{forge},\ k = i,\ x^*_k \neq x^0_i \mid b = 0] \leq \varepsilon$.

These claims implies the thesis since

$$\begin{aligned}
2 \cdot \mathsf{Adv}(\mathcal{B}) &= |\Pr[\mathcal{B} \to 0 \mid b = 0] - \Pr[\mathcal{B} \to 0 \mid b = 1]| \\
&\geq \Pr[\mathcal{B} \to 0 \mid b = 0] \\
&\geq \Pr\left[\mathsf{forge},\ k = i,\ x^*_k = x^0_i \;\middle|\; b = 0\right] \\
&\geq \Pr[\mathsf{forge},\ k = i \mid b = 0] - \Pr\left[\mathsf{forge},\ k = i,\ x^*_k \neq x^0_i \;\middle|\; b = 0\right] \\
&\geq \Pr[\mathsf{forge} \mid b = 0,\ k = i] \Pr[k = i \mid b = 0] - \varepsilon \\
&\geq \mathsf{Adv}(\mathcal{A}) \cdot \frac{1}{n} - \varepsilon
\end{aligned}$$

$\mathcal{B}(\mathsf{pp})$:

1 : Guess the index $\mathcal{A}$ is going to forge, $i \leftarrow^{\$} [n]$
2 : Sample $x_i^0, x_i^1 \leftarrow^{\$} \mathsf{VC.M}$ with $x_i^0 \neq x_i^1$
3 : For all $j \neq i$: $x_j \leftarrow^{\$} \mathsf{VC.M}$, and set $x_j^0 \leftarrow x_j$, $x_j^1 \leftarrow x_j$
4 : Query $\mathbf{x}^0, \mathbf{x}^1$ and wait for $(c, (\Lambda_j)_{j \neq i})$
5 : Run $\mathcal{A}(\mathsf{pp}, c)$
6 : **When** $\mathcal{A}$ queries j:
7 : **If** $j = i$: Return 1 // i.e. abort
8 : **Else**: $\mathcal{A} \leftarrow (x_j, \Lambda_j)$
9 : **When** $\mathcal{A}$ returns (k, x_k^*, Λ_k^*)
10 : **If** $k \neq i$ or $0 \leftarrow \mathsf{VC.Vfy}(\mathsf{pp}, c, x_k^*, k, \Lambda_k^*)$ or $x_k^* = x_i^1$:
11 : Return 1
12 : **Else**: Return 0

Figure 4.5: Reduction $\mathcal{B}$ executed in the hiding game of Fig. 4.2.

where we used the fact that $\mathcal{A}$ has no information on i, thus $\Pr[k = i] = 1/n$ and forge is independent on $k = i$.

Thus $\mathsf{Adv}(\mathcal{A}) \leq 2n\mathsf{Adv}(\mathcal{B}) + n\varepsilon$, that is negligible. Next we prove the Claim providing a reduction $\mathcal{C}$ that uses $\mathcal{A}$ to break position binding. The idea is that $\mathcal{C}$ can emulate the behavior of $\mathcal{B}$ when $b = 0$ until the final step. If the adversary manages to produce an opening to a message x_k^* for the right position $k = i$ but with $x_k^* \neq x_i^0$, then $\mathcal{C}$ breaks position binding returning this opening for x_k^*, and the correct one he can generate for x_i^0. A full description appears in Figure 4.6.

$\mathcal{C}(\mathsf{pp})$:

1 : // Simulate $\mathcal{B}$ and its challenger
2 : Sample $i \leftarrow^{\$} [n]$ and $\mathbf{x} \leftarrow^{\$} \mathsf{VC.M}^n$
3 : $(c, \mathsf{aux}) \leftarrow \mathsf{VC.Com}(\mathsf{pp}, \mathbf{x})$
4 : Run $\mathcal{A}(\mathsf{pp}, c)$
5 : **When** $\mathcal{A}$ queries j:
6 : **If** $j = i$: Return $\perp$
7 : **Else**: $\pi_j \leftarrow \mathsf{VC.Open}(\mathsf{pp}, j, \mathsf{aux})$, $\mathcal{A} \leftarrow (x_i, \Lambda_i)$
8 : **When** $\mathcal{A}$ return (k, x_k^*, Λ_k^*)
9 : **If** $k \neq i$ and $\mathsf{VC.Vfy}(\mathsf{pp}, c, x_k^*, k, \Lambda_k)$ and $x_k^* \neq x_i$:
10 : $\Lambda_i \leftarrow \mathsf{VC.Open}(\mathsf{pp}, i, \mathsf{aux})$
11 : Return $(c, i, x_k^*, \Lambda_k^*, x_i, \Lambda_i)$
12 : **Else**: Return $\perp$

Figure 4.6: Reduction $\mathcal{C}$ breaking position binding.

By inspection $\mathcal{C}$ simulates $\mathcal{B}$ when $b = 0$ and $\mathbf{x} = \mathbf{x}^0$ correctly. Note that there is no need to simulate $\mathbf{x}^1$ as when $b = 0$, $\mathcal{A}$ gets no information about it. Finally, if the condition at step 9 is executed, $\mathcal{C}$ breaks position binding correctly since $x_k^* \neq x_i$, $1 \leftarrow \mathsf{VC.Vfy}(\mathsf{pp}, c, x_k^*, k, \Lambda_k^*)$ and, by correctness, $1 \leftarrow \mathsf{VC.Vfy}(\mathsf{pp}, c, x_i, i, \Lambda_i)$ with $i = k$. Hence

$$\mathsf{Adv}(\mathcal{C}) = \Pr\left[\mathsf{forge},\ k = i,\ x_k^* \neq x_i^0 \ \middle|\ b = 0\right].$$

which proves the claim. □

4.3 Bounds for Algebraic Vector Commitments

In this section we combine the negative results for algebraic signature with the security achieved by our generic compiler starting from any (hiding) VC. We derive in this way lower bounds for algebraic vector commitments. In particular, for general VC in the GGM, we prove that $\ell_c \cdot \ell_\pi = \Omega(n)$ where ℓ_c, ℓ_π are respectively the commitment and opening bit length, and n is the committed vector's length. For hiding VC instead we get a stronger bound, namely $\ell_c = \Omega(n)$.

4.3.1 Bounds for Strictly Linear and Generic Verification VC

Theorem 4.3.1. *Given a position binding algebraic VC with strictly linear verification, let $\ell_c = \ell_c(n)$ and $\ell_\pi = \ell_\pi(n)$ be respectively the commitment and opening bit length to commit to a vector of n entries. Then*

$$\nu_2 + \frac{\lambda + \ell_c + \nu_2 \cdot (\ell_\pi + \log|\mathsf{VC.M}|)}{\log|\mathsf{VC.M}|} \ \geq\ n.$$

Proof. Assume there exists an algebraic VC with strictly linear verification contradicting the above inequality and satisfying position binding. Then by Theorem 4.2.1 the signature scheme obtained through the transformation in Fig. 4.3 would satisfy ϑ-UF with

$$\vartheta(\mathsf{vk}, Q) \ =\ \frac{\lambda + \ell_c + |Q| \cdot (\ell_\pi + \log|\mathsf{VC.M}|)}{\log|\mathsf{VC.M}|}$$

and its message space would have size $|\mathsf{S_{VC}.M}| = n$. Since vk contains ν_2 group elements excluding those that belong to the CRS, i.e. the public parameters of the original Vector Commitment, the attacker $\mathcal{A}$ from Theorem 3.2.1 can produce at least $n - \nu_2$ forgeries performing at most ν_2 queries. Called Q the set of queries performed by $\mathcal{A}$ we would have that

$$\vartheta(\mathsf{vk}, Q) \ \leq\ \frac{\lambda + \ell_c + \nu_2 \cdot (\ell_\pi + \log|\mathsf{VC.M}|)}{\log|\mathsf{VC.M}|} \ <\ n - \nu_2$$

where we use the fact that $|Q| \leq \nu_2$ in the first inequality. This is then a contradiction since $\mathcal{A}$ would breaks the ϑ-UF of the derived signature, implying that the given vector commitment was not binding. □

Theorem 4.3.2. *Given an algebraic VC with generic verification that is position binding against unbounded adversaries performing polynomially bounded queries to the GGM oracles $\mathcal{O}_{\mathsf{add}}$, $\mathcal{O}^0_{\mathsf{eq}}$, using the same notation of Theorem 4.3.1, then*

$$\nu_2 + \frac{\lambda + \ell_c + \nu_2 \cdot (\ell_\pi + \log |\mathsf{VC.M}|)}{\log |\mathsf{VC.M}|} \;\geq\; n.$$

Proof. Assuming again by contradiction that the above inequality is not satisfied, Theorem 4.2.1 implies that the associated signature scheme is ϑ-UF against any unbounded adversary $\mathcal{C}$ making at most polynomially many signature and group operations queries, or otherwise $\mathcal{R}^{\mathcal{C}}$ would break position binding with significant advantage. Notice that since $\mathcal{R}$ is PPT, $\mathcal{R}^{\mathcal{C}}$ still performs polynomially many generic group operations. As in the proof of Theorem 4.3.1 then, our initial assumption implies $\vartheta \leq n - \nu_2$. Since the adversary $\mathcal{A}$ of Theorem 3.3.1 returns $n - \nu_2$ signatures performing at most ν_2 queries, this contradicts the ϑ-UF of the associated signature against this adversary. □

Corollary 4.3.3. *Given an algebraic vector commitment with strictly linear verification, then $\ell_c \cdot \ell_\pi = \Omega(n)$. Analogously, given an algebraic vector commitment with generic verification position binding against unbounded adversary performing at most polynomially many queries to the GGM oracles, $\ell_c \cdot \ell_\pi = \Omega(n)$.*

Note that this lower bound implies in both cases that either $\ell_c = \Omega(\sqrt{n})$ or $\ell_\pi = \Omega(\sqrt{n})$.

4.3.2 Bounds for Hiding VC

In Section 3.3 we proved that any algebraic signature satisfying ϑ unforgeability with message space of size n and a verification key with m group elements[2] must satisfy $m \geq n + \vartheta$. Recall that, as mentioned, the standard Unforgeability notion is equivalent to ϑ-unforgeability for $\vartheta = 0$. Moreover, in the reduction provided in Fig. 4.3, the verification key only consist of the public parameters (that can be given in the CRS) and one commitment. We thus conclude that

Theorem 4.3.4. *Any position-binding and hiding Algebraic Vector Commitment with* GPPT *computable procedures, whose commitment for a vector of length n contains $\ell_c = \ell_c(\lambda, n)$ group elements, satisfies $\ell_c \geq n$.*

Proof. According to Proposition 4.2.2, any hiding VC for vectors of length n yields an unforgeable signatures with message space of size n. Let ν_2 be the number of group elements in the commitment. If by contradiction $\nu_2 < n$, then Theorem 4.3.2 implies there exists an adversary $\mathcal{A}$ finding at least one forgery for the corresponding signature scheme. Thus $\nu_2 \geq n$ and in particular $\ell_c \geq \nu_2$ implies $\ell_c = \Omega(n)$. □

Notice that an Algebraic VC that is both position-binding and hiding with linear opening proof size and constant commitment size would violate this Theorem, but not the bound in the previous section. Looking ahead this will prove useful when studying Algebraic NIZK-AoK

[2]Excluding those group elements contained in the CRS for which the signer has no trapdoor information.

in the GGM for certain relations. We finally remark that the above Theorem also captures VC scheme that are efficient only with respect to group operations. This follows as the attack presented in Section 3.3 also works in such case (i.e. when the signature is efficient only in terms of group operations).

Chapter 5

Non-Interactive Zero-Knowledge in Generic Groups

Chapter Overview

In this chapter we continue our study focusing on Non-Interactive Zero-Knowledge arguments in Maurer's GGM. In this setting two classes of construction are known from prime order groups. The first one consists of algebraic constructions in pairing groups, e.g. Groth-Sahai proofs [72]. The second one instead only requires pairing-free groups, but heavily relies on the group element representation by instantiating Fiat-Shamir with a correlation intractable hash [25, 75].

Eventually we show that *best of both worlds* constructions do not exists. More specifically the two main results of this chapter, presented in Theorem 5.2.2 and 5.3.2, states that in Maurer GGM is impossible to construct:

- Non-Interactive Zero-Knowledge arguments of knowledge for the *preimage relation* $\mathcal{R} = \{(x, w) : f(w) = x\}$ where f is one-way in the GGM (with mild limitations). This include proving knowledge of a discrete logarithm.
- Non-Interactive Zero-Knowledge arguments, for hard subset-membership problems, i.e. languages $\mathcal{L}$ where is possible to sample indistinguishably (against a GPPT distinguisher) from $\mathcal{L}$ and its complement. This notably include the DDH problem.

Note the two results are independent. The first cannot imply the second one because arguments of knowledge are a special type of NIZK. Conversely the second one cannot imply the first one since the latter also holds for relations with trivial languages, i.e. where all instances have a witness, such as the discrete-logarithm relation. Our fist result follows by showing how such NIZKs could be used to build Hiding Pedersen-like VC violating the lower bound presented in Chapter 4. The second one instead is based on an adaptation of the attack on signatures given in Section 3.3.

5.1 One Way Functions in Maurer GGM

5.1.1 Definition

In this section we provide definitions that allow us to capture one way functions that only uses a black-box group and whose hardness reduces only to hard problems in the group. The first notion is easily captured by assuming f access the group through the oracles $\mathcal{O}_{\mathsf{add}}$ and $\mathcal{O}_{\mathsf{eq}}$. To capture the second one we follow the approach of [53] used in previous chapters where security is provided against all GPPT adversaries, i.e. unbounded machines restricted to perform a polynomially bounded number of queries to the GGM random oracles.

Definition 5.1.1. *We define Algebraic OWF Family a couple* (Gen, f) *of* PPT *algorithms with*

$$k \leftarrow^{\$} \mathsf{Gen}(1^\lambda), \qquad f_k : \{0,1\}^{n_1} \times \mathbb{G}^{n_2} \to \{0,1\}^{m_1} \times \mathbb{G}^{m_2}.$$

such that for all GPPT *adversaries* $\mathcal{A}$ *there exists a negligible* ε *such that*

$$\Pr\left[\mathcal{A}(y) \to z, \;\; f_k(z) = y \;\middle|\; x \leftarrow^{\$} \{0,1\}^{n_1} \times \mathbb{G}^{n_2}, \;\; y \leftarrow f(x)\right] \leq \varepsilon(\lambda).$$

The simplest example of algebraic OWF is the group exponentiation $f : \mathbb{F}_q \to \mathbb{G}$ with $f(x) = x \cdot G$, whose hardness in the GGM was proved in [88].

Without loss of generality we can assume that f_k outputs only group elements, as the output bits can be encoded *in the exponent*. More precisely given f as in the definition above we can define for each key k

$$f'_k : \{0,1\}^{n_1} \times \mathbb{G}^{n_2} \to \mathbb{G}^{m_1+m_2} \;\; : \;\; f_k(x) = ((b_i)_{i=1}^{m_1}, \mathbf{H}) \;\; \Rightarrow \;\; f'_k(x) = ((b_i \cdot G)_{i=1}^{m_1}, \mathbf{H}).$$

In the GGM $f'(x)$ can be computed from $f(x)$ and vice versa.

5.1.2 Collision Resistance

Our first impossibility result for NIZK-AoK will apply to NP relations defined for a large family of OWF, but we will need the OWF to be collision resistant. This is problematic as not every OWF is collision resistant, effectively restricting the scope of our result.

To address this issue we show that any *algebraic* OWF family with domain $\{0,1\}^n$ and Gen returning only random group elements can be transformed to achieve collision resistance by simply restricting its domain. The idea is that, with unbounded computation, we could find for a given key k a subset $\mathcal{X} \subseteq \{0,1\}^n$ such that $f_k(\mathcal{X}) = \operatorname{Im} f_k$ and f_k is injective over $\mathcal{X}$. However this would be inefficient in terms of group operations[1]. Therefore we show that if the OWF key is a vector of random group elements, it is possible to restrict the function's domain in GPPT time so that finding collisions implies finding linear relations among the group elements in the key.

Concretely in the following lemma we provide two GPPT algorithm, Memb and Samp, respectively testing membership with the restricted domain $x \in \mathcal{X}$ and sampling elements from $\mathcal{X}$.

[1]Each evaluation of f_k would required access to the GGM, implying exponentially many queries.

Lemma 5.1.1. *Given* (Gen, f) *algebraic OWF family with* $f : \{0,1\}^n \to \mathbb{G}^m$ *and* Gen *returning a uniformly distributed key* $k \sim U(\mathbb{G}^\kappa)$*, there exists a set* $\mathcal{X} \subseteq \{0,1\}^n$ *and two* GPPT *algorithm* Memb *and* Samp *such that*

- ***Correctness 1****:* $\mathsf{Memb}(x) \to 1 \;\Leftrightarrow\; x \in \mathcal{X}$.
- ***Correctness 2****:* $x \leftarrow^{\$} \mathsf{Samp}(1^\lambda) \;\Rightarrow\; x \in \mathcal{X}$.
- ***Indistinguishability****:* $\exists \varepsilon$ *negligible s.t. for* $k \leftarrow^{\$} \mathsf{Gen}(1^\lambda)$, $x_1 \leftarrow^{\$} \{0,1\}^n$ *and* $x_2 \leftarrow^{\$} \mathsf{Samp}(1^\lambda)$,
$$\Delta\Big((k, f_k(x_1)), (k, f_k(x_2))\Big) \;\leq\; \varepsilon(\lambda).$$
- ***Collision Resistance****:* $\exists \varepsilon$ *negligible s.t. for all* GPPT *adversaries* $\mathcal{A}$ *, given* $k \leftarrow^{\$} \mathsf{Gen}(1^\lambda)$,
$$\Pr\left[\mathcal{A}(k) \to (x_1, x_2),\;\; x_1, x_2 \in \mathcal{X},\;\; x_1 \neq x_2,\;\; f_k(x_1) = f_k(x_2)\right] \leq \varepsilon(\lambda).$$

We remark that the above Lemma could be extended to $k \sim \mathbb{G}^\kappa$ (not necessarily uniform) such that finding non-trivial linear relations among its group elements is hard for GPPT adversary. This holds as in the GGM such a vector k would be indistinguishable from $k' \sim U(\mathbb{G}^\kappa)$, implying that f can be extended to f' with key space $\mathbb{G}^\kappa$ by running the evaluation algorithm for f also for those keys for which f is not formally defined. Since Memb and Samp exists for f', they satisfy the above properties also for f (or else we could build a distinguisher for k and k').

Proof of Lemma 5.1.1. We will describe Memb and Samp by first providing a GPPT algorithm $\mathcal{T}$ that generates an exponentially long explicit description of the set $\mathcal{X}$ along with a probability distribution. Then Memb and Samp can check membership in $\mathcal{X}$ and sample from it without using the group at all.

In order to describe $\mathcal{T}$ we make the following preliminary observation. Given any deterministic and polynomial time oracle Turing Machine $\mathcal{F}$ that computes $f_{\mathbf{k}}(x)$ for $\mathbf{k} \in \mathbb{G}^\kappa$ and $x \in \{0,1\}^n$, there exists an extractor $\mathcal{E}$ that on input $\mathbf{k}, x$ returns a matrix A such that $f_{\mathbf{k}}(x) = A\mathbf{k}$. The idea is simply that $\mathcal{E}$ executes $\mathcal{F}$ forwarding all the queries to the GGM's oracles while keeping a representation of each queried element in base $\mathbf{k}$. Although there may be many matrices A such that $f_{\mathbf{k}}(x) = A\mathbf{k}$, since $\mathcal{F}$ is deterministic, so is $\mathcal{E}$, implying that there is only one such matrix returned on input $(\mathbf{k}, x)$. We can thus define

$$A_{\mathbf{k},x} \in \mathbb{F}_q^{m,\kappa} \quad : \quad \mathcal{E}(\mathbf{k}, x) \to A_{\mathbf{k},x}, \quad f_{\mathbf{k}}(x) = A_{\mathbf{k},x} \cdot \mathbf{k}.$$

The above definition works when $\mathbb{G}$ is a prime order group and $\mathbf{k}$ is uniformly sampled from $\mathbb{G}^\kappa$. However in the following we will use a different generic group modeling $\mathbb{G}^\kappa$ instead of $\mathbb{G}$ where $\mathbf{k}^* = (e_1, \ldots, e_\kappa) \in (\mathbb{G}^\kappa)^\kappa$, with $e_1, \ldots, e_\kappa$ being the canonical base of $\mathbb{G}^\kappa$. This will be done to create an environment indistinguishable from the standard one (where the GGM models $\mathbb{G}$) in which the group elements in the key satisfy no linear relation. By executing $\mathcal{E}$ with this GGM oracle modeling $\mathbb{G}^\kappa$ we define for all $x \in \{0,1\}^n$

$$A^*_x \in \mathbb{F}_q^{m,\kappa} \quad : \quad \mathcal{E}(\mathbf{k}^*, x) \to A^*_x.$$

We are now ready to provide a description of $\mathcal{T}$. The idea is that computing A_x^* does not require any real GGM oracle query because $\mathcal{T}$ can simulate a GGM in which any non trivial linear relation query among the elements in $\mathbf{k}$ is answered with 0. Hence $\mathcal{T}$ can compute A_x^* for all x and insert in $\mathcal{X}$ only those element whose associated matrix does not collide with other elements already in $\mathcal{X}$. In this way we have that the map

$$\mathcal{X} \to \mathbb{F}_q^{m,\kappa} \quad : \quad x \mapsto A_x^*$$

is injective. Furthermore $\mathcal{T}$ can compute in exponential space a map $F : \{0,1\}^n \to \mathcal{X}$ such that x and $F(x)$ have the same associated matrix. This defines a distribution over $\mathcal{X}$ that is the image through F of the uniform one over $\{0,1\}^n$. A full description of $\mathcal{T}$ is provided in Figure 5.1, along with a Memb and Samp.

Procedure $\mathcal{T}(1^\lambda)$:

Initialize $\mathcal{X} \leftarrow \emptyset$ and $F : 0, 1^n \to \mathcal{X}$ partial function
For $x \in \{0,1\}^n$:
  Compute $A_x^* \leftarrow \mathcal{E}(\mathbf{k}^*, x)$
  If there exists $z \in \mathcal{X}$ with $A_z^* = A_x^*$:
    Set $F(x) \leftarrow z$
  Else:
    Add $\mathcal{X} \leftarrow \mathcal{X} \cup \{x\}$ and set $F(x) = x$
Return $(\mathcal{X}, F)$

$\mathsf{Memb}(1^\lambda, x)$

Run $(\mathcal{X}, F) \leftarrow \mathcal{T}$
If $x \in \mathcal{X}$: Return 1
Else: Return 0

$\mathsf{Samp}(1^\lambda)$

Run $(\mathcal{X}, F) \leftarrow \mathcal{T}$
Sample $x \leftarrow^{\$} \{0,1\}^n$
Else: Return $F(x)$

Figure 5.1: Intermediate procedure $\mathcal{T}$ describing $\mathcal{X}$.

Note that $\mathcal{X}$ is well defined because $\mathcal{T}$ is deterministic, although it depends on the arbitrary choice of an ordering in $\{0,1\}^n$ and a procedure $\mathcal{F}$ to compute f, used to construct $\mathcal{E}$. Having defined $\mathcal{X}$ we can now state and prove the following claims, which trivially imply the thesis.

Claim 5.1.1. *Both correctness properties holds. I.e. for all $x \in \{0,1\}^n$, $1 \leftarrow \mathsf{Memb}(x)$ iff $x \in \mathcal{X}$ and $x \leftarrow \mathsf{Samp}(1^\lambda)$ implies $x \in \mathcal{X}$.*

Claim 5.1.2. *There exists a negligible ε such that for any distribution $\mathcal{D}$ over $\{0,1\}^n$, given $\mathbf{k} \leftarrow^{\$} \mathsf{Gen}(1^\lambda)$ and $x \leftarrow^{\$} \mathcal{D}$,*

$$\Pr\left[A_x^* \neq A_{\mathbf{k},x}\right] \leq \varepsilon(\lambda).$$

Claim 5.1.3. *There exists a negligible ε such that for any distribution $\mathcal{D}$ over $\{0,1\}^n$, given $\mathbf{k} \leftarrow^{\$} \mathsf{Gen}(1^\lambda)$ and $x \leftarrow^{\$} \mathcal{D}$,*

$$\Delta((\mathbf{k}, A_x^*), (\mathbf{k}, A_{\mathbf{k},x})) \leq \varepsilon(\lambda).$$

Claim 5.1.4. *Given* $\mathbf{k} \leftarrow \mathsf{Gen}(1^\lambda)$, $x_1 \leftarrow^\$ \{0,1\}^n$ *and* $x_2 \leftarrow^\$ \mathsf{Samp}(1^\lambda)$ *then*

$$\Delta((\mathbf{k}, A^*_{x_1}), (\mathbf{k}, A^*_{x_2})) \;=\; 0.$$

Claim 5.1.5. *Given* $\mathbf{k} \leftarrow \mathsf{Gen}(1^\lambda)$, $x_1 \leftarrow^\$ \{0,1\}^n$ *and* $x_2 \leftarrow^\$ \mathsf{Samp}(1^\lambda)$ *then*

$$\Delta((\mathbf{k}, f_{\mathbf{k}}(x_1)), (\mathbf{k}, f_{\mathbf{k}}(x_2))) \;\leq\; \varepsilon(\lambda).$$

Claim 5.1.6. $\exists \varepsilon$ *negligible s.t. for all* GPPT *adversaries* $\mathcal{A}$ *, given* $\mathbf{k} \leftarrow^\$ \mathsf{Gen}(1^\lambda)$,

$$\Pr\left[\mathcal{A}(\mathbf{k}) \to (x_1, x_2),\;\; x_1, x_2 \in \mathcal{X},\;\; x_1 \neq x_2,\;\; f_{\mathbf{k}}(x_1) = f_{\mathbf{k}}(x_2)\right] \leq \varepsilon(\lambda).$$

□

Proof of Claim 5.1.1. By inspection $\mathsf{Memb}(x)$ returns 1 if and only if $x \in \mathcal{X}$ and is a GPPT algorithm since both $\mathcal{T}$ is, as no GGM query is ever performed. Analogously by construction $F : \{0,1\}^n \to \mathcal{X}$, so $\mathsf{Samp}(1^\lambda)$ always returns an element in $\mathcal{X}$. □

Proof of Claim 5.1.2. Given a distribution $\mathcal{D}$, even if not efficiently sampleable, we will build an adversary $\mathcal{A}$ who tries to find a linear relation among κ group elements. The idea is that A^*_x differs from $A_{\mathbf{k},x}$ only if $\mathcal{E}$ (indirectly) queried if $\mathbf{k}$ satisfies a non-trivial linear relation and got 1 as a reply in the real GGM. This happens since $\mathcal{E}$ is deterministic, thus, if it does not find any non-trivial linear relation, it gets the same replies from both the real GGM and the generic group modeling $\mathbb{G}^\kappa$. A description of $\mathcal{A}$ appears in Figure 5.2.

Reduction $\mathcal{A}(\mathbf{k})$:

1 : Sample in exponential time $x \leftarrow^\$ \mathcal{D}$
2 : Run $\mathcal{E}(\mathbf{k}, x)$
3 : **When** $\mathcal{E}$ queries $\mathcal{O}_{\mathsf{add}}(T_1, T_2)$:
4 : Retrieve $\mathbf{a}_1, \mathbf{a}_2 \in \mathbb{F}_q^\kappa$ such that $T_i = \mathbf{a}_i^\top \mathbf{k}$
5 : Query $\mathcal{O}_{\mathsf{add}}(T_1, T_2)$ and store $\mathbf{a}_1 + \mathbf{a}_2$ as a representation of the result
6 : **When** $\mathcal{E}$ queries $\mathcal{O}_{\mathsf{eq}}(T_1, T_2)$:
7 : Retrieve $\mathbf{a}_1, \mathbf{a}_2 \in \mathbb{F}_q^\kappa$ such that $T_i = \mathbf{a}_i^\top \mathbf{k}$
8 : Query $b \leftarrow \mathcal{O}_{\mathsf{eq}}(T_1, T_2)$ and return $\mathcal{E} \leftarrow b$.
9 : *If* $b = 1$ and $\mathbf{a}_1 - \mathbf{a}_2 \neq \mathbf{0}$:
10 : Return $\mathbf{a}_1 \neq \mathbf{a}_2$
11 : **When** $\mathcal{E}$ halts: Return $\perp$

Figure 5.2: Reduction $\mathcal{A}$ finding a linear relation among κ elements $\mathbf{k}$.

First of all we observe that inductively $\mathcal{A}$ can store a representation of each element in base $\mathbf{k}$ as initially $k_i = \mathbf{e}_i^\top \mathbf{k}$, with $\mathbf{e}_i$ being 1 in the i-th position and 0 elsewhere. Next, if $\mathcal{A}$ ever executes line 10, it returns a linear relation since $b = 1$ implies $\mathbf{a}_1^\top \mathbf{k} = \mathbf{a}_2^\top \mathbf{k}$ and therefore $(\mathbf{a}_1 - \mathbf{a}_2)^\top \mathbf{k}$.

Finally, if line 10 is never executed, then $\mathcal{E}$ receives 1 from $\mathcal{O}_{\mathsf{eq}}$ only when $\mathbf{a}_1 \neq \mathbf{a}_2$, implying that $\mathcal{A}$ correctly simulates simultaneously the standard GGM and the generic group modeling $\mathbb{G}^\kappa$. Thus in this case $A^*_x = A_{\mathbf{k},x}$.

In conclusion, $(\mathcal{A} \to \perp) \Rightarrow A^*_x = A_{\mathbf{k},x}$, therefore

$$\Pr\left[\mathcal{A}(\mathbf{k}) \to \mathbf{v},\ \mathbf{v}^\top \mathbf{k} = 0\right] \geq \Pr\left[A^*_x \neq A_{\mathbf{k},x}\right].$$

Since finding linear relations on a vector of random group elements is equivalent to the discrete logarithm problem, we have that $\Pr\left[A^*_x \neq A_{\mathbf{k},x}\right]$ is negligible. □

Proof of Claim 5.1.3. To simplify notation, the summations below are taken for all $\mathbf{k}_0 \in \mathbb{G}^\kappa$, $A_0 \in \mathbb{F}_q^{m,\kappa}$ and $x_0 \in \{0,1\}^n$.

$$\begin{aligned}
&\Delta((\mathbf{k}, A^*_x), (\mathbf{k}, A_{\mathbf{k},x})) = \\
&= \frac{1}{2} \cdot \sum_{\mathbf{k}_0,\, A_0} \left|\Pr\left[\mathbf{k} = \mathbf{k}_0,\ A^*_{\mathbf{k},x} = A_0\right] - \Pr\left[\mathbf{k} = \mathbf{k}_0,\ A_{\mathbf{k},x} = A_0\right]\right| \\
&\leq \sum_{\mathbf{k}_0,\, A_0,\, x_0} \frac{1}{2} \left|\Pr\left[A^*_{\mathbf{k},x} = A_0 \middle| \mathbf{k} = \mathbf{k}_0,\ x = x_0\right] - \Pr\left[A_{\mathbf{k},x} = A_0 | \mathbf{k} = \mathbf{k}_0,\ x = x_0\right]\right| \\
&\qquad\qquad \ldots \cdot \Pr\left[\mathbf{k} = \mathbf{k}_0,\ x = x_0\right] \\
&= \sum_{\mathbf{k}_0,\, x_0} \Pr\left[A^*_x \neq A_{\mathbf{k},x} | \mathbf{k} = \mathbf{k}_0,\ x = x_0\right] \cdot \Pr\left[\mathbf{k} = \mathbf{k}_0,\ x = x_0\right] \\
&= \Pr\left[A^*_x \neq A_{\mathbf{k},x}\right] \leq \varepsilon
\end{aligned}$$

where in last step we applied Claim 5.1.2 and in the third step we used the fact that

$$\sum_{A_0} \frac{1}{2} \left|\Pr\left[A^*_{\mathbf{k},x} = A_0 \middle| \mathbf{k} = \mathbf{k}_0,\ x = x_0\right] - \Pr\left[A_{\mathbf{k},x} = A_0 | \mathbf{k} = \mathbf{k}_0,\ x = x_0\right]\right|$$

is equal to $\Pr\left[A^*_x \neq A_{\mathbf{k},x} | \mathbf{k} = \mathbf{k}_0,\ x = x_0\right]$ because

- If $A^*_{x_0} = A_{\mathbf{k}_0,x_0}$ then the summation only contains terms that are 0, as A_0 is either different or equal to both matrices. Thus the above sum is 0, and so is the probability of the two matrices being different when $\mathbf{k} = \mathbf{k}_0$ and $x = x_0$.
- If $A^*_{x_0} \neq A_{\mathbf{k}_0,x_0}$ then the only non zero terms of the summation are those in which A_0 is equal to either $A^*_{x_0}$ or $A_{\mathbf{k}_0,x_0}$. This yield only two terms both equal to $1/2$, implying that the sum equals 1, and so does the probability of the two matrix being different when $\mathbf{k} = \mathbf{k}_0$ and $x = x_0$.

□

Proof of Claim 5.1.4. We begin observing that $\mathbf{k}$ is independent from both x_1 and x_2, therefore

$$\Delta((\mathbf{k}, A^*_{x_1}), (\mathbf{k}, A^*_{x_2})) = \Delta(\mathbf{k}, \mathbf{k}) + \Delta(A^*_{x_1}, A^*_{x_2}) = \Delta(A^*_{x_1}, A^*_{x_2}).$$

From the way we defined Samp, see Figure 5.1, there exists a random variable x_3 uniformly distributed over $\{0,1\}^n$ such that $x_2 = F(x_3)$. Thus, since for each element z, F satisfy the identity $A^*_z = A^*_{F(z)}$, we have that

$$\Delta(A^*_{x_1}, A^*_{x_2}) \;=\; \Delta(A^*_{x_1}, A^*_{F(x_3)}) \;=\; \Delta(A^*_{x_1}, A^*_{x_3}) \;=\; 0$$

where the last equation follows as x_1 and x_3 are both uniformly distributed. □

Proof of Claim 5.1.5. First of all we observe that

$$\Delta((\mathbf{k}, f_{\mathbf{k}}(x_1)), (\mathbf{k}, f_{\mathbf{k}}(x_2))) \;\leq\; \Delta((\mathbf{k}, A_{\mathbf{k},x_1}), (\mathbf{k}, A_{\mathbf{k},x_2}))$$

since the two distribution on the left hand can be obtained from those in the right hand through the map

$$(\mathbf{k}, A) \;\mapsto\; (\mathbf{k}, A \cdot \mathbf{k})$$

where we use the fact that $A_{\mathbf{k},x} \cdot \mathbf{k} = f_{\mathbf{k}}(x)$ by the way these matrices are defined. Next let z_1, z_2 be two random variables, with $z_1 \leftarrow^{\$} \{0,1\}^n$ and $z_2 \leftarrow^{\$} \mathsf{Samp}(1^\lambda)$. Applying the triangular inequality twice we get

$$\begin{aligned}
&\Delta((\mathbf{k}, A^*_{x_1}), (\mathbf{k}, A_{x_2})) \\
&\leq \Delta((\mathbf{k}, A_{\mathbf{k},x_1}), (\mathbf{k}, A^*_{z_1})) \;+\; \Delta((\mathbf{k}, A^*_{z_1}), (\mathbf{k}, A^*_{z_2})) \;+\; \Delta((\mathbf{k}, A^*_{z_2}), (\mathbf{k}, A_{\mathbf{k},x_2})) \\
&\leq 2\varepsilon
\end{aligned}$$

where we the first and last term are smaller than ε from Claim 5.1.3 and the central term is zero from Claim 5.1.4. □

Proof of Claim 5.1.6. Given a GPPT adversary $\mathcal{A}$ that given $\mathbf{k}$ returns two different points $x_1, x_2 \in \mathcal{X}$ for which $f_{\mathbf{k}}(x_1) = f_{\mathbf{k}}(x_2)$, we build a GPPT adversary $\mathcal{B}$ that finds linear relation on a vector of κ random group elements.

The idea is that if x_1 and x_2 lies in $\mathcal{X}$ are distinct, then their associated matrices $A^*_{x_1}$ and $A^*_{x_2}$ need also to be different, or else one of these two points would not be included in $\mathcal{X}$ by $\mathcal{T}$. If $\mathcal{A}$ returned inputs for which $A^*_{x_1} \neq A_{\mathbf{k},x_1}$ or $A^*_{x_2} \neq A_{\mathbf{k},x_2}$, then as done in the proof of Claim 5.1.2, we could extract a linear relation over the elements of $\mathbf{k}$. Conversely, if $A^*_{x_1} = A_{\mathbf{k},x_1}$ and $A^*_{x_2} = A_{\mathbf{k},x_2}$, then $A^*_{x_1} - A^*_{x_2}$ is a non zero matrix that vanishes on $\mathbf{k}$. A full description of $\mathcal{B}$ appears in Figure 5.3

Let coll be the event that the condition at step 2 is not satisfied, i.e. the event in which $\mathcal{A}$ returns a valid collision, and equal be the event $A^*_{x_b} = A_{\mathbf{k},x_b}$ for $b \in \{0,1\}$. If coll and $\neg\mathsf{equal}$ occurs, $\mathcal{B}$ finds a linear relation with probability 1, as observed in the proof of Claim 5.1.2. Conversely, if coll and equal we have that

$$f_{\mathbf{k}}(x_b) \;=\; A_{\mathbf{k},x_b} \cdot \mathbf{k} \;=\; A^*_{x_b} \cdot \mathbf{k}. \qquad f_{\mathbf{k}}(x_1) = f_{\mathbf{k}}(x_2) \quad \Rightarrow \quad A^*_{x_1}\mathbf{k} = A^*_{x_2}\mathbf{k}.$$

Which implies that $(A^*_{x_1} - A^*_{x_2})$ vanishes on $\mathbf{k}$. Furthermore this is a non trivial relation since

$$x_1, x_2 \in \mathcal{X},\; x_1 \neq x_2 \quad \Rightarrow \quad A^*_{x_1} \neq A^*_{x_2} \quad \Rightarrow \quad A^*_{x_1} - A^*_{x_2} \neq \mathbf{0}.$$

We can thus conclude that if coll occurs, then $\mathcal{A}$ finds a a linear relation which implies that $\Pr[\mathsf{coll}]$ is negligible. □

Reduction $\mathcal{B}(\mathbf{k})$:

1 : Run $\mathcal{A}(\mathbf{k}) \to (x_1, x_2)$
2 : **If** $\neg(x_1 \neq x_2 \wedge x_1, x_2 \in \mathcal{X} \wedge f_{\mathbf{k}}(x_1) \neq f_{\mathbf{k}}(x_2))$: Return $\perp$
3 : Compute $A^*_{x_1}$, $A^*_{x_2}$, $A_{\mathbf{k},x_1}$, $A_{\mathbf{k},x_2}$
4 : **If** $A^*_{x_b} \neq A_{\mathbf{k},x_b}$ for $b \in \{0,1\}$:
5 : Compute as done in Fig. 5.2 a linear relation $\mathbf{v} \ : \ \mathbf{v}^\top \mathbf{k} = 0$
6 : Return $\mathbf{v}$
7 : **Else**:
8 : Return $A^*_{x_1} - A^*_{x_2}$

Figure 5.3: Reduction $\mathcal{A}$ finding a linear relation among κ elements $\mathbf{k}$.

5.1.3 Hard-Core Predicate

In [66] Goldreich and Levin proved that in the standard model, any OWF f with domain in $\{0,1\}^n$ can be transformed into another OWF $f'(\mathbf{x}, \mathbf{r}) = (f(\mathbf{x}), \mathbf{r})$ that admits the hard-core predicate $\mathbf{x}^\top \mathbf{r}$.

We observe that, given an algebraic OWF family, that is secure against any GPPT adversary, even when the function's domain is restricted as discussed in the previous section, the same result applies.

Theorem 5.1.2. *Let* (Gen, f) *an algebraic OWF familty with* $f : \{0,1\}^n \to \mathbb{G}^m$ *and* Gen *returning* $k \sim U(\mathbb{G}^\kappa)$. *Then there exists* ε *negligible such that for all* GPPT *adversaries* $\mathcal{A}$

$$\Pr\left[\mathcal{A}(k, y, \mathbf{r}) \to b, \quad b = \mathbf{x}^\top \mathbf{r} \;\middle|\; \begin{matrix} k \leftarrow^{\$} \mathsf{Gen}(1^\lambda), & \mathbf{r} \leftarrow^{\$} \{0,1\}^n \\ \mathbf{x} \leftarrow^{\$} \mathsf{Samp}(1^\lambda), & y \leftarrow f_k(\mathbf{x}) \end{matrix}\right] \leq \varepsilon(\lambda)$$

The proof is identical to the original result up to observing that in this case the function's input is sampled with a different distribution than the uniform one and that the reduction only needs black-box access to the group.

5.2 Negative Results for Algebraic NIZK-AoK

5.2.1 Intuition

The final step to obtain our claimed result on Algebraic NIZK-AoK is to show that it would allow to construct a vector commitment in GPPT violating the negative result of Theorem 4.3.4.

To build up intuition we first provide a toy construction assuming we have a NIZK-AoK $(\mathsf{G}, \mathsf{P}, \mathsf{V})$ for the discrete logarithm relation, i.e. that given $K, H \in \mathbb{G}$ proves knowledge of x such that $H = x \cdot K$. The idea is to tweak a regular Petersen commitment for n field elements until we make it hiding. An initial approach is, given $x_1, \ldots, x_n \in \mathbb{F}_q$, to commit to them by sending $C = x_1 K_1 + \ldots + x_n K_n$ with K_i random group elements in the public parameters.

To open position i we can send, together with x_i, the elements $x_j K_j$ along with a proof of knowledge of x_j.

This would be position binding, as from any two opening of the same position a challenger can extract (using the NIZK extractor) two different representations of C in base $K_1, \ldots, K_n$, which would break the security of standard Petersen commitments. However this would not yet be hiding, even if the argument used is zero knowledge. The issue is that $x \cdot K$ does not hide x. More concretely, in the hiding game, an adversary could send $\mathbf{x}^0 = (1, 0, \ldots 0)$ and $\mathbf{x}^1 = (2, 0, \ldots, 0)$. Later, testing if C is equal to K_1 or $2K_1$, it would be able to understand which was the committed vector.

A way to address this issue is resorting to an hard-core predicate $\ell : \mathbb{F}_q \to \{0,1\}$ for the discrete logarithm OWF, such as the least significant bit. This time, instead of committing to $x_1, \ldots, x_n$, we present a commitment to bits: given $b_1, \ldots, b_n$ the committer samples

$$x_i \leftarrow^{\$} \mathbb{F}_q \ : \ \ell(x_i) = b_i \qquad C = x_1 K_1 + \ldots + x_n K_n.$$

An opening to b_i can again be the message x_i together with $x_j K_j$ and a proof of knowledge for x_j, but now the verifier has to further verify that $\ell(x_i) = b_i$.

This scheme would be binding as before, up to observing that $\ell(x_i^0) \neq \ell(x_i^1)$ implies $x_i^0 \neq x_i^1$. Conversely the scheme is hiding because until position i is opened, nothing about x_i is revealed apart from $x_i K_i$, also because our argument is zero-knowledge. Thus guessing the message at position i reduces to the hardness of predicting the hard-core predicate ℓ.

5.2.2 Hiding VC from NIZK-AoK

We now discuss how to generalize the construction in Section 5.2.1 to algebraic OWF families.

The first issue is that not all OWFs admit hard-core predicates. We address this using the Goldreich-Levin transformation, see Section 5.1.3, $f'_k(\mathbf{x}, \mathbf{r}) = (f_k(\mathbf{x}), \mathbf{r})$ which admits the hard-core bit $\mathbf{x}^\top \mathbf{r}$.

The second issue is that OWF may not be collision resistant[2]. An example is $f_K : \{0, \ldots, 2q-1\} \to \mathbb{G}$ such that $f_K(x) = x \cdot K$ where $f_K(0) = f_K(q)$. This may allow an adversary to break position-binding by finding two $\mathbf{x}, \mathbf{x}'$ with $f_k(\mathbf{x}) = f_k(\mathbf{x}')$ and different hard-core bits. To address this we introduced in Section 5.1.2 two GPPT procedures Memb and Samp to restrict the domain of a OWF in order to make it collision resistant and to sample from this restricted domain.

Given these observations, in Fig. 5.4 we provide a complete description of the resulting VC, with (Gen, f) being an algebraic OWF where $f : \{0,1\}^\mu \to \mathbb{G}^m$ and Gen samples uniformly from $\mathbb{G}^\kappa$, and $(\mathsf{G}, \mathsf{P}, \mathsf{V})$ is a NIZK-AoK for the relation

$$\mathcal{R} \ = \ \{((k, y), x) \ : \ f_k(x) = y\}.$$

Theorem 5.2.1. *If $(\mathsf{G}, \mathsf{P}, \mathsf{V})$ is a NIZK-AoK, the VC described in Fig. 5.4 is computable in* GPPT *time, position-binding, hiding and returns commitments with $O(1)$ group elements.*

[2]In the previous example $x \mapsto x \cdot K$ is collision resistant because it is a bijection

$\mathsf{VC.Setup}^*(1^\lambda, n)$:

1 : $\mathsf{crs}_{i,j} \leftarrow \mathsf{G}(1^\lambda)$
2 : $k_i \leftarrow \mathsf{Gen}(1^\lambda)$
3 : $\mathsf{pp} \leftarrow \{\mathsf{crs}_{i,j}, k_i \ : \ i,j \in [n]\}$

$\mathsf{VC.Com}^*(\mathsf{pp}, b_1, \dots, b_n)$:

1 : $\mathbf{x}_i \leftarrow^{\$} \mathsf{Samp}(1^\lambda)$
2 : $\mathbf{r}_i \leftarrow^{\$} \{0,1\}^\mu$ such that $\mathbf{x}_i^\top \mathbf{r}_i = b_i$
3 : $C \leftarrow \sum_{i=1}^n f_{k_i}(\mathbf{x}_i)$
4 : $\mathsf{aux} \leftarrow (\mathbf{x}_i)_{i=1}^n$
5 : Return $(C, (\mathbf{r}_j)_{j=1}^n)$, aux

$\mathsf{VC.Open}^*(\mathsf{pp}, i, \mathsf{aux})$:

1 : $\pi_{i,j} \leftarrow \mathsf{P}(\mathsf{crs}_{i,j}, k_j, f_{k_j}(\mathbf{x}_j), \mathbf{x}_j)$
2 : $\Lambda_i = (\mathbf{x}_i, (f_{k_j}, \pi_{i,j})_{j \neq i})$
3 : Return Λ_i

$\mathsf{VC.Vfy}^*(\mathsf{pp}, (C, D), b_i, i, \Lambda_i)$

1 : Parse $\Lambda_i = (\mathbf{x}_i, (\mathbf{Y}_j, \pi_{i,j})_{j \neq i})$
2 : Accept if and only if:
3 : $\quad C = f_{k_i}(\mathbf{x}_i) + \sum_{j \neq i} \mathbf{Y}_j$
4 : $\quad 1 = \mathsf{V}(\mathsf{crs}_j, k_j, \mathbf{Y}_j, \pi_j)$
5 : $\quad 1 = \mathsf{Memb}(\mathbf{x}_i), \quad b_i = \mathbf{x}_i^\top \mathbf{r}_i$

Figure 5.4: Hiding Vector Commitment from a NIZK-AoK for $\mathcal{R}$.

Proof. **Efficiency**: To show that all procedures can be computed in GPPT time it suffices to observe that all steps are efficiently computable, and by Lemma 5.1.1 $\mathsf{Samp}, \mathsf{Memb}$ are computable in GPPT time.

Constant Group-Elements Commitment: Commitments only contains m group elements in $C \in \mathbb{G}^m$ with $\mathbb{G}^m$ being the domain of f. Because m does not depend on n, the commitment only contains a constant number of group elements in n (although it may depends on λ).

Position Binding: Given $\mathcal{A}$ breaking position binding we build $\mathcal{B}$ which, given κn random group elements in $\mathbb{G}$, finds a linear relation among them. Note that this is equivalent to breaking the discrete logarithm problem, which in Maurer's GGM is known to be hard [88].

$\mathcal{B}(\mathbf{V})$:

1 : Parse the input as n OWF keys $\mathbf{V} = (\mathbf{k}_1, \dots, \mathbf{k}_n) \in (\mathbb{G}^\kappa)^n$
2 : Sample with the NIZK extractor $(\mathsf{crs}_{i,j}, \mathsf{td}_{i,j}) \leftarrow \mathsf{E}(1^\lambda)$
3 : $\mathsf{pp} \leftarrow \{\mathsf{crs}_{i,j}, \mathbf{k}_i \ : \ i,j \in [n]\}$
4 : $\mathcal{A}(\mathsf{pp}) \to (c, i, b_0, \Lambda_0, b_1, \Lambda_1)$
5 : Parse $c = (C, (\mathbf{r}_j)_{j=1}^n)$ and $\Lambda_\beta = (\mathbf{x}_i^\beta, (\mathbf{Y}_j^\beta, \pi_{i,j}^\beta)_{j \neq i})$ for $\beta \in \{0,1\}$
6 : Extract $\mathbf{x}_j^\beta \leftarrow \mathsf{E}(\mathsf{td}_{i,j}, \mathbf{k}_j, \mathbf{Y}_j^\beta, \pi_{i,j}^\beta)$ for $j \in [n] \setminus \{i\}$
7 : Compute $A_j^\beta \in \mathbb{F}_q^{m,\kappa}$ such that $f_{\mathbf{k}_j}(\mathbf{x}_j^\beta) = A_j^\beta \cdot \mathbf{k}_j$ for $j \in [n]$
8 : Return $(A_j^0 - A_j^1)_{j=1}^n$

Figure 5.5: $\mathcal{B}$ reducing position binding to the discrete logarithm problem.

We preliminary notice that in line 7, A_j^β can be computed efficiently. A way to achieve this is locally storing during the execution of $f_{\mathbf{k}_j}(\mathbf{x}_j^\beta)$ a representation for each element queried to the GGM oracle as a linear combination of the group elements in $\mathbf{k}_j$. Doing so, the matrix A_j^β is given by the output elements' representations .

Next we define the following events:

$$\begin{array}{ll} \mathcal{B}\text{ wins} & : \sum_{j=1}^n (A_j^0 - A_j^1)\mathbf{k}_j = \mathbf{0} \text{ and } (A_j^0 - A_j^1)_{j=1}^n \text{ is a non-zero matrix} \\ \mathcal{A}\text{ wins} & : \mathcal{A} \text{ breaks position-binding} \\ \mathsf{Ext} & : f_{\mathbf{k}_j}(\mathbf{x}_j^\beta) = \mathbf{Y}_j^\beta \text{ for all } j \neq i \text{ and } \beta \in \{0,1\} \\ \mathsf{Coll} & : \mathbf{x}_i^0 \neq \mathbf{x}_i^1 \ \wedge \ f_{\mathbf{k}_i}(\mathbf{x}_i^0) = f_{\mathbf{k}_i}(\mathbf{x}_i^1) \ \wedge \ 1 = \mathsf{Memb}(\mathbf{x}_i^0) = \mathsf{Memb}(\mathbf{x}_i^1) \end{array}$$

Since $\mathcal{A}$ wins only if the openings are correct, $\pi_{i,j}^\beta$ are all accepted. Calling ε_1 the extractor error, see Section 2.2.3, we have that

$$\begin{aligned} \Pr[\mathsf{Ext} \mid \mathcal{A}\text{ wins}] &= 1 - \Pr\left[\bigvee\nolimits_{j\neq i}\bigvee\nolimits_{\beta=0}^{1} f_{\mathbf{k}_j}(\mathbf{x}_j^\beta) \neq \mathbf{Y}_j^\beta \;\middle|\; \mathcal{A}\text{ wins}\right] \\ &\geq 1 - \sum\nolimits_{j\neq i}\sum\nolimits_{\beta=0}^{1} \Pr\left[f_{\mathbf{k}_j}(\mathbf{x}_j) = \mathbf{Y}_j^\beta \;\middle|\; \mathcal{A}\text{ wins}\right] \\ &\geq 1 - (2n-2)\varepsilon_1(\lambda) \;=\; 1 - \varepsilon_1^*(\lambda). \end{aligned}$$

with $\varepsilon_1^* = (2n-2)\varepsilon_1$ being a negligible function. By Lemma 5.1.1 we also have that $\Pr[\mathsf{Coll}] \leq \varepsilon_2(\lambda)$. Next, we notice that $\mathcal{B}$ wins if $\mathcal{A}$ wins, Ext and $\neg\mathsf{Coll}$ occurs. Indeed in this case

$$C \;=\; f_{\mathbf{k}_i}(\mathbf{x}_i^\beta) + \sum\nolimits_{j\neq i} \mathbf{Y}_j^\beta \;=\; \sum\nolimits_{j=1}^n f_{\mathbf{k}_j}(\mathbf{x}_j^\beta) \;=\; \sum\nolimits_{j=1}^n A_j^\beta \mathbf{k}_j$$

for both $\beta \in \{0,1\}$, where the first equality follows by $\mathcal{A}$ wins, the second one from Ext and the third one by construction. This implies $\sum_{j=1}^n (A_j^0 - A_j^1)\mathbf{k}_j = \mathbf{0}$. Moreover this relation is non trivial since, as $\mathcal{A}$ wins, we must have

$$\begin{aligned} b^0 \neq b^1 \quad \Rightarrow \quad (\mathbf{x}_i^0)^\top \mathbf{r}_i \neq (\mathbf{x}_i^1)^\top \mathbf{r}_i \quad \Rightarrow \quad \mathbf{x}_i^0 \neq \mathbf{x}_i^1 \quad \Rightarrow \\ \Rightarrow \quad f_{\mathbf{k}_i}(\mathbf{x}_i^0) \neq f_{\mathbf{k}_i}(\mathbf{x}_i^1) \quad \Rightarrow \quad A_i^0\mathbf{k}_i \neq A_i^1\mathbf{k}_i \quad \Rightarrow \quad A_i^0 - A_i^1 \neq \mathbf{0}. \end{aligned}$$

Where the fourth implication comes from $\neg\mathsf{Coll}$. To conclude we finally bound the advantage of $\mathcal{A}$ with the probability that $\mathcal{B}$ successfully finds a linear relation.

$$\begin{aligned} \Pr[\mathcal{B}\text{ wins}] &\geq \Pr[\mathcal{A}\text{ wins},\ \mathsf{Ext},\ \neg\mathsf{Coll}] \\ &\geq \Pr[\mathcal{A}\text{ wins},\ \mathsf{Ext}] - \Pr[\mathsf{Coll}] \\ &\geq \Pr[\mathsf{Ext} \mid \mathcal{A}\text{ wins}] \cdot \Pr[\mathcal{A}\text{ wins}] - \varepsilon_2(\lambda) \\ &\geq (1 - \varepsilon_1^*(\lambda)) \Pr[\mathcal{A}\text{ wins}] - \varepsilon_2(\lambda). \end{aligned}$$

Since $\mathcal{B}$ succeeds with negligible probability, the advantage of $\mathcal{A}$ must be negligible as well.

Hiding: We show that given any GPPT adversary $\mathcal{A}$ executed in the game described in Fig. 4.2, we can build an adversary $\mathcal{B}$ guessing the Goldreich-Levin hard-core predicate for f.

The idea is, given $(\mathbf{k}, \mathbf{Y}, \mathbf{r})$, to setup the VC parameters with the simulator, and use $\mathbf{k}$ as the OWF key for a randomly guessed entry i. Next $\mathcal{A}$ proposes its two vectors of bits $\mathbf{b}_0, \mathbf{b}_1$. If they differ on the guessed position i, $\mathcal{B}$ proceeds computing the commitment, where it uses $\mathbf{Y}, \mathbf{r}$ as the OWF image for the i-th entry, and simulating the required openings. Finally, once $\mathcal{A}$ guesses a bit, it returns the same value. A detailed description appears in Fig. 5.6.

$\mathcal{B}(\mathbf{k}, \mathbf{Y}, \mathbf{r})$:

Sample $i \leftarrow^{\$} \{1, \dots, n\}$ a guess on the position $\mathcal{A}$ will choose
Sample $k_j \leftarrow \mathsf{Gen}(1^\lambda)$ for $j \neq i$ and set $k_i \leftarrow \mathbf{k}$
Sample with the NIZK simulator $(\mathsf{crs}_{\ell,j}, \mathsf{td}_{\ell,j}) \leftarrow \mathsf{S}(1^\lambda)$ with $\ell, j \in [n]$
$\mathsf{pp} \leftarrow \{\mathsf{crs}_{\ell,j}, k_j \ : \ \ell, j \in [n]\}$
$\mathcal{A}(\mathsf{pp}) \to \mathbf{b}^0, \mathbf{b}^1$ differing only at position i' (wlog $b^0_{i'} = 0$ and $b^1_{i'} = 1$)
If $i \neq i'$: Return $\perp$
// Simulate the commitment
For $j \neq i$:
  Sample $\mathbf{x}_j \leftarrow^{\$} \mathsf{Samp}(1^\lambda)$ and $\mathbf{r}_j \leftarrow^{\$} \{0,1\}^\mu$ with $\mathbf{x}_j^\top \mathbf{r}_j = b^0_j = b^1_j$
Set $\mathbf{r}_i \leftarrow \mathbf{r}$
Compute $C \leftarrow \mathbf{Y} + \sum_{j \neq i} f_{k_j}(\mathbf{x}_j)$
Create the commitment $c \leftarrow (C, (\mathbf{r}_j)_{j=1}^n)$
// Simulate the openings
For $\ell \neq i$:
  $\pi_{\ell,j} \leftarrow \mathsf{S}(\mathsf{td}_{\ell,j}, k_j, f_{k_j}(\mathbf{x}_j))$ for all $j \neq \ell$
  $\Lambda_\ell = (\mathbf{x}_\ell, (f_{k_j}(\mathbf{x}_j), \pi_{\ell,j})_{j \neq \ell})$
// Execute $\mathcal{A}$ to guess the hard-core bit
Execute $\mathcal{A}(\mathsf{pp}, C, (\Lambda_\ell)_{\ell \neq i}) \to b$ and return b

Figure 5.6: Reduction $\mathcal{B}$ guessing the Goldwasser-Levin hardcore predicate of $f_{\mathbf{k}}$.

First we observe that due to the Zero-Knowledge property, distinguishing $\mathsf{crs}_{\ell,j}, \pi_{\ell,j}$ generated by $\mathcal{B}$ from the ones returned by real challenger of $\mathsf{ExpHideVC}$ in Fig. 4.2 cannot be done with advantage greater than a negligible ε by any GPPT adversary.

Next, assume $i = i'$. Calling β the hard-core predicate $\mathcal{B}$ has to guess, $\mathcal{B}$ correctly commits to $\mathbf{b}_\beta$ since its challenger sets $\mathbf{Y} = f_{\mathbf{k}}(\mathbf{x}) = f_{k_i}(\mathbf{x})$ with[3] $b^\beta_i = \beta = \mathbf{x}^\top \mathbf{r}$, $\mathbf{x} \leftarrow \mathsf{Samp}(1^\lambda)$ and $\mathbf{r} \leftarrow^{\$} \{0,1\}^\mu$. Thus $\mathcal{B}$ wins if $\mathcal{A}$ correctly guesses β, and the initial guess is correct, i.e. $i = i'$. We conclude that

$$\begin{aligned} \mathsf{Adv}(\mathcal{B}) &= |\Pr[\mathcal{A} \to 0,\ i = i' | \beta = 0] - \Pr[\mathcal{A} \to 0,\ i = i' | \beta = 1]| \\ &= \Pr[i = i'] \cdot |\Pr[\mathcal{A} \to 0 | \beta = 0] - \Pr[\mathcal{A} \to 0 | \beta = 1]| \\ &\geq \frac{\mathsf{Adv}(\mathcal{A}) - \varepsilon(\lambda)}{n}. \end{aligned}$$

[3] note that we assumed without loss of generality $b^0_i = 0$ and $b^1_i = 1$.

Where the third inequality uses $\Pr[i = i'] = 1/n$, which follows as $\mathcal{A}$ has no information on i when it computes $\mathbf{b}_0, \mathbf{b}_1$ (and in particular i'). □

5.2.3 Final Result

Combining Theorem 5.2.1 and Theorem 4.3.4 we can eventually derive the following

Theorem 5.2.2. *Given* (Gen, f) *a one way function family with* Gen *returning a uniformly sampled vector in* $\mathbb{G}^\kappa$ *and* $f : \{0,1\}^\mu \to \mathbb{G}^m$*, then there exists no Algebraic NIZK-AoK for the relation*

$$\mathcal{R} = \{((k, y), x) \ : \ f_k(x) = y\}.$$

5.3 Algebraic NIZK

5.3.1 Hard Subset Membership Problem

In this section we recall the definition of Hard Subset Membership Problem, presented in [60]. Given an NP relation $\mathcal{R}$, its associated language $\mathcal{L}$ is the set of all statements x for which $(x, w) \in \mathcal{R}$ for some witness w. Informally, the relation $\mathcal{R}$ is a hard subset problem if there are two ways to sample from $\mathcal{L}$ and its complement $\{0,1\}^* \setminus \mathcal{L}$ that are computationally hard to distinguish. As mentioned this captures DDH since the distributions (G, aG, bG, abG) and (G, aG, bG, cG) with a, b, c random field elements and $c \neq a \cdot b$ are hard distinguish. More generally this captures decisional assumptions and their related relations such as Decision Linear and Matrix-DDH. More formally:

Definition 5.3.1. *A Subset Membership Problem is a tuple* $(\mathcal{R}, \mathsf{SampGood}, \mathsf{SampBad})$ *with* $\mathcal{R}$ *an NP relation, and* $\mathsf{SampGood}$, $\mathsf{SampBad}$ *such that*

- $\mathsf{SampGood}(1^\lambda) \to (x, w) \quad \Rightarrow \quad (x, w) \in \mathcal{R}.$
- $\mathsf{SampBad}(1^\lambda) \to x \quad \Rightarrow \quad \nexists w \ : \ (x, w) \in \mathcal{R}.$

A subset membership problem is called hard (against GPPT *adversaries) if* $\exists \varepsilon$ *negligible such that for all* $\mathcal{A}$ GPPT

$$x_0 \leftarrow^{\$} \mathsf{SampBad}(1^\lambda), \quad (x_1, w_1) \leftarrow^{\$} \mathsf{SampGood}(1^\lambda)$$
$$\Rightarrow \quad |\Pr[\mathcal{A}(x_0) \to 0] - \Pr[\mathcal{A}(x_1) \to 0]| \ \leq \ \varepsilon(1^\lambda).$$

5.3.2 Preliminary Adversary

Having defined relations with a hard subset problem against GPPT adversaries, in the rest of this section we show that these relations do not admit a NIZK argument in Maurer's GGM. Toward this goal we first construct an adversary $\mathcal{A}$ that, given a NIZK crs and oracle access to the simulator, either returns a proof of a false statement or it finds a linear relation among the group elements in the CRS. In order to ensure sequential executions of $\mathcal{A}$ we give it an affine space V in input, containing linear relations already found among the crs elements. Finally, we will allow $\mathcal{A}$ to fail with an arbitrary small (but non-negligible) probability $1/p$ with $p = \mathsf{poly}(\lambda)$. More formally

Lemma 5.3.1. *Let* $(\mathcal{R}, \mathsf{SampGood}, \mathsf{SampBad})$ *be a hard subset problem and* $(\mathsf{G}, \mathsf{P}, \mathsf{V})$ *a NIZK argument for* $\mathcal{R}$ *with simulator* S*. Then, for any* $p = \mathsf{poly}(\lambda)$*, there exists a* GPPT *adversary* $\mathcal{A}$ *such that: given*

$$(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{S}(1^\lambda), \quad x \leftarrow \mathsf{SampBad}(1^\lambda) \quad : \quad \begin{array}{l} \mathsf{crs} \ = \ (\mathbf{Y}, c') \in \mathbb{G}^n \times \{0,1\}^* \\ x \ = \ (\mathbf{Z}, z') \in \mathbb{G}^m \times \{0,1\}^* \end{array}$$

and $V \leq \mathbb{F}_q^n$ *such that* $\mathbf{Y} \in V \cdot G$*, calling* $\mathbf{Z} = \mathbf{z} \cdot G$ *then either:*

1. $\mathcal{A}(V, \mathsf{crs}, x) \to (\mathsf{proof}, \pi)$ *such that* $1 \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi)$*.*
2. $\mathcal{A}(V, \mathsf{crs}, x) \to \mathsf{query}$*. Then setting* $\pi \leftarrow \mathsf{S}(\mathsf{td}, x)$*,* $\mathcal{A}(V, \mathsf{crs}, x, \pi)$ *either aborts with probability smaller than* $1/p(\lambda)$ *or it returns* L *such that*

$$(\mathbf{Y}, \mathbf{Z}) \in L \cdot G \quad \wedge \quad L \cap \left(\mathbb{F}_q^n \times \{\mathbf{z}\}\right) \lneq (V \times \{\mathbf{z}\})$$

First of all we remark that the second condition simply states that the affine space L contains a new linear relation among the elements $\mathbf{Y}, \mathbf{Z}$ that is non-trivial with respect to $\mathbf{Y}$. Next, we observe that this adversary could be trivially used to break the *simulation soundness* property of the underlying NIZK. This is a stronger version of soundness in which the adversary has oracle access to a simulator and wins if it returns a proof for a false statement that was not queried. The way to use $\mathcal{A}$ is sampling $n+1$ independent elements with $\mathsf{SampBad}(1^\lambda)$ and sequentially passing them to $\mathcal{A}$, using the simulation oracle to reply query requests. At each step (assuming $\mathcal{A}$ does not abort) either $\mathcal{A}$ finds a new linear relationship on the CRS' group elements, reducing the dimension of V by 1, or it returns a proof for x breaking soundness. Calling n the number of group elements in the CRS, $\mathcal{A}$ can find at most n linear relations, implying by the pigeonhole principle that eventually it has to return a valid proof. However, note that using $\mathcal{A}$ to break the standard notion of soundness is not as trivial since in that case no simulator oracle is provided.

Although the construction of $\mathcal{A}$ is rather technical, we simply adapt the approach of Sections 3.2 and 3.3. First, we describe an adversary $\mathcal{B}$ that on input (crs, x) either return a proof or, with one simulation query, finds a linear relation among the group elements in (crs, x). Next, using $\mathcal{B}$ we build $\mathcal{A}$ which ensures that the linear relation found is non-trivial for those elements in the crs with probability $1 - 1/p$.

Proof of Lemma 5.3.1. In order to provide a description of $\mathcal{A}$, we begin by building a signature scheme whose message space consists of only one element from a NIZK. This will allow us using the adversary $\mathcal{B}$ described in Section 3.3 (Figure 3.4) which, on an algebraic signature scheme with a single message, either produces a forgery or finds a linear relation among the group elements of the verification key. The idea is, given a NIZK for an hard subset membership problem, to set the verification key as the crs and a false statement x, and the signing key is the simulation trapdoor td. A signature for the only message 0 is then any proof π for x. In this way the signer can create a proof for x using the simulation trapdoor, while no adversary can provide a proof for x unless soundness does not hold. A full description of the scheme is presented in Fig. 5.7.

Given this signature scheme, calling $x = (\mathbf{Z}, x') \in \mathbb{G}^m \times \{0,1\}^*$ and $\mathsf{crs} = (\mathbf{Y}, c') \in \mathbb{G}^n \times \{0,1\}^*$, we claim:

$\mathsf{S.Setup}(1^\lambda)$:

1 : Sample $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{S}(1^\lambda)$, $x \leftarrow^{\$} \mathsf{SampBad}(1^\lambda)$
2 : Set $\mathsf{vk} \leftarrow (\mathsf{crs}, x)$, $\mathsf{sk} \leftarrow \mathsf{td}$ and Return $(\mathsf{vk}, \mathsf{sk})$

$\mathsf{S.Sign}(\mathsf{sk}, 0)$:

1 : Return $\pi \leftarrow \mathsf{S}(\mathsf{td}, x)$

$\mathsf{S.Vfy}(\mathsf{vk}, 0, \pi)$:

1 : Return $b \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi)$

Figure 5.7: Signature scheme from any NIZK for a hard subset membership problem.

Claim 5.3.1. *There exists a* GPPT *adversary* $\mathcal{B}$ *such that, given* $V \leq \mathbb{F}_q^n$ *and* $W \leq \mathbb{F}_q^m$ *containing respectively the discrete logarithm of* $\mathbf{Y}$ *and* $\mathbf{Z}$*, either*

- $\mathcal{B}(V, W, \mathsf{vk}) \to (\pi, L)$*, with* π *a valid forgery*
- $\mathcal{B}(V, W, \mathsf{vk})$ *queries a signature for* 0 *and upon receiving a valid* π*, such that*

$$\mathcal{B}(V, W, \mathsf{vk}) \to (\bot, L) \quad \Rightarrow \quad L \lneq V \times W, \quad (\mathbf{Y}, \mathbf{Z}) \in L \cdot G.$$

Note that this adversary almost satisfies the property we wish $\mathcal{A}$ to have. However, when no forgery is found, it instead returns a linear relation among all the group elements in the verification key $(\mathbf{Y}, \mathbf{Z})$ and not only $\mathbf{Y}$. This means that linear relations found could be trivial in $\mathbf{Y}$.

In order to refine this adversary we the same technique introduced for the attacks in Chapter 3: The idea is to run $\mathcal{B}$ several times in a simulated environment with the real statement x and a fresh crs^* generated with a trapdoor td^*. In this simulation either $\mathcal{A}$ returns a *bad* L from which $\mathcal{B}$ can extract a non-trivial linear relation among the elements of x, or for sufficiently many times L satisfies

$$L \cap (\mathbb{F}_q^n \times \{\mathbf{z}\}) \ \lneq \ V \times \{\mathbf{z}\}$$

If this is the case, then $\mathcal{A}$ executes $\mathcal{B}$ one last time with the real crs it receives and, if needed, replies to the signature query from $\mathcal{B}$ using its only simulation query. Since the space L satisfied the above property for sufficiently many iterations, it will likely (but not *overwhelmingly*) be satisfied also in the last execution.

One issue with this approach though is that $\mathbf{z}$ is not known to $\mathcal{A}$, so testing $L \cap (\mathbb{F}_q^n \times \{\mathbf{z}\}) \lneq V \times \{\mathbf{z}\}$ might be hard. The next claim address this.

Claim 5.3.2. *Given* $V \leq \mathbb{F}_q^n$*,* $W \leq \mathbb{F}_q^m$ *and* $L \leq V \times W$ *affine spaces and calling* $\eta_2 : \mathbb{F}_q^n \times \mathbb{F}_q^m \to \mathbb{F}_q^m$ *the projection on the second entry*[4]*, then for all* $\mathbf{z} \in W$

$$L \leq V \times W, \quad \eta_2(L) = W \quad \Rightarrow \quad L \cap (\mathbb{F}_q^n \times \{\mathbf{z}\}) \lneq V \times \{\mathbf{z}\}.$$

We are now ready to give a description of the adversary $\mathcal{A}$ parametrized by a polynomially bounded t, which appears in Fig. 5.8.

[4]i.e. $\eta_2(\mathbf{x}, \mathbf{y}) = \mathbf{y}$. Typically projections are denoted with π, but we have to depart from this notation to avoid any confusion as π already denotes proofs.

$\mathcal{A}_t(V, \mathsf{crs}, x)$:

Initialize $W \leftarrow \mathbb{F}_q^m$ the space of possible exponents for $\mathbf{Z}$
For $j \in \{1, \dots, m+1\}$:
  // Each execution tries to reduce $\dim W$
  Set $W' \leftarrow W$ an affine space storing information on x gathered later on
  For $i \in \{1, \dots, t\}$:
    Sample $(\mathsf{crs}^*, \mathsf{td}^*) \leftarrow \mathsf{S}(1^\lambda)$ with $\mathsf{crs}^* = (\mathbf{Y}^*, c^*)$ and $\mathbf{Y}^* \in V \cdot G$
    Run $\mathcal{B}(V, W, \mathsf{crs}^*, x)$
    When $\mathcal{B}$ queries a signature for 0:
      Compute a simulated proof $\pi \leftarrow \mathsf{S}(\mathsf{td}^*, x)$ and send it $\mathcal{B} \leftarrow \pi$
    When $\mathcal{B}$ returns (π, L):
      If $\eta_2(L) \neq W$: Store the result $W' \leftarrow \eta_2(L)$
  If $W' \lneq W$: Update $W' \leftarrow W$
  Else: Break the outer for-loop
// Execute $\mathcal{B}$ one last time with the real crs
Run $\mathcal{B}(V, W, \mathsf{crs}, x)$
When $\mathcal{B}$ queries a signature for 0:
  Return query and on input π send $\mathcal{B} \leftarrow \pi$
When $\mathcal{B}$ returns (π, L)
  If π is a valid proof: Return π
  Elif $\eta_2(L) = W$: Return L
  Else: Return $\perp$

Figure 5.8: Adversary $\mathcal{A}_t$ parametrized by $t = \mathsf{poly}(\lambda)$.

Given this algorithm we break the proof that $\mathcal{A}_t$ is indeed the right algorithm for some t into the following claims.

Claim 5.3.3. *$\mathcal{A}$ is* GPPT.

Claim 5.3.4. *For each step of the execution of $\mathcal{A}$, calling $x = (\mathbf{Z}, x') \in \mathbb{G}^m \times \{0,1\}^*$, then $\mathbf{Z} \in W \cdot G$.*

Claim 5.3.5. *For any choice of $t = \mathsf{poly}(\lambda)$*

$$\Pr\left[\mathcal{A}(V, \mathsf{crs}, x) \to \perp\right] \leq \frac{m+1}{t+1}.$$

These claims imply the thesis since $\mathcal{A}$ only uses the group efficiently and, setting $t = (m+1) \cdot p(\lambda) - 1$, aborts with probability $p(\lambda)^{-1}$. Finally, if it does not abort either it returns a proof π for x, which happens without performing any query since $\mathcal{A}$ ask for a proof if and only if $\mathcal{B}$ ask for a signature, or it outputs L with $\eta_2(L) = W$. In this latter case, since $\mathbf{z} \in W$

by Claim 5.3.4, Claim 5.3.2 implies that

$$L \cap (\mathbb{F}_q^n \times \{\mathbf{z}\}) \lneq V \times \{\mathbf{z}\}$$

where we used the fact that by Claim 5.3.1, $\mathcal{A}$ returns $L \subseteq V \times W$.

Proof of Claim 5.3.2. By contradiction assume $L \cap \mathbb{F}_q^n \times \{\mathbf{z}\} = V \times \{\mathbf{z}\}$. Then for all $(\mathbf{v}, \mathbf{w}) \in V \times W$, since $\eta_2(L) = W$ there exists a point $\mathbf{u} \in V$ such that $(\mathbf{u}, \mathbf{w}) \in L$. Using the initial hypothesis we also have that

$$(\mathbf{u}, \mathbf{z}), (\mathbf{v}, \mathbf{z}) \in L \quad \Rightarrow \quad (\mathbf{u}, \mathbf{w}) + (\mathbf{v}, \mathbf{z}) - (\mathbf{u}, \mathbf{z}) \in L \quad \Rightarrow \quad (\mathbf{v}, \mathbf{w}) \in L.$$

Hence $V \times W \leq L$ which is a contradiction as we assumed $L \lneq V \times W$. □

Proof of Claim 5.3.3. Since t and m are polynomially bounded, it suffice to show that each individual line can be computed in GPPT time. This is evident for all commands with the exception of Line 3. That step however can be computed inefficiently, but with only polynomially many group operations. This is done first computing the conditional distribution of the *exponents* of the crs, conditioned to $\mathbf{Y} \in V \cdot G$, sampling from this distribution (which take exponential space), and finally computing crs from the sampled exponents, which takes polynomially many group operations. □

Proof of Claim 5.3.4. We proceed by induction. Initially $\mathbf{Z} \in W \cdot G$ since $W = \mathbb{F}_q^m$. Next assume that until the i-th iteration of the outer for-loop, $\mathbf{Z} \in W \cdot G$. At the end of the loop either $W' = W$, implying that the thesis still holds, or $W' \neq W$. In this second case $W' = \eta_2(L)$ with L being the output of $\mathcal{B}(V, W, \mathsf{crs}^*, x)$. Since by hypothesis $\mathbf{Y} \in V \cdot G$ and $\mathbf{Z} \in W \cdot G$, we have that $(\mathbf{Y}, \mathbf{Z})$ is contained in $V \times W$. Calling $\mathbf{y}, \mathbf{z}$ the discrete logarithm respectively of $\mathbf{Y}$ and $\mathbf{Z}$, we have that

$$(\mathbf{y}, \mathbf{z}) \in L \quad \Rightarrow \quad \mathbf{z} = \eta_2(\mathbf{y}, \mathbf{z}) \in \eta_2(L) = W' \quad \Rightarrow \quad \mathbf{Z} \in W' \cdot G. \qquad \square$$

Proof of Claim 5.3.5. We define J a random variable denoting the value index j has before terminating the outer loop by executing Line 12. Note that J is well defined since each time Line 12 is not executed, setting $W \leftarrow W'$ reduces the dimension of W by 1 which can only happens at most m times.

Next we define the event $\mathsf{E}_{i,j}$ as $J \geq j$ and at the i-th iteration of the inner for loop, the condition of Line 10 is not satisfied. We further let Fail be the event $\mathcal{A}(V, \mathsf{crs}, x) \to \perp$. Note that the events $\mathsf{E}_{i,j}$ are also well defined since either $J < j$ or when $J \geq j$ the inner loop is executed for all i.

Next we observe that, for each j, the inner loop runs $\mathcal{A}$ with the same vector spaces (V, W) and equally distributed crs^* and x. Thus we have that $\Pr[\mathsf{E}_{i,j}]$ is constant for all j, allowing us to define

$$p_j \;=\; \Pr[E_{1,j}] \;=\; \ldots \;=\; \Pr[E_{t,j}].$$

Next, we observe that conditioning on $J = j$, Fail occurs if and only if $\mathcal{B}(V, W, \mathsf{crs}, x)$ returns a vector space L satisfying $\eta_2(L) = W$ with crs following the same distribution simulated in Line 3. Thus

$$\Pr[\mathcal{A}(V, \mathsf{crs}, x) \to \perp \mid J = j] \;=\; 1 - p_j$$

In conclusion

$$\begin{aligned}\Pr\left[\mathcal{A}(V,\mathsf{crs},x)\to\perp\right] &\leq \sum_{j=1}^{m+1}\Pr\left[\mathcal{A}(V,\mathsf{crs},x)\to\perp \mid J=j\right]\Pr\left[J=j\right]\\ &\leq \sum_{j=1}^{m+1}(1-p_j)\Pr\left[\mathsf{E}_{1,j}\ \wedge\ \ldots\ \wedge\ \mathsf{E}_{t,j}\right]\\ &\leq \sum_{j=1}^{m+1}(1-p_j)\cdot p_j^t \ \leq\ \sum_{j=1}^{m+1}\frac{1}{t+1}\ \leq\ \frac{m+1}{t+1}.\end{aligned}$$

where in the third step we used the fact that the events $E_{i,j}$ for a fixed j are mutually independent and in the second to last step, we upper bound $(1-p_j)\cdot p_j^t \leq (t+1)^{-1}$ since $p_j \in [0,1]$. □

This concludes the Lemma's proof. □

5.3.3 Attack Description

As mentioned, the main difficulty of using $\mathcal{A}$ to break soundness is the absence of a simulator oracle. In this section we explain how to circumvent this issue, describing an adversary $\mathcal{Z}$ that breaks soundness using $\mathcal{A}$, and eventually derive our second impossibility result for algebraic NIZKs.

The core idea is that NIZKs for hard subset problem allow to produce proofs in two indistinguishable ways, that is either

1. sampling $\mathsf{crs} \leftarrow \mathsf{G}(1^\lambda)$, $(x,w) \leftarrow^{\$} \mathsf{SampGood}(1^\lambda)$ and producing the proof using P and the witness w
2. sampling $(\mathsf{crs},\mathsf{td}) \leftarrow \mathsf{S}(1^\lambda)$, $x \leftarrow^{\$} \mathsf{SampBad}(1^\lambda)$ and producing the proof using the simulator.

Thus, assuming we were able to predict whether $\mathcal{A}$ is going to return proof or query, our adversary $\mathcal{Z}$ could

1. sample $(x,w) \leftarrow \mathsf{SampGood}(1^\lambda)$ when $\mathcal{A}$ is going to ask a query. In this way it can simulate $\mathsf{S}(x)$ with $\mathsf{P}(x,w)$ and get a linear relation on the CRS' elements.
2. sample $x \leftarrow \mathsf{SampBad}(1^\lambda)$ when $\mathcal{A}$ is going to return a proof π. In this way π proves a false statement and $\mathcal{Z}$ breaks soundness

Unfortunately we don't have a way to predict $\mathcal{A}$'s behavior. However, since the only difference in the two approaches above is how x is sampled, $\mathcal{A}$ cannot distinguish between them. Hence by flipping a random coin $\mathcal{Z}$ can guess $\mathcal{A}$'s reply and act accordingly. Since $\mathcal{A}$ replies almost independently from $\mathcal{Z}$'s choice, its guess is correct with probability close to $1/2$. Amplifying this in a way that makes $\mathcal{Z}$ guess correctly at least $n+1$ times allows us to conclude that $\mathcal{A}$ proves a false statement at least once, because at most n linear relations can be found on the CRS' elements. A complete description of $\mathcal{Z}$ appears in Fig. 5.9.

We remark that the computation of $\mathbf{z}$ in line 5 can be done in polynomial time since $\mathsf{SampBad}$ and $\mathsf{SampGood}$ are generic algorithm: Therefore, by reading their queries to the GGM oracles, it is possible to locally store the discrete logarithm in base G of any queried group element during their execution, and in particular of output's group elements.

$\mathcal{Z}(\mathsf{crs})$:

Initialize $V \leftarrow \mathbb{F}_q^n$ and $\pi^* \leftarrow \perp$
For $i \in \{1, \ldots, \lambda(n+1)\}$: // $\lambda(n+1)$ iterations to guess correctly $n+1$ times
  Sample $\beta_i \leftarrow^{\$} \{0,1\}$
  If $\beta_i = 0$: $x \leftarrow \mathsf{SampBad}(1^\lambda)$; **Else** $(x, w) \leftarrow \mathsf{SampGood}(1^\lambda)$
  Parse $x = (\mathbf{Z}, z') \in \mathbb{G}^m \times \{0,1\}^*$ and get $\mathbf{z}$ such that $\mathbf{Z} = \mathbf{z} \cdot G$
  If $\mathcal{A}(V, \mathsf{crs}, x) \rightarrow \mathsf{query}$:
    If $\beta_i = 0$: Continue the for loop
    Else:
      Create a proof $\pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w)$
      Get $\mathcal{A}(V, \mathsf{crs}, x, \pi) \rightarrow L$ and let V' be s.t. $L \cap (\mathbb{F}_q^n \times \{\mathbf{z}\}) = V' \times \{\mathbf{z}\}$
      Update $V \leftarrow V'$
  Elif $\mathcal{A}(V, \mathsf{crs}, x) \rightarrow (\mathsf{proof}, \pi)$:
    If $\beta_i = 0$ and $1 \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi)$: store $\pi^* \leftarrow \pi$
Return π^*

Figure 5.9: GPPT Adversary $\mathcal{Z}$ breaking soundness using $\mathcal{A}$ from Lemma 5.3.1.

Theorem 5.3.2. *Let* $(\mathcal{R}, \mathsf{SampGood}, \mathsf{SampBad})$ *be a subset membership problem hard against* GPPT *adversaries. Then there exists no algebraic NIZK for* $\mathcal{R}$*.*

Proof. We show that given a complete and zero-knowledge non-interactive argument, $\mathcal{Z}$ breaks soundness. First let us fix some notation. $\mathbf{Y}$ will be the vector of group elements in crs, i.e. $\mathsf{crs} = (\mathbf{Y}, c') \in \mathbb{G}^n \times \{0,1\}^*$. $\mathcal{A}$ will be the adversary from Lemma 5.3.1 chosen with failure probability

$$\frac{1}{p(\lambda)} = \frac{1}{4\lambda(n+1)}$$

and for the i-th execution of the for-loop in $\mathcal{Z}$ we define the events:

$$\begin{aligned}
\mathsf{GoodProof}_i &: \beta_i = 0 \text{ and } \mathcal{A}(V, \mathsf{crs}, x) \rightarrow (\mathsf{proof}, \pi) \\
\mathsf{BadProof}_i &: \beta_i = 1 \text{ and } \mathcal{A}(V, \mathsf{crs}, x) \rightarrow (\mathsf{proof}, \pi) \\
\mathsf{GoodQuery}_i &: \beta_i = 1 \text{ and } \mathcal{A}(V, \mathsf{crs}, x) \rightarrow \mathsf{query} \\
\mathsf{BadQuery}_i &: \beta_i = 0 \text{ and } \mathcal{A}(V, \mathsf{crs}, x) \rightarrow \mathsf{query} \\
\mathsf{Bad}_i &: \mathsf{BadProof}_i \vee \mathsf{BadQuery}_i \\
\mathsf{Fail}_i &: \mathcal{A}(V, \mathsf{crs}, x) \rightarrow \perp \text{ or } \mathbf{Y} \notin V \cdot G
\end{aligned}$$

We further define Fail the event $\exists i : \mathsf{Fail}_i$. Next, we break the proof into the following sequence of claims.

Claim 5.3.6. $\Pr[\mathsf{Fail}] \leq 1/2$.

Claim 5.3.7. *The probability of happening λ sequential* Bad *events is negligible, i.e. there exist a negligible ε_0 such that*

$$\forall j_0 \leq n\lambda \qquad \Pr\left[\bigwedge\nolimits_{i=1}^{\lambda} \mathsf{Bad}_{j_0+i} \;\middle|\; \neg\mathsf{Fail}\right] \leq \varepsilon_0.$$

Claim 5.3.8. *The probability that* $\neg\mathsf{Bad}$ *occurs less than $n+1$ times is negligible, i.e.*

$$\Pr\left[\,|\{i \;:\; \mathsf{Bad}_i\}| \leq n \mid \neg\mathsf{Fail}\right] \;\leq\; (n+1)\cdot\varepsilon_0.$$

Claim 5.3.9. *If* $\mathsf{GoodQuery}_i$ *occurs, then at step 11 of Fig. 5.9, the dimension of V decreases with overwhelming probability, i.e. there exists a negligible ε_1 such that*

$$\Pr\left[\mathsf{GoodQuery}_i \;\wedge\; \neg(V' \lneq V) \mid \neg\mathsf{Fail}\right] \;\leq\; \varepsilon_1.$$

Claim 5.3.10. *If* $\mathsf{GoodProof}_i$ *occurs, then at step 13 of Fig. 5.9, the proof π is correct with overwhelming probability, i.e. there exists a negligible ε_2 such that*

$$\Pr\left[\mathsf{GoodProof}_i \;\wedge\; 0 \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi) \mid \neg\mathsf{Fail}\right] \;\leq\; \varepsilon_2.$$

Before proving these claims we show they imply that with significant probability $\mathcal{Z}$ produces a proof for a false statement. From Claim 5.3.8, $1-(n+1)\varepsilon_0 \leq$

$$\begin{aligned}
&\leq \Pr\left[\exists i_1, \ldots, i_{n+1} \;:\; \neg\mathsf{Bad}_{i_j} \;\middle|\; \neg\mathsf{Fail}\right] \\
&\leq \Pr\left[\exists i_1, \ldots, i_{n+1} \;:\; \mathsf{GoodQuery}_{i_j} \;\middle|\; \neg\mathsf{Fail}\right] + \Pr\left[\exists i \;:\; \mathsf{GoodProof}_i \mid \neg\mathsf{Fail}\right].
\end{aligned}$$

Regarding the first term, if $\mathsf{GoodQuery}$ occurs $n+1$ times, in at least one of these events the affine space returned by $\mathcal{A}$ does not yield $V' < V$, because the dimension of V can decrease at most n times. Hence, calling wrong_i the event $\neg(V' < V)$ at iteration i, we have that for some j, wrong_{i_j} occurs. Then

$$\begin{aligned}
&\Pr\left[\exists i_1, \ldots, i_{n+1} \;:\; \mathsf{GoodQuery}_{i_j} \;\middle|\; \neg\mathsf{Fail}\right] \\
&= \Pr\left[\exists i_1, \ldots, i_{n+1} \;:\; \mathsf{GoodQuery}_{i_j} \quad\wedge\quad \exists j : \mathsf{wrong}_{i_j} \;\middle|\; \neg\mathsf{Fail}\right] \\
&\leq \Pr\left[\exists i \;:\; \mathsf{GoodQuery}_i \wedge \mathsf{wrong}_i \mid \neg\mathsf{Fail}\right] \\
&\leq \textstyle\sum_{i=1}^{\lambda(n+1)} \Pr\left[\mathsf{GoodQuery}_i \;\wedge\; \mathsf{wrong}_i \mid \neg\mathsf{Fail}\right] \;\leq\; \lambda(n+1)\varepsilon_1.
\end{aligned}$$

Regarding the second term, calling valid_i the event that a proof returned at step i is accepted by the verifier.

$$\begin{aligned}
&\Pr\left[\exists i \;:\; \mathsf{GoodProof}_i \mid \neg\mathsf{Fail}\right] \\
&\leq \Pr\left[\exists i \;:\; \mathsf{GoodProof}_i \wedge \mathsf{valid}_i \mid \neg\mathsf{Fail}\right] + \Pr\left[\exists i \;:\; \mathsf{GoodProof}_i \wedge \neg\mathsf{valid}_i \mid \neg\mathsf{Fail}\right] \\
&\leq \Pr\left[1 \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi^*) \mid \neg\mathsf{Fail}\right] + \lambda(n+1)\varepsilon_2
\end{aligned}$$

Combining this two upper bounds together we get that $\mathcal{Z}$ returns a correct proof with probability negligibly close to 1/2.

$$\begin{aligned}
\Pr\left[1 \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi^*)\right] \;&\geq\; (1-(n+1)(\varepsilon_0+\lambda\varepsilon_1+\lambda\varepsilon_2))\cdot\Pr\left[\neg\mathsf{Fail}\right] \\
&\geq \frac{1-(n+1)(\varepsilon_0+\lambda\varepsilon_1+\lambda\varepsilon_2)}{2}.
\end{aligned}$$

□

Proof of Claim 5.3.6. Calling $\varepsilon_{\mathsf{zk}}$ the advantage of distinguishing a crs generated by G from one produced by S and $\varepsilon_{\mathcal{R}}$ the advantage of guessing an instance of a hard subset membership problem, we will show that

$$\Pr\left[\mathsf{Fail}_i \mid \neg\mathsf{Fail}_1 \wedge \ldots \wedge \neg\mathsf{Fail}_{i-1}\right] \;\leq\; \frac{1}{4\lambda(n+1)} + 2(\varepsilon_{\mathsf{zk}} + \varepsilon_{\mathcal{R}}).$$

Summing all this $\lambda(n+1)$ terms will give an upper bound $\Pr[\mathsf{Fail}] \leq 1/4 + \mathsf{negl}(\lambda)$ that for sufficiently large values of λ is less that $1/2$. To show this we study two cases:

- $\beta_i = 0$. Then $\mathcal{A}$ receives (V, crs, x) with $\mathsf{crs} \leftarrow \mathsf{G}(1^\lambda)$ and $x \leftarrow \mathsf{SampBad}(1^\lambda)$. By Zero-Knowledge, we have that any $\mathcal{D}$ distinguishing (crs_0, π_0) generated with G and P from (crs_1, π_1) generated by S has advantage at most $\varepsilon_{\mathsf{zk}}$. This holds for any statement, and in particular also for (x, w) chosen by $\mathcal{D}$ (not depending on the crs).

  Next we sketch a distinguisher $\mathcal{D}$ using $\mathcal{A}$. Initially $\mathcal{D}$ samples (x_i, w_i) from $\mathsf{SampGood}$, set it as the challenge statement and receives (crs, π_i) either generated correctly using w_i or simulated. For the first $i-1$ rounds $\mathcal{D}$ behaves as $\mathcal{Z}$. At the i-th round if $\mathcal{A}$ outputs query it replies with π_i. If $\mathcal{A}$ fails $\mathcal{D}$ returns 1, otherwise it returns 0. When the (crs, π_i) is honestly generated, $\mathcal{A}$ fails with probability $1/p(\lambda)$ by Lemma 5.3.1. Hence when (crs, π_i) is simulated $\mathcal{A}$ fails with probability smaller than $1/p(\lambda) + \mathsf{Adv}(\mathcal{D}) \leq 1/p(\lambda) + \varepsilon_{\mathsf{zk}}$.

  In conclusion $\Pr\left[\mathsf{Fail}_i \mid \neg\mathsf{Fail}_1 \wedge \ldots \wedge \neg\mathsf{Fail}_{i-1}\right] =$

  $$\Pr\left[\mathcal{A}(V, \mathsf{crs}, x) \rightarrow \perp \middle| \bigwedge\nolimits_{j=1}^{i-1} \neg\mathsf{Fail}_j\right] \;\leq\; \frac{1}{p(\lambda)} + \varepsilon_{\mathsf{zk}} \;\leq\; \frac{1}{4\lambda(n+1)} + 2(\varepsilon_{\mathsf{zk}} + \varepsilon_{\mathcal{R}}).$$

- $\beta_i = 1$. Then $\mathcal{A}$ receives (V, crs, x) with $\mathsf{crs} \leftarrow \mathsf{G}(1^\lambda)$, $(x, w) \leftarrow \mathsf{SampGood}(1^\lambda)$. By Definition 5.3.1 the advantage of distinguishing (crs, x) from (crs, x') with $x' \leftarrow \mathsf{SampBad}(1^\lambda)$ is less than $\varepsilon_{\mathcal{R}}$. The previous argument allow us to conclude

  $$\Pr\left[\mathcal{A}(V, \mathsf{crs}, x) \rightarrow \perp \mid \neg\mathsf{Fail}_1 \wedge \ldots \wedge \neg\mathsf{Fail}_{i-1}\right] \leq \frac{1}{4\lambda(n+1)} + \varepsilon_{\mathsf{zk}} + \varepsilon_{\mathcal{R}}.$$

  Analogously, since $(\mathbf{Y}, \mathbf{Z}) \in L \cdot G$ if and only if $\mathbf{Y} \in V' \cdot G$, $\Pr[\mathbf{Y} \notin V' \cdot G] \leq \varepsilon_{\mathsf{zk}} + \varepsilon_{\mathcal{R}}$, or else $\mathcal{A}$ could be used as a distinguisher as shown before. Using a union bound yields again the claimed inequality.

□

Proof of Claim 5.3.7. We describe $\mathcal{M}$ using $\mathcal{A}$ to guess λ instances of a hard subset membership problem.

By inspection $\mathcal{M}$ perfectly emulates the behavior of $\mathcal{Z}$ for the first j_0 executions of the initial For-loop. Regarding the subsequent λ calls to $\mathcal{A}$ we proceed inductively assuming $\mathcal{M}$ correctly guessed all challenges and simulated $\mathcal{Z}$ until the $(i-1)$-th step. Let b'_i be the challenger's bit, such that if $b'_i = 0$ then x_i is generated with $\mathsf{SampBad}$ or else $\mathsf{SampGood}$ was used. When $b'_i = 0$, $\mathcal{M}$ correctly executes $\mathcal{A}(V, \mathsf{crs}, x_i)$ as $\mathcal{Z}$ would with $\beta_{j_0+i} = 0$. Similarly when b'_0, $\mathcal{M}$

$\mathcal{M}(x_1, \ldots, x_\lambda)$:

1 : Initialize $\mathsf{crs} \leftarrow \mathsf{G}(1^\lambda)$, $V \leftarrow \mathbb{F}_q^n$
2 : **For** j_0 times: // Behave as $\mathcal{Z}$
3 : Sample $\beta \leftarrow^{\$} \{0,1\}$
4 : **If** $\beta = 0$: $x \leftarrow \mathsf{SampBad}(1^\lambda)$; **Else** $(x, w) \leftarrow \mathsf{SampGood}(1^\lambda)$
5 : **If** $\mathcal{A}(V, \mathsf{crs}, x) \rightarrow \mathsf{query}$:
6 : **If** $\beta = 1$:
7 : Create a proof $\pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w)$
8 : Get $\mathcal{A}(V, \mathsf{crs}, x, \pi) \rightarrow V'$ and update $V \leftarrow V'$
9 : // After the For-loop, pass for λ times the challenges to $\mathcal{A}$
10 : **For** $i \in \{1, \ldots, \lambda\}$:
11 : **If** $\mathcal{A}(V, \mathsf{crs}, x_i) \rightarrow \mathsf{proof}$: Set $b_i \leftarrow 1$
12 : **Else**: Set $b_i \leftarrow 0$
13 : Return $(b_1, \ldots, b_\lambda)$

Figure 5.10: Reduction $\mathcal{M}$ guessing λ instances of a Hard Subset Membership Problem.

correctly run $\mathcal{A}(V, \mathsf{crs}, x_i)$ as $\mathcal{Z}$ would with $\beta_{j_0+i} = 1$. Thus, assuming $\neg\mathsf{Fail}$

$$b_i = b_i' \quad \Leftrightarrow \quad (b_i' = 0 \rightarrow b_i = 0) \ \wedge \ (b_i' = 1 \rightarrow b_i = 1) \quad \Leftrightarrow$$
$$\Leftrightarrow \quad \mathsf{BadQuery}_{j_0+i} \wedge \mathsf{BadProof}_{j_0+i} \quad \Leftrightarrow \quad \mathsf{Bad}_{j_0+i}.$$

As a consequence, if $\mathcal{M}$ correctly guesses b_i', not updating V keeps its behavior identical to $\mathcal{Z}$, which only updates V if $\mathsf{GoodQuery}_{j_0+i}$ occurs. Therefore

$$\begin{aligned} \Pr\left[b_i = b_i',\ i \in [\lambda]\right] \ &\geq \ \Pr\left[\neg\mathsf{Fail}\right] \cdot \Pr\left[b_i = b_i',\ i \in [\lambda] \quad \mid \neg\mathsf{Fail}\right] \\ &= \ \Pr\left[\neg\mathsf{Fail}\right] \cdot \Pr\left[\neg\mathsf{Bad}_{j_0+i},\ i \in [\lambda] \mid \neg\mathsf{Fail}\right] \end{aligned}$$

Since the probability of $b_i' = b_i$ for all i is negligible, and by Claim 5.3.6 $\Pr\left[\neg\mathsf{Fail}\right] \geq 1/2$ we conclude that the claim is true. □

Proof of Claim 5.3.8. If $\neg\mathsf{Bad}$ occurs less than $n+1$ times, by the pigeonhole principle for at least one of the intervals $I_k = \{\lambda k + 1, \ldots, \lambda(k+1)\}$ Bad_i occurs for all $i \in I_k$. A union bound yields

$$\begin{aligned} \Pr\left[\left|\{i \ : \ \neg\mathsf{Bad}_i\}\right| < n+1 \ \middle| \ \neg\mathsf{Fail}\right] \ &\leq \ \textstyle\sum_{k=0}^{n} \Pr\left[\forall i \in I_k,\ \mathsf{Bad}_i \mid \neg\mathsf{Fail}\right] \\ &\leq \ (n+1) \cdot \varepsilon(\lambda). \end{aligned}$$

□

Proof of Claim 5.3.9. We first observe that if (crs, x) is generated with S and $\mathsf{SampBad}$, if $\mathcal{A}(V, \mathsf{crs}, x) \rightarrow \mathsf{query}$ the affine space L it returns satisfies by Lemma 5.3.1 $L \cap (\mathbb{F}_q^n \times \{\mathbf{z}\}) \subsetneq V \times \{\mathbf{z}\}$. By definition then

$$V' \times \{\mathbf{z}\} \ = \ L \cap \left(\mathbb{F}_q^n \times \{\mathbf{z}\}\right) \ \subsetneq \ V \times \{\mathbf{z}\} \quad \Rightarrow \quad V' \subsetneq V.$$

Therefore, borrowing notation from the proof of Claim 5.3.6, when $\beta_i = 1$, the probability that $\mathsf{GoodQuery}_i \wedge \neg(V' \lneq V)$ is smaller than $\varepsilon_{\mathsf{zk}} + \varepsilon_{\mathcal{R}}$, or else $\mathcal{A}$ could be used to distinguish (crs, x) from (crs', x') respectively generated with $\mathsf{G}, \mathsf{SampGood}$ and $\mathsf{S}, \mathsf{SampBad}$. Finally since $\neg\mathsf{Fail}$ occurs with significant probability,

$$\begin{aligned}\Pr\left[\mathsf{GoodQuery}_i \ \wedge \ \neg(V' \lneq V) \mid \neg\mathsf{Fail}\right] \ &\leq \ \frac{\Pr\left[\mathsf{GoodQuery}_i \ \wedge \ \neg(V' \lneq V)\right]}{\Pr\left[\neg\mathsf{Fail}\right]} \\ &\leq \ \frac{\varepsilon_{\mathsf{zk}} + \varepsilon_{\mathcal{R}}}{\Pr\left[\neg\mathsf{Fail}\right]} \ \leq \ \frac{\varepsilon_{\mathsf{zk}} + \varepsilon_{\mathcal{R}}}{2}.\end{aligned}$$

□

Proof of Claim 5.3.10. Analogous to the proof of Claim 5.3.9. □

Chapter 6

Secret Reconstruction in the Generic Action Model

Chapter Overview

In this chapter we move our attention from generic groups to generic group actions (see Sections 2.1.3 and 2.1.4 for definitions). In this context, due to the limited algebraic structure, many primitives and protocols cannot be instantiated as efficiently as we could over prime order groups. Examples of this includes Schnorr-like sigma-protocols, and signatures [20]. The problem we will focus on is, given a secret shared value s, how to reconstruct its actions $s \star E$ on a given set element $E \in \mathcal{E}$. t The main results we prove here are Theorem 6.1.5 and 6.2.4. The first one states that for any t out of n shared secret s, computing $s \star E$ in the GAM requires at least t rounds of interaction. The second one instead addresses somewhat *fair* protocols, where all parties receive the result in the last round. In this case we show that $\Omega(n \log n)$ total computation and communication is required, proving that the binary splitting protocol [48] is optimal.

Regarding our first result, we begin by studying general computation in the GAM and show that a *single party* computing $E_{\mathsf{out}} = s \star E_0$ must have reached such result through a chain of queries to $\mathcal{O}_{\mathsf{act}}$. This is formally stated in out *sequentiality lemma* (Lemma 6.1.1). Next we provide a generalization in the interactive case, see Lemma 6.1.4. There we show that in any t rounds protocol computing $s \star E_0$ there exists a chain of group actions starting from E_0 to E_{out} involving for each round queries performed by at most one user. Finally, using such Lemma we conclude that any $t-1$ round protocol to compute $s \star E_0$ only needs to involve $t-1$ users, and those users can thus reconstruct the secret despite being not at authorized set.

Our second result instead follows from a graph-theoretic argument. At a high level we associate to each "optimal" protocol a tree. On the one hand we show the number of edges to be related to communication and computation costs. On the other hand, we prove such tree must satisfy what we call the *tall sub-tree property* (TSP), discussed in Section 6.2.3. The lower bound then follows as we show that TSP trees have $\Omega(n \log n)$ edges.

6.1 Round Lower-Bound

6.1.1 Sequentiality Lemma

Our starting point to study the number of rounds required to compute the action, of a secret-shared scalar will be the *Sequentiality Lemma*, stating that any procedure computing $s \star E_0$ can obtain the right result only through sequential applications of the group action to E_0.

First let us introduce some notation. Given $\mathcal{A}$ a PPT algorithm with oracle access to $\mathcal{O}_{\mathsf{act}}$ initially receiving E_0 and performing q queries, we denote these with $E_k \leftarrow \mathcal{O}_{\mathsf{act}}(a_k, D_k)$ for $k \in \{1, \dots, q\}$. Next we introduce a relation $\to$ among indices $\{0, \dots, q\}$.

$$k_1 \to k_2 \quad \Leftrightarrow \quad k_1 < k_2, \quad E_{k_1} = D_{k_2}.$$

Intuitively this means that the k_1-th set element was used to compute the k_2-th. Note this relation is not yet a (strict) partial order as it is not transitive, but its transitive closure is. This is explicitly defined as

$$i \to^+ j \quad \Leftrightarrow \quad \exists k_1, \dots, k_m : \quad k_1 \to k_2 \to \dots \to k_m, \quad k_1 = i, \quad k_m = j$$

The lemma then ensures that if an algorithm computes $s \star E_0$ for a known s in the GAM, with high probability $s \star E_0$ is the output of some query, say the k-th (meaning $s \star E_0 = E_k$), and $0 \to^+ k$.

Lemma 6.1.1. *For any $\mathcal{A}$ PPT algorithm with oracle access to $\mathcal{O}_{\mathsf{act}}$ making at most q queries and any $s \in \mathbb{G}$, such that $(s, E_{\mathsf{out}}) \leftarrow \mathcal{A}(E_0)$ then*

$$\Pr\left[E_{\mathsf{out}} = s \star E_0,\ \nexists k \left(E_{\mathsf{out}} = E_k,\ 0 \to^+ k\right)\right] \ \leq\ \varepsilon_{\mathsf{vec}}(2q) + \frac{1}{|\mathcal{E}| - (q+1)} \coloneqq \varepsilon_{\mathsf{seq}}(q)$$

where $\varepsilon_{\mathsf{vec}}(q)$ is the advantage of breaking the vectorization problem in q queries, see Definition 2.1.7.

Proof. In the following we will use $\mathcal{E} \subseteq \{0,1\}^\mu$ to denote the set of labels and $\mathcal{E}'$ the set in our group action, so that $\star : \mathbb{G} \times \mathcal{E}' \to \mathcal{E}'$ and $\sigma : \mathcal{E}' \to \mathcal{E}$ is the labeling function. We further denote with abuse of notation $\star : \mathbb{G} \times \mathcal{E} \to \mathcal{E}$ the action defined by $\mathcal{O}_{\mathsf{act}}$.

We begin proving that if E_{out} is not a label returned by $\mathcal{O}_{\mathsf{act}}$, then

$$\Pr\left[E_{\mathsf{out}} = s \star E_0 \,|\, E_{\mathsf{out}} \notin \{E_0, \dots, E_q\}\right] \ \leq\ \frac{1}{|\mathcal{E}| - (q+1)}.$$

This holds because if $s \star E_0 \in \{E_0, \dots, E_q\}$ or $E_{\mathsf{out}} \notin \mathcal{E}$ then the probability of $E_{\mathsf{out}} = s \star E_0$ is zero. Conversely, when these two events do not occur, $\mathcal{A}$ correctly guessed the label $s \star E_0$ given only the image of σ (the labeling function) on $q+1$ different points. Since $\sigma : \mathcal{E}' \to \mathcal{E}$ is uniformly sampled

$$\Pr\left[E_{\mathsf{out}} = s \star E_0 \,|\, E_{\mathsf{out}}, s \star E_0 \in \mathcal{E} \setminus \{E_i\}_{i=0}^q\right] \ \leq\ \frac{1}{|\mathcal{E}| - (q+1)}.$$

Next, we assume $E_{\mathsf{out}} = E_k$ for some $k \in \{0, \ldots, q\}$. To conclude we need to prove that the probability of $E_{\mathsf{out}} = k$ but not $0 \to^+ k$ is negligible. We do so by reducing to the discrete logarithm problem. The idea is that since $\to^+$ is a partial order on a finite set, there exists a minimal element $h \to^+ k$ that is non-zero. Since E_h was not obtained through the action oracle, the adversary must have randomly sampled its representation and in particular it has (almost) no information on its discrete logarithm in base E_0. However because $h \to^+ k$ the adversary has to know $a \in \mathbb{G}$ for which $a \star E_h = E_k = s \star E_0$, implying that $E_h = (-a + s) \star E_0$, i.e. that it can find the discrete logarithm of E_0.

More formally we provide a reduction $\mathcal{B}$ to the vectorization problem in the GAM described in Figure 6.1. At a high level $\mathcal{B}(E_0, H)$ lazily simulate a GAM oracle through a labeling function $\widetilde{\sigma}$.

Every time $\mathcal{A}$ makes a query to the GAM on a previously seen set element the query is stored in $\mathcal{Q}$. Conversely, if a query is done on the representation of a point $\widetilde{D}$ not yet obtained through other queries, $\mathcal{B}$ either rejects with the same probability $\mathcal{O}_{\mathsf{act}}$ would, or internally maps $\widetilde{D}$ to the set element $r \star H$ with r uniformly sampled in $\mathbb{G}$, storing the tuple (r, D) in a set $\mathcal{C}$, and then performs the requested query invoking $\mathcal{O}_{\mathsf{act}}$.

Finally, if $\mathcal{A}$ returns the right output $E_{\mathsf{out}} = E_k$ but $0 \not\to^+ k$, then $\mathcal{B}$ uses the information in $\mathcal{Q}$ and $\mathcal{C}$ to break the vectorization problem. To sum up the set $\mathcal{Q}$ and $\mathcal{C}$ respectively contain:

- $\mathcal{Q}$: GAM queries of the form (a, D, E), meaning that $E = a \star D$, as well as trivial relations of the form $(0, E, E)$.
- $\mathcal{C}$: Challenges (r, D) with $D = r \star H$, created when $\mathcal{A}$ query a point not previously seen.

Initially observe that $\mathcal{B}$ perfectly simulates the $\mathcal{O}_{\mathsf{act}}$ oracle since for previously queried elements, for which a representation $\widetilde{\sigma}$ was chosen, the operation is consistent with $\mathcal{O}_{\mathsf{act}}$, while queries on set element not previously obtained from the oracle do not gives errors with probability

$$p_i = \frac{|\mathrm{Im}\,(\widetilde{\sigma} \setminus \{\perp\})|}{2^\mu - |\mathsf{Dom}(\widetilde{\sigma})|}.$$

and in this case the obtained element is different from previously queried ones.

To continue, we break down the proof in the following sequence of claims:

Claim 6.1.1. *For all $(a, D_j, E_i) \in \mathcal{Q}$ then $a \star D_j = E_i$.*

Claim 6.1.2. *For all $(r, D_i) \in \mathcal{C}$ then $D_i = r \star H$.*

Claim 6.1.3. *Given $h \in \{1, \ldots, q\}$ that is minimal with respect to $\to^+$ then after the h-th query $(0, D_h, D_h) \in \mathcal{Q}$ and $(r, D_h) \in \mathcal{C}$.*

Claim 6.1.4. *Given $h \in \{1, \ldots, q\}$ that is minimal with respect to $\to^+$, then $h \to^+ k$ implies that $(a, D_h, E_k) \in \mathcal{Q}$.*

To see why these Claims implies the thesis, let us assume that the three condition on steps 18, 20 and 22 are false. Then $\widetilde{E}_{\mathsf{out}} = E_k$ and $0 \not\to^+ k$. Since $\to^+$ is a partial order on a finite

$\mathcal{B}^{\mathcal{O}_{\mathsf{act}}}(E_0, H)$

Setup a partial function $\widetilde{\sigma} : \{0,1\}^{\mu} \to \{0,1\}^{\mu} \cup \{\perp\}$, initially $\widetilde{\sigma} = \varnothing$
Sample $\widetilde{E}_0 \leftarrow^{\$} \{0,1\}^{\mu}$ and set $\widetilde{\sigma}(\widetilde{E}_0) = E_0$
Initialize two sets $\mathcal{Q} = \{(0, E_0, E_0)\}$, $\mathcal{C} = \varnothing$
Run $\mathcal{A}(\widetilde{E}_0)$
When $\mathcal{A}$ queries $\mathcal{O}_{\mathsf{act}}(a_i, \widetilde{D}_i)$:
  If $\widetilde{D}_i \notin \mathsf{Dom}(\widetilde{\sigma})$:
    Let p_i be the probability that $\widetilde{D}_i$ is a valid set element
    With probability $1 - p_i$: Set $\widetilde{\sigma}(\widetilde{D}_i) = \perp$
    Else:
      Sample $r_i \leftarrow^{\$} \mathbb{G}$ such that $r_i \star H \notin \{E_0, \dots, E_{i-1}\}$
      Set $D_i \leftarrow r_i \star H$ and $\widetilde{\sigma}(\widetilde{D}_i) = D_i$
      Store $\mathcal{C} \leftarrow \mathcal{C} \cup \{(r_i, D_i)\}$ and $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(0, D_i, D_i)\}$
  Get $D_i \leftarrow \widetilde{\sigma}(\widetilde{D}_i)$ and $E_i \leftarrow \mathcal{O}_{\mathsf{act}}(a_i, D_i)$
  For all $(b, D_j, D_i) \in \mathcal{Q}$ store $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(a_i + b, D_j, E_i)\}$
  Sample $\widetilde{E}_i \leftarrow^{\$} \{0,1\}^{\mu} \setminus \mathsf{Dom}(\widetilde{\sigma})$ and set $\widetilde{\sigma}(\widetilde{E}_i) = E_i$
  Answer the query with $\mathcal{A} \leftarrow \widetilde{E}_i$
When $(s, \widetilde{E}_{\mathsf{out}}) \leftarrow \mathcal{A}$:
  If $\widetilde{E}_{\mathsf{out}} \notin \{\widetilde{E}_0, \dots, \widetilde{E}_q\}$: Return $\perp$.
  Find k such that $\widetilde{E}_{\mathsf{out}} = \widetilde{E}_k$
  If $0 \nrightarrow^{+} k$: Return $\perp$
  Let $E_k \leftarrow \widetilde{\sigma}(\widetilde{E}_k)$ and find $(a, D_j, E_k) \in \mathcal{Q}$ and $(r, D_j) \in \mathcal{C}$
  If $E_k \neq \mathcal{O}_{\mathsf{act}}(s, E_0)$: Return $\perp$
  Return the group element $-(a + r) + s$

Figure 6.1: Reduction $\mathcal{B}$ using $\mathcal{A}$ to break the vectorization problem.

set, there exists a minimal $h \nrightarrow^{+} k$ and, importantly, $h \neq 0$. From Claim 6.1.4 we have that $(a, D_h, E_k) \in \mathcal{Q}$, which by Claim 6.1.1 means $E_k = a \star D_h$. Similarly by Claim 6.1.3, $(r, D_h) \in \mathcal{C}$ which by Claim 6.1.2 means $D_h = r \star H$. Since we assumed all three final checks to fail, we further have that $E_k = s \star E_0$, meaning that

$$s \star E_0 = E_k = a \star D_h = (a + r) \star H \quad \Rightarrow \quad H = (-(a + r) + s) \star E_0.$$

That is, $\mathcal{B}$ returns the correct value. Finally observe that the three final conditions are false if

$$\widetilde{E}_{\mathsf{out}} = s \star E_0 \ \wedge \ \widetilde{E}_{\mathsf{out}} = \widetilde{E}_k \ \wedge \ 0 \nrightarrow^{+} k.$$

Hence, switching to the original notation

$$\Pr\left[E_{\mathsf{out}} = s \star E_0, \ E_{\mathsf{out}} = E_k, \ 0 \nrightarrow^{+} k\right] \ \leq \ \mathsf{Adv}(\mathcal{B}) \ \leq \ \varepsilon(2q)$$

which completes the proof. □

Proof of Claim 6.1.1. By induction, initially $(0, E_0, E_0)$ satisfies the hypothesis. Furthermore the tuple $(0, D_i, D_i)$ added in step 12 also satisfies the hypothesis. Assume this holds for all elements in $\mathcal{Q}$ when $(a_i + b, D_j, E_i)$ is added. Then $(b, D_j, D_i) \in \mathcal{Q}$ implies $D_i = b \star D_j$ so $E_i = a_i \star D_i = (a_i + b) \star D_j$. □

Proof of Claim 6.1.2. Follows by construction. □

Proof of Claim 6.1.3. Since h is minimal, the h-th query $\widetilde{E}_h = a_h \star \widetilde{D}_h$ is such that $\widetilde{D}_h$ was never returned by $\mathcal{B}$ through the action queries, meaning that before the h-th query $\widetilde{D}_h \notin \mathsf{Dom}(\widetilde{\sigma})$. By construction then $(0, D_h, D_h)$ and (r_h, D_h) are added to $\mathcal{Q}$, $\mathcal{C}$ respectively. □

Proof of Claim 6.1.4. Since h is minimal, when D_h is queried, $(0, D_h, D_h) \in \mathcal{Q}$ and after $\mathcal{B}$ computes the correct query, (a_h, D_h, E_h) is added in $\mathcal{Q}$. Next, by definition of $\rightarrow^+$ there exists indices $i_1, \ldots, i_m$ with $i_1 = h$, $i_m = k$ such that $\widetilde{D}_{i_j} = \widetilde{E}_{i_j}$. By induction we will prove that for all j there exists some $a \in \mathbb{G}$ such that $(a, D_h, E_{i_j}) \in \mathcal{Q}$. Assuming this holds for j, when $(a_{i_{j+1}}, \widetilde{D}_{i_{j+1}})$ we have that $\widetilde{D}_{i_{j+1}} = \widetilde{E}_{i_j}$ so $D_{i_{j+1}} = \widetilde{\sigma}(\widetilde{D}_{i_{j+1}}) = E_{i_j}$. Thus by construction, in step 14 $\mathcal{B}$ will add $(a_{i_{j+1}}, D_h, E_{i_{j+1}})$. □

6.1.2 Interactive Protocols

The main limitation of the Sequentiality Lemma is that it only applies to a single machine. Here we introduce notation for interactive protocols in the GAM in order to extend in the next section this result to the interactive case.

An interactive protocol is defined by n PPT machines $P_1, \ldots, P_n$ with access to point-to-point (i.e. non-broadcast[1]) communication channels. We assume them to be synchronous and simultaneous i.e. such that messages from all parties are atomically sent at the beginning of each round and delivered at the end. To formally describe this model we initially call $\mathsf{trs}_0 = \perp$ and inductively define for initial inputs $x_1, \ldots, x_n$ the messages sent and transcript at round r as

$$M_{r,i} \leftarrow^{\$} P_i(x_i, \mathsf{trs}_{r-1}), \quad \mathsf{trs}_r = (\mathsf{trs}_{r-1}, M_{r,1}, \ldots, M_{r,n})$$

where $M_{r,i} = (M_{r,i}^{(1)}, \ldots, M_{r,i}^{(n)})$ is the tuple of messages sent by P_i, and in particular $M_{r,i}^{(j)}$ is the message P_i sends to P_j. To ensure parties only use message delivered to them, we assume P_j can only read entries of the form $M_{r,i}^{(j)}$ in trs for any r and i.

Regarding the interaction with the GAM we denote $E_{r,i,j} \leftarrow \mathcal{O}_{\mathsf{act}}(a_{r,i,j}, D_{r,i,j})$ the j-th query made by P_i to $\mathcal{O}_{\mathsf{act}}$ during the r-th round. As done in Section 6.1.1 we then define a relation $\twoheadrightarrow$ among the indices (r, i, j) and (r', i', j') which indicates that the result from the former was used as input in the latter. Formally

$$(r, i, j) \twoheadrightarrow (r', i', j') \quad \Leftrightarrow \quad (r < r' \ \vee \ (r = r', \ i = i', \ j < j')) \ \wedge \ E_{r,i,j} = D_{r',i',j'}.$$

This condition on the indices says that one query precedes another one only if it was performed on a previous round, or if both were asked in the same round *by the same party* P_i one before

[1]Even though broadcast could be achieved simply sending the same message to all parties, as we only focus on semi-honest adversaries.

the other. Like $\rightarrow$, the relation $\twoheadrightarrow$ is not yet a (strict) partial order. However its transitive closure $\twoheadrightarrow$ is. This is explicitly defined as $(r,i,j) \twoheadrightarrow^+ (r',i',j') \Leftrightarrow$

$$\begin{aligned}\Leftrightarrow\ & \exists (r_1,i_1,j_1),\dots,(r_t,i_t,j_t): \\ & (r_1,i_1,j_1) \twoheadrightarrow \dots \twoheadrightarrow (r_t,i_t,j_t), \quad (r_1,i_1,j_1)=(r,i,j), \quad (r_t,i_t,j_t)=(r',i',j').\end{aligned}$$

Finally, to further include E_0 in this relation, which might be technically never queried, we could either assume that parties initially query $E_0 \leftarrow \mathcal{O}_{\mathsf{act}}(0,E_0)$, or more simply say that

$$0 \twoheadrightarrow (r,i,j) \quad \Leftrightarrow \quad D_{r,i,j} = E_0.$$

We conclude this section with two elementary properties of $\twoheadrightarrow$.

Lemma 6.1.2. *Let* $(r_1,i_1,j_1) \twoheadrightarrow \dots \twoheadrightarrow (r_t,i_t,j_t)$. *Then*

$$E_{r_t,i_t,j_t} = (a_{r_t,i_t,j_t} + \dots + a_{r_1,i_1,j_1}) \star D_{r_1,i_1,j_1}.$$

Lemma 6.1.3. *Let* $0 \twoheadrightarrow (r_1,i_1,j_1) \twoheadrightarrow \dots \twoheadrightarrow (r_t,i_t,j_t)$. *Then*

$$|\{i_1,\dots,i_t\}| \leq |\{r_1,\dots,r_t\}|.$$

6.1.3 Interactive Sequentiality Lemma

In this section we state the Interactive Sequentiality Lemma, which extends Lemma 6.1.1. This applies to a set of parties $P_1,\dots,P_n$ each holding an input $s_i \in \{0,1\}^{\mathsf{poly}(\lambda)}$ and wishing to compute $f(s_1,\dots,s_n) \star E_0$ for a given function f. Informally it states that the result must, up to negligible probability, come from the sequential application of $\mathcal{O}_{\mathsf{act}}$ to E_0 *and* that such sequence of queries involves at most one player for each round.

Lemma 6.1.4. *Let* $P_1,\dots,P_n$ *be a* k *round protocol in the GAM and* f *a function such that given inputs* $s_1,\dots,s_n \sim \{0,1\}^{\mathsf{poly}(\lambda)}$ *there exists* P_i *which at round* k *returns* $E_{\mathsf{out}} = f(s_1,\dots,s_n) \star E_0$. *Then*

$$\Pr\left[\exists (r',i',j') \ : \ E_{\mathsf{out}} = E_{r',i',j'}, \quad 0 \twoheadrightarrow^+ (r',i',j')\right] \geq 1 - (k+1)\cdot \varepsilon_{\mathsf{seq}}(q)$$

where q *is an upper bound on the total number of queries performed.*

Proof. In order to bridge the Sequentiality Lemma in this context we wrap the execution of the distributed protocol within an environment Ω. This will take as input all s_i, execute parties and manage messages delivery. A full description of Ω appears in Figure 6.2

Because Ω forwards parties' queries to $\mathcal{O}_{\mathsf{act}}$ we can define a partial function $\xi : \mathbb{N}^3 \to \mathbb{N}$ so that the j-th query of P_i at round r corresponds to the $\xi(r,i,j)$-th query of Ω. From the correctness of the distributed protocol we have that $E_{\mathsf{out}} = f(s_1,\dots,s_n) \star E_0$. By Lemma 6.1.1, up to probability $\varepsilon_{\mathsf{seq}}(q)$, E_{out} is the output of the $\xi(r',i',j')$-th query and that $0 \rightarrow^+ \xi(r',i',j')$. This implies that E_{out} is the output of the j'-th query $P_{i'}$ performs at round r', but it is not enough to imply $0 \twoheadrightarrow^+ (r',i',j')$. Indeed, chains from E_0 to E_{out} may involve queries performed at the same round by different players, something $\twoheadrightarrow^+$ does not allow. This is addressed by the following claim.

$\Omega^{\mathcal{O}_{\mathsf{act}}}(s_1, \ldots, s_n)$

Set $\mathsf{trs}_0 \leftarrow \perp$
For all $r \in \{1, \ldots, k-1\}$:
  For all $i \in \{1, \ldots, n\}$:
    $M_{r,i} \leftarrow P_i^{\mathcal{O}_{\mathsf{act}}}(s_i, \mathsf{trs}_{r-1})$
  // Update the view at the end of a round
  $\mathsf{trs}_r \leftarrow \mathsf{trs}_{r-1} \cup \{M_{r,i}\}_{i=1}^n$
// For the last round, execute all parties until we get the output
For $i \in \{1, \ldots, n\}$:
  If $E_{\mathsf{out}} \leftarrow P_i^{\mathcal{O}_{\mathsf{act}}}(s_i, \mathsf{trs}_{k-1})$:
    Return $(f(s_1, \ldots, s_n),\ E_{\mathsf{out}})$

Figure 6.2: Environment Ω executing $P_1, \ldots, P_n$ to compute $E_{\mathsf{out}} = f(s_1, \ldots, s_n) \star E_0$.

Claim 6.1.5. *If $0 \rightarrow^+ \xi(r,i,j)$, then up to probability $r \cdot \varepsilon_{\mathsf{seq}}$ we have $0 \rightarrowtriangle^+ (r,i,j)$, i.e.*

$$\Pr\left[0 \rightarrow^+ \xi(r,i,j), \quad 0 \not\rightarrowtriangle^+ (r,i,j)\right] \leq r \cdot \varepsilon_{\mathsf{seq}}$$

We immediately observe this claim implies the thesis as

$$\begin{aligned}
&\Pr\left[\nexists\, r,i,j \ :\ E_{\mathsf{out}} = E_{r,i,j},\ \ 0 \rightarrowtriangle^+ (r,i,j)\right] \\
\leq\ &\Pr\left[\nexists\, r,i,j \ :\ E_{\mathsf{out}} = E_{r,i,j},\ \ 0 \rightarrow^+ \xi(r,i,j)\right] \ + \\
&+\ \Pr\left[E_{\mathsf{out}} = E_{r',i',j'},\ \ 0 \rightarrow^+ \xi(r',i',j'),\ \ 0 \not\rightarrowtriangle^+ (r',i',j')\right] \\
\leq\ &\varepsilon_{\mathsf{seq}} + r'\varepsilon_{\mathsf{seq}} \ \leq\ (r'+1)\varepsilon_{\mathsf{seq}} \ \leq\ (k+1)\varepsilon_{\mathsf{seq}}.
\end{aligned}$$

□

Proof of Claim 6.1.5. Proceeding by induction on r, the base case $r = 0$ is trivially true. Assume now the statement to be true for all $r' < r$. We then construct $\mathcal{A}$ computing $E_{r,i,j}$ which behaves as Ω for the first $r-1$ rounds. At round r it initially executes P_i and then the remaining $P_1, \ldots, P_n$. If it finds a path $0 \rightarrow^+ \xi(r,i,j)$, it computes α such that $E_{\mathsf{out}} = \alpha \star E_0$ and returns $(\alpha, E_{\mathsf{out}})$, otherwise it aborts. A full description appears in Figure 6.3.

First, as for Ω, we can define an indexing partial function $\eta : \mathbb{N}^3 \to \mathbb{N}$ such that the j-th query performed by P_i at round r is also the $\eta(r,i,j)$-th query of $\mathcal{A}$. By Sequentiality Lemma 6.1.1 we get

$$\Pr\left[0 \rightarrow^+ \xi(r,i,j),\ 0 \not\twoheadrightarrow^+ \eta(r,i,j)\right] \leq \varepsilon_{\mathsf{seq}}$$

where we used the fact that $0 \rightarrow \xi(r,i,j)$ implies that $\mathcal{A}$'s outputs satisfies $\alpha \star E_0 = E_{\mathsf{out}}$ and $\eta(r,i,j)$ is known to be the query index in which $\mathcal{O}_{\mathsf{act}}$ returns $E_{r,i,j}$.

Assuming instead $0 \rightarrow^+ \eta(r,i,j)$, let (r',i',j') the maximal element in the chain such that $r' < r$. Then we will show that

$$0 \rightarrow^+ \eta(r,i,j) \quad \Rightarrow \quad 0 \rightarrow^+ \xi(r',i',j') \ \wedge\ (r',i',j') \rightarrowtriangle^+ (r,i,j).$$

$\mathcal{A}^{\mathcal{O}_{\mathsf{act}}}(s_1, \ldots, s_n)$

// Behave as Ω for the first $r-1$ rounds
Set $\mathsf{trs}_0 \leftarrow \perp$
For all $r' \in \{1, \ldots, r-1\}$:
  For all $i' \in \{1, \ldots, n\}$:
    $M_{r,i'} \leftarrow P_{i'}^{\mathcal{O}_{\mathsf{act}}}(s_{i'}, \mathsf{trs}_{r'-1})$
  $\mathsf{trs}_{r'} \leftarrow \mathsf{trs}_{r'-1} \cup \{M_{r',i'}\}_{i'=1}^{n}$
// Execute P_i first at round r
Run $P_i^{\mathcal{O}_{\mathsf{act}}}(s_i, \mathsf{trs}_{r-1})$
For $i' \in \{1, \ldots, n\} \setminus \{i\}$: Run $P_{i'}^{\mathcal{O}_{\mathsf{act}}}(s_{i'}, \mathsf{trs}_{r-1})$
// Find a chain according to Ω's query order
If $0 \not\twoheadrightarrow^{+} \xi(r,i,j)$: **Return** $\perp$
Use a chain $0 \rightarrow^{+} \xi(r,i,j)$ to find α such that $E_{r,i,j} = \alpha \star E_0$
Return $(\alpha, E_{r,i,j})$

Figure 6.3: Program $\mathcal{A}$ computing $E_{r,i,j}$.

The first part of the implication follows since for all rounds before the r-th, $\mathcal{A}$ and Ω behaves identically. Therefore they make the same queries in the same order, meaning that $0 \rightarrow^{+} \eta(r',i',j')$ if and only if $0 \rightarrow^{+} \xi(r',i',j')$. For the second part, from the way we defined (r',i',j'), all queries in a chain from (r',i',j') to (r,i,j) occurs at round r. Since $\mathcal{A}$ first executes P_i at round r, it means all these queries only involves P_i , therefore $(r',i',j') \twoheadrightarrow^{+} (r,i,j)$.

As a consequence we can bound the studied probability in the case $0 \rightarrow^{+} \eta(r,i,j)$ as follows:

$$
\begin{aligned}
&\Pr\left[0 \rightarrow^{+} \xi(r,i,j),\ 0 \not\twoheadrightarrow^{+} (r,i,j),\ 0 \rightarrow^{+} \eta(r,i,j)\right] \\
\leq\ &\Pr\left[0 \rightarrow^{+} \xi(r,i,j),\ 0 \not\twoheadrightarrow^{+} (r,i,j),\ 0 \rightarrow^{+} \xi(r',i',j'),\ (r',i',j') \twoheadrightarrow^{+} (r,i,j)\right] \\
\leq\ &\Pr\left[0 \rightarrow^{+} \xi(r',i',j'),\ 0 \not\twoheadrightarrow^{+} (r',i',j')\right] \\
\leq\ &r' \cdot \varepsilon_{\mathsf{seq}}.
\end{aligned}
$$

where the last step follows by induction. Note that the second inequality follows because on LHS, if $0 \twoheadrightarrow^{+} (r',i',j')$ then by transitivity $0 \twoheadrightarrow^{+} (r,i,j)$, contradicting the clause $0 \not\twoheadrightarrow^{+} (r,i,j)$. The event on the LHS then implies $0 \not\twoheadrightarrow^{+} (r',i',j')$. In conclusion, since $r' < r$

$$
\begin{aligned}
&\Pr\left[0 \rightarrow^{+} \xi(r,i,j),\ 0 \not\twoheadrightarrow^{+} (r,i,j)\right] \\
\leq\ &\Pr\left[0 \rightarrow^{+} \xi(r,i,j),\ 0 \not\twoheadrightarrow^{+} \eta(r,i,j)\right] \ + \\
&\quad + \Pr\left[0 \rightarrow^{+} \xi(r,i,j),\ 0 \not\twoheadrightarrow^{+} (r,i,j),\ 0 \rightarrow^{+} \eta(r,i,j)\right] \\
\leq\ &\varepsilon_{\mathsf{seq}} + r'\varepsilon_{\mathsf{seq}} \ \leq\ r\varepsilon_{\mathsf{seq}}.
\end{aligned}
$$

□

6.1.4 Lower Bound

We now present our first result for distributed computation over black box HHS. We assume that parties $P_1, \ldots, P_n$ initially receive a secret input $s_i \in \{0,1\}^{\mathsf{poly}(\lambda)}$ and a public function f and execute a protocol compute $f(s_1, \ldots, s_n) \star E_0$ in Shoup's GAM. In this setting then we will prove that if such computation only requires k rounds, then a subset of k users can passively collude, and recover $f(s_1, \ldots, s_n)$.

Evaluating the group action of a t out of n secret shared group element $s \in \mathbb{G}$ is then a specific case of interest, as it affects threshold signatures and cryptosystems. This is captured by our result setting f as the reconstruction function. More concretely, for n out of n additive secret sharing, $f(s_1, \ldots, s_n)$ is simply the sum of all shares. Similarly, for t out of n Shamir secret sharing f is

$$f(s_1, \ldots, s_n) \;=\; \sum\nolimits_{i \in R} \lambda_{i,R} \cdot s_i$$

with R being a reconstruction set of size t and $\lambda_{i,R} \in \mathbb{Z}_N$ the Lagrange coefficients[2]. In all these cases, since any set with less than t users is not entitled to recover s, our result implies that passive security cannot be achieved with less than t rounds. Finally, this will further imply the optimality of round-robin protocols in such cases.

Theorem 6.1.5. *Let $P_1, \ldots, P_n$ be a k-round protocol in the GAM and f a function such that on input $s_1, \ldots, s_n \sim \{0,1\}^{\mathsf{poly}(\lambda)}$ there exists P_i returning at round k the element $E_{\mathsf{out}} = f(s_1, \ldots, s_n) \star E_0$.*

Then up to probability $(k+1)\varepsilon_{\mathsf{seq}}$ there exists $S \subseteq \{1, \ldots, n\}$ and a PPT machine $\mathcal{A}$ such that, calling ρ_i the random coins of P_i

1. *$|S| \leq k$*
2. *$s' \leftarrow \mathcal{A}(\mathsf{trs}, \{s_i, \rho_i\}_{i \in S})$ with $s' \star E_0 = E_{\mathsf{out}}$.*

Proof. We begin by applying the *Interactive Sequentiality Lemma* 6.1.4, stating that up to probability $(k+1)\varepsilon_{\mathsf{seq}}$ there exists (r, i, j) such that $E_{\mathsf{out}} = E_{r,i,j}$ and $0 \twoheadrightarrow^{+} (r, i, j)$. Next let $(r_1, i_1, j_1), \ldots (r_t, i_t, j_t) = (r, i, j)$ be a chain for the above relation, i.e.

$$0 \twoheadrightarrow (r_1, i_1, j_1) \twoheadrightarrow \ldots \twoheadrightarrow (r_t, i_t, j_t).$$

We define $S = \{i_1, \ldots, i_t\}$ the set of users involved in the chain[3]. To upper bound the size of S we use Lemma 6.1.3: because the protocol has k round, $|S| \leq |\{r_1, \ldots, r_t\}| \leq k$. Next we provide an explicit description of $\mathcal{A}$ computing s' from $\{s_i, \rho_i\}_{i \in S}$ and trs. Initially $\mathcal{A}$ executes P_i with $i \in S$ for all rounds. In this way it performs all queries $(r_1, i_1, j_1), \ldots, (r_t, i_t, j_t)$ and in particular knows the group elements used in those queries. The sum s' of all these group elements then is such that $s' \star E_0 = E_{\mathsf{out}}$. A formal description is provided in Figure 6.4.

Since the query $(r_\alpha, i_\alpha, j_\alpha)$ for $\alpha \in \{1, \ldots, t\}$ is performed by P_{i_α} at round r_α, then $\mathcal{A}$ also performs this query as by construction $i_\alpha \in S$ and $\mathcal{A}$ runs $P_{i_\alpha}^{\mathcal{O}_{\mathsf{act}}}(s_{i_\alpha}, \mathsf{trs}_{r_\alpha - 1}; \rho_{i_\alpha})$ in line 4. Finally, by Lemma 6.1.2

$$s' \star E_0 \;=\; (a_{r_t, i_t, j_t} + \ldots + a_{r_1, i_1, j_1}) \star E_0 \;=\; E_{r_t, i_t, j_t} \;=\; E_{\mathsf{out}}.$$

[2]These can be defined for $\mathbb{G}$ if $\mathbb{Z}_N$ has an *exceptional set* of size at least n, with N being the order of $\mathbb{G}$.
[3]Note $|S|$ may be smaller than t if there are repetitions among $i_1, \ldots, i_t$.

$\mathcal{A}^{\mathcal{O}_{\mathsf{act}}}(\mathsf{trs}, \{s_i, \rho_i\}_{i \in S})$

1 : Compute $\mathsf{trs}_0, \mathsf{trs}_1 \ldots, \mathsf{trs}_r$ from trs
2 : // Execute all users in S. Note that at round r' users get $\mathsf{trs}_{r'-1}$
3 : **For** all $i' \in S$ and $r \in \{1, \ldots, k\}$:
4 : Run $P_{i'}^{\mathcal{O}_{\mathsf{act}}}(s_{i'}, \mathsf{trs}_{r'-1}; \rho_{i'})$
5 : // Compute s' from the users' queries
6 : Find $(r_1, i_1, j_1), \ldots (r_t, i_t, j_t)$ such that $0 \rightarrowtriangle (r_1, i_1, j_1) \rightarrowtriangle \ldots \rightarrowtriangle (r_t, i_t, j_t)$
7 : Retrieve queried group elements $a_{r_1,i_1,j_1}, \ldots, a_{r_t,i_t,j_t}$
8 : **Return** $s' \leftarrow a_{r_t,i_t,j_t} + \ldots + a_{r_1,i_1,j_1}$

Figure 6.4: Adversary $\mathcal{A}$ computing s' such that $s' \star E_0 = E_{\mathsf{out}}$

This completes the proof. □

6.2 Fair Protocols Lower-Bound

6.2.1 Fair Protocols

As proven in the previous section, round-robin protocols achieve the best round complexity in the GAM. These however do not achieve *fairness* even against weak adversaries. Indeed, since only the last user gets the result, it can simply halt instead of communicating it to others. Remarkably, in order to carry out this attack, an adversary only needs to be able to

- corrupt only 1 user.
- deviate from the protocol only through crashes.

Moreover, it does not even have to be *rushing*, i.e. able to receive for each round honest users' messages before computing and sending its own.

In this section we will study protocols that address this issue, and eventually provide communication and computation lower bounds for them in the GAM. More specifically we focus on protocols among n users to compute a function of all parties' private inputs acting on a given set elements such that:

1. it prevents $n-1$ honest-but-curious users from reconstructing the secret,
2. in honest executions, all parties obtain their output in the last round,
3. it requires exactly n rounds of communication (i.e. it is round optimal according to Theorem 6.1.5).

We immediately observe that these simple restrictions, the second one being necessary for fairness in general, imply fairness against the weak class of attacks described above. The idea is that if an adversary obtains the output before the n-th round, and then crashes the corrupted party, then using Theorem 6.1.5, one could find a subset of $n-1$ or less users who are able to recover the secret group element. Conversely, if it crashes at round n, as

we assumed it not to be rushing, it can only halt after sending its own messages. Hence all parties eventually get their output as well.

Noticeably, the above argument does not imply that fair protocols have to be round-optimal. Even more so, as our lower bounds will only apply to round-optimal protocols, this leaves open the possibility that fair solutions with sub-optimal round complexity but better communication and computational costs exist. Since our techniques do not seem to easily generalize in such case, we leave this as an interesting open question.

Finally, it may appear uninteresting in practice to only study security against such a weak class of attacks. We remark that, as we will prove lower bounds for these protocols, our results applies to stronger models of corruption as well.

6.2.2 Refined Interactive Sequentiality Lemma

In order to provide our second lower bound we will need an improved version of the Interactive Sequentiality Lemma, Section 6.1.3. First let us recall its statement. Given n parties with inputs $s_1, \ldots, s_n$ jointly computing $E_{\mathsf{out}} = f(s_1, \ldots, s_n) \star E_0$, Lemma 6.1.4 states that up to negligible probability, $E_{\mathsf{out}} = E_{r,i,j}$ and $0 \twoheadrightarrow^+ (r, i, j)$. Let $(r_1, i_1, j_1), \ldots, (r_t, i_t, j_t)$ be a chain of queries for $0 \twoheadrightarrow^+ (r, i, j)$, i.e. such that

$$0 \twoheadrightarrow (r_1, i_1, j_1) \twoheadrightarrow (r_2, i_2, j_2) \twoheadrightarrow \ldots \twoheadrightarrow (r_t, i_t, j_t) = (r, i, j).$$

In our improved lemma we will show that this chain can be chosen so that the first query occurring at round r_α is *minimal* among all queries performed in the same round r_α with respect to the relation $\twoheadrightarrow^+$. This means that the set elements used to perform this minimal query was not computed in the same round by P_{i_α}. This property will prove useful when studying communication lower bounds, as it roughly implies that the set element used in minimal queries highly depends on messages previously received. More formally we give the following definition:

Definition 6.2.1. *Given $P_1, \ldots, P_n$ PPT defining a k rounds protocol, and a sequence of queries $(r_1, i_1, j_1), \ldots, (r_t, i_t, j_t)$ such that*

$$0 \twoheadrightarrow (r_1, i_1, j_1) \twoheadrightarrow \ldots \twoheadrightarrow (r_t, i_t, j_t)$$

we call this a refined chain if for all $r \in \{r_1, \ldots, r_t\}$ there exists an index α such that $(r_\alpha, i_\alpha, j_\alpha)$ is minimal among all queries of the form $(r_\alpha, \cdot, \cdot)$ with respect to $\twoheadrightarrow^+$.

Lemma 6.2.1. *Let $P_1, \ldots, P_n$ be a k round protocol in the GAM and f a function such that on inputs $s_1, \ldots, s_n \sim \{0,1\}^{\mathsf{poly}(\lambda)}$, there exists P_i which at round k returns $E_{\mathsf{out}} = f(s_1, \ldots, s_n) \star E_0$. Up to probability $(k+1) \cdot \varepsilon_{\mathsf{seq}}$ then $E_{\mathsf{out}} = E_{r,i,j}$ and there exists a refined chain such that $0 \twoheadrightarrow^+ (r, i, j)$.*

Proof. The proof is similar to Lemma 6.1.4. As in that case, we define Ω as in Figure 6.2 and $\xi(r, i, j)$ and indexing function so that the j-th query performed by P_i at round r corresponds to the $\xi(r, i, j)$-th performed by Ω. Through the Sequentiality Lemma 6.1.1 we then have that up to probability $\varepsilon_{\mathsf{seq}}$

$$E_{\mathsf{out}} = E_{r',i',j'} \qquad 0 \rightarrow^+ \xi(r', i', j').$$

For notational convenience we call $\mathsf{Ref}(r,i,j)$ the event $0 \mathrel{-\!\!\triangleright}^{+} (r,i,j)$ through a refined chain. To conclude we only need to prove that

Claim 6.2.1. $\Pr\left[0 \rightarrow^{+} \xi(r,i,j),\ \neg\mathsf{Ref}(r,i,j)\right] \leq r \cdot \varepsilon_{\mathsf{seq}}$.

This, along with the aforementioned condition, completes the proof. □

Proof of Claim 6.2.1. We proceed by induction on r. If $r = 0$ the statement is trivially true. Assuming it holds for all $r' < r$ we will prove it for (r,i,j). Let j^* be a minimal element in the poset of queries performed by P_i at round r, ordered with $\rightarrow^{+}$.

By definition of $\rightarrow^{+}$ then there exists a chain $j_0, \ldots, j_t$ of queries such that

$$(r,i,j^*) \;=\; (r,i,j_0) \rightarrow (r,i,j_1) \rightarrow \ldots \rightarrow (r,i,j_t) \;=\; (r,i,j)$$

and in particular $(r,i,j^*) \mathrel{-\!\!\triangleright}^{+} (r,i,j)$. If $0 \rightarrow^{+} \xi(r,i,j)$ then, summing the group elements appearing in a chain for this relation and for $(r,i,j^*) \mathrel{-\!\!\triangleright}^{+} (r,i,j)$ we would obtain two group elements α, β such that

$$\alpha \star E_0 \;=\; E_{r,i,j}, \quad \beta \star E_{r,i,j^*} \;=\; E_{r,i,j} \quad \Rightarrow \quad E_{r,i,j^*} \;=\; (-\beta + \alpha) \star E_0$$

Summing the group elements appearing in this chain of queries, as well as Next we define $\mathcal{A}$ which computes E_{r,i,j^*} by executing at round r first P_i and then all other users. A full description of $\mathcal{A}$ appears in Figure 6.5.

$\mathcal{A}^{\mathcal{O}_{\mathsf{act}}}(s_1, \ldots, s_n)$

// Behave as Ω for the first $r-1$ rounds
Set $\mathsf{trs}_0 \leftarrow \perp$
For all $r' \in \{1, \ldots, r-1\}$:
  For all $i' \in \{1, \ldots, n\}$:
    $M_{r,i'} \leftarrow P_{i'}^{\mathcal{O}_{\mathsf{act}}}(s_{i'}, \mathsf{trs}_{r'-1})$
  $\mathsf{trs}_{r'} \leftarrow \mathsf{trs}_{r'-1} \cup \{M_{r',i'}\}_{i'=1}^{n}$
// Execute P_i first at round r
Run $P_i^{\mathcal{O}_{\mathsf{act}}}(s_i, \mathsf{trs}_{r-1})$
For $i' \in \{1, \ldots, n\} \setminus \{i\}$: Run $P_{i'}^{\mathcal{O}_{\mathsf{act}}}(s_{i'}, \mathsf{trs}_{r-1})$
// Find a chain according to Ω's query order
If $0 \not\mathrel{-\!\!\triangleright}^{+} \xi(r,i,j)$: **Return** $\perp$
Use a chain $0 \rightarrow^{+} \xi(r,i,j)$ to find α such that $E_{r,i,j} = \alpha \star E_0$
Use a chain $(r,i,j^*) \mathrel{-\!\!\triangleright}^{+} (r,i,j)$ to find β such that $E_{r,i,j} = \beta \star E_{r,i,j^*}$
Return $(-\beta + \alpha, E_{r,i,j^*})$

Figure 6.5: Program $\mathcal{A}$ computing E_{r,i,j^*}.

As done with Ω, we define an indexing function η so that the j-th query performed by P_i at round r is $\mathcal{A}$'s $\eta(r,i,j)$-th query. Since $\mathcal{A}$ and Ω executes parties in the same order until round r, ξ and η agrees for $r' < r$.

Next, applying the Sequentiality Lemma 6.1.1 we have that

$$\Pr\left[0 \rightarrow^{+} \xi(r,i,j), \quad 0 \not\rightarrow^{+} \eta(r,i,j^*)\right] \;\leq\; \varepsilon_{\mathsf{seq}}$$

because $\mathcal{A}$ computes the correct element if $0 \rightarrow^{+} \xi(r,i,j)$ and $(r,i,j^*) \rightarrowtriangle^{+} (r,i,j)$ which is true by construction.

Next, assuming $0 \rightarrow^{+} \eta(r,i,j^*)$, let (r',i',j') the predecessor in a given chain for this relation. We will show the following implications:

$$\begin{bmatrix} 0 \rightarrow^{+} \xi(r,i,j) \\ \neg\mathsf{Ref}(r,i,j) \\ 0 \rightarrow^{+} \eta(r,i,j^*) \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} 0 \rightarrow^{+} \xi(r,i,j) \\ r' < r \\ \neg\mathsf{Ref}(r',i',j') \end{bmatrix}$$

The first is trivial. The second one follow since if $r = r'$ then $\eta(r,i',j') \rightarrow \eta(r,i,j^*)$ implies $i = i'$ since P_i is the first executed player in round r. In particular this implies $(r,i,j') \rightarrowtriangle (r,i,j^*)$ which contradicts the minimality of j^*. We also prove the third one by contradiction. If $\mathsf{Ref}(r',i',j')$ then we can extend any refined chain for $0 \rightarrowtriangle^{+} (r',i',j')$ with

$$(r',i',j') \rightarrowtriangle (r,i,j^*) \rightarrowtriangle^{+} (r,i,j)$$

where the first relation is true as $r' < r$ and $\eta(r',i',j') \rightarrow \eta(r,i,j^*)$ and the second one follows by construction of j^*. By minimality of j^* the resulting chain would be refined, implying $\mathsf{Ref}(r,i,j)$, which is assumed to be false.

Using this implication we next bound the following probability:

$$\begin{aligned} &\Pr\left[0 \rightarrow^{+} \xi(r,i,j),\ \neg\mathsf{Ref}(r,i,j),\ 0 \rightarrow^{+} \eta(r,i,j^*)\right] \\ \leq\ &\Pr\left[0 \rightarrow^{+} \xi(r,i,j),\ \neg\mathsf{Ref}(r',i',j'),\ r' < r\right] \\ \leq\ &r' \cdot \varepsilon_{\mathsf{seq}} \;\leq\; (r-1)\varepsilon_{\mathsf{seq}}. \end{aligned}$$

Finally, combining the two bounds we conclude that the claim is true.

$$\begin{aligned} &\Pr\left[0 \rightarrow^{+} \xi(r,i,j),\ \neg\mathsf{Ref}(r,i,j)\right] \\ \leq\ &\Pr\left[0 \rightarrow^{+} \xi(r,i,j),\ 0 \not\rightarrow^{+} \eta(r,i,j^*)\right] \\ &\quad +\ \Pr\left[0 \rightarrow^{+} \xi(r,i,j),\ \neg\mathsf{Ref}(r,i,j),\ 0 \rightarrow^{+} \eta(r,i,j^*)\right] \\ \leq\ &\varepsilon_{\mathsf{seq}} + (r-1)\varepsilon_{\mathsf{seq}} \;=\; r\varepsilon_{\mathsf{seq}}. \end{aligned}$$

□

6.2.3 Tall Sub-tree Property

Our technique to study fair protocols will be to associate a tree with special properties to the protocol, and translate bounds for the tree size to communication and computation lower bounds. In this section we therefore introduce the *tall sub-tree* (TS for short) property for tree graphs, and lower bound their size.

Informally a tree is *tall* if all leaves have the same distance from the root, and its height is higher than the number of leaves. A tree then satisfies the TS property if all its (non-trivial)

sub-trees are tall. To be more formal we introduce some notation. Given $T = (V, E)$ a tree, $\mathsf{height}(T)$ is its height (the longest path's length) and $\mathsf{leaves}(T)$ its number of leaves. T_v for $v \in V$ is the sub-tree rooted in v.

Definition 6.2.2. *A tree $T = (V, E)$ is tall if all leaves have the same distance from the root and either $|V| = 1$ or $\mathsf{height}(T) \geq \mathsf{leaves}(T)$. T satisfies the tall sub-tree (TS) property if T_v is tall for all $v \in V$.*

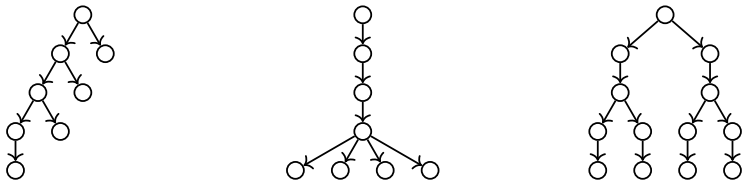

Figure 6.6: Examples of non-TS (left), tall but non-TS (center) and TS (right) trees.

Proposition 6.2.2. *Let $T = (V, E)$ be a TS tree with $\mathsf{height}(T) = m$ and $\mathsf{leaves}(T) = n$. Then*

$$|E| \;\geq\; m + n \log_2 n.$$

We quickly observe that because in any tree $|V| = |E| + 1$, the Proposition above could be restated as $|V| \geq m + 1 + n \log_2 n$. Next we prove a bound for the number of nodes of distance at least two from the root.

Proposition 6.2.3. *Let $T = (E, V)$ a TS tree with $\mathsf{height}(T) = m$ and $\mathsf{leaves}(T) = n$. Furtherfore let $V_{\geq 2}$ the set of nodes with distance at least two from the root. Then*

$$|V_{\geq 2}| \geq m + n \log_2 n - 2.$$

Proofs for these propositions appear in [42].

6.2.4 Lower Bound

We are finally ready to state and prove our second lower bound for fair protocols with optimal round complexity. Regarding our notation, we remind that μ denotes the GAM label size, i.e. the number of bits used to represent set elements, and that trs denotes the tuple of messages exchanged throughout the protocol's execution. In order to give meaningful lower bound on the communication complexity we define for the tuple of messages $M_r^{(i)}$ received by P_i at round r

$$\ell(M_r^{(i)}) \;\coloneqq\; \mathrm{H}(M_r^{(i)} \,|\, \mathsf{trs}_{r-1}) \qquad \ell_{\mathsf{tot}} \;=\; \sum_{r=1}^{k} \sum_{i=1}^{n} \ell_{\mathsf{tot}}(M_r^{(i)}).$$

Roughly $\ell(\,\cdot\,)$ represent the amount of information contained in $M_r^{(i)}$ given all previous messages, and lower bound the information P_i receives conditioned only to messages it previously saw. Hence ℓ_{tot} lower bounds the total information sent throughout the protocol.

Theorem 6.2.4. *Let $P_1, \ldots, P_n$ be an n-round protocol in the GAM and f a function such that on input $s_1, \ldots, s_n \sim \{0,1\}^{\mathsf{poly}(\lambda)}$, every P_i returns at last round the element $E_{\mathsf{out}_i} = f(s_1, \ldots, s_n) \star E_0$. If there exists no set $S \subseteq \{1, \ldots, n\}$ and adversary $\mathcal{A}$ satisfying*

the conditions of Theorem 6.1.5 then up to probability $(n+1)\varepsilon_{\mathsf{seq}}$*, calling* q *the total number of* $\mathcal{O}_{\mathsf{act}}$ *queries*

$$q \geq n(1+\log n), \qquad \ell_{\mathsf{tot}} \geq (n(1+\log n)-2)\cdot\left(\mu - \frac{q}{2^{\mu-1}-q}\right).$$

As for Theorem 6.1.5, this result readily generalizes to protocol reconstructing the action of a t out of n secret shared value, which requires exactly t rounds and at least a subset of t users get the output. In such case the protocol must involve at least $t(1+\log_2 t)$ queries and no less than $\approx t(1+\log_2)\mu$ bits of communication.

Proof. The proof consists of four steps:

1. Observing that any chain for $0 \twoheadrightarrow^+ \mathsf{out}_i$ contains at least a query from each user. In particular we can associate to each chain a permutation π_i assigning to round r the (only) user whose round r queries appears in the chain.

2. Given $\pi_1, \ldots, \pi_n$ permutations we build their *prefix tree* and show it is a TS tree, see Section 6.2.3. In particular it contains at least $n(1+\log n)$ nodes, excluding the root.

3. Using Lemma 6.2.1, we find refined chains for $0 \twoheadrightarrow^+ \mathsf{out}_i$ so that the prefix tree of the associated permutations $\pi_1, \ldots, \pi_n$ satisfies a certain minimality condition. Then we build an injective function f from the nodes (root excluded) to the set of query indexes. By Proposition 6.2.2 this yields $q \geq n(1+\log n)$.

4. Proving each $M_r^{(i)}$ must have enough information about set element which figures as input in queries in $\operatorname{Im} f$ performed by P_i at round $r+1$. The bound on ℓ_{tot} is then a consequence of Proposition 6.2.3.

Regarding the first step, we begin with the following claim, stating that each chain for $0 \twoheadrightarrow^+ \mathsf{out}_i$ must contain queries from all users and cannot skip any round.

Claim 6.2.2. *For all* $(r_1, i_1, j_1), \ldots, (r_t, i_t, j_t)$ *such that*

$$0 \twoheadrightarrow (r_1, i_1, j_1) \twoheadrightarrow \ldots \twoheadrightarrow (r_t, i_t, j_t) = \mathsf{out}_i$$

then $\{i_1, \ldots, i_t\} = \{1, \ldots, n\} = \{r_1, \ldots, r_t\}$.

Using this, for all chains $(r_1, i_1, j_1), \ldots, (r_t, i_t, j_t)$ for $0 \twoheadrightarrow^+ \mathsf{out}_i$ we define a function π_i associating to each round r_α the user i_α who performed at least one query in the chain at that round

$$\pi_i : \{1, \ldots, n\} \to \{1, \ldots, n\} \quad : \quad \pi_i(r_\alpha) = i_\alpha \quad \forall \alpha \in \{1, \ldots, t\}.$$

This is a function because $\twoheadrightarrow$ implies there is at most one user performing queries for each round and, by Claim 6.2.2, π_i is defined for all r. In fact this is a permutation as stated in the next claim, which completes the first step.

Claim 6.2.3. *For all* $(r_1, i_1, j_1), \ldots, (r_t, i_t, j_t)$ *chain for* $0 \twoheadrightarrow^+ \mathsf{out}_i$*,* π_i *is a bijection and* $\pi_i(n) = i$.

Regarding the second step, assuming $0 \not\rightarrow^{+} \mathsf{out}_i$ for all i, we can define $\pi_1, \dots, \pi_n$ for any choice of chains realizing these relations. We then construct their prefix tree. This is done by defining for each $r \in \{0, \dots, n\}$ an equivalence relation where π_i is equivalent to π_j if the two functions agree on the first r evaluations. Note that for $r = 0$ all permutations are equivalent and, due to Claim 6.2.3, for $r = n$ no two distinct permutations are. Then the equivalence classes are

$$[\pi_i]_r \;=\; \{\pi_j \;:\; \pi_j(1) = \pi_i(1), \dots, \pi_i(r) = \pi_j(r)\}\,.$$

For the sake of clarity we notice that $[\pi_i]_0 = \{\pi_1, \dots, \pi_n\}$ and, by Claim 6.2.3, $[\pi_i]_n = \{\pi_i\}$. With this notation their prefix tree $T = (V, E)$ is defined as

$$\begin{aligned} v_{i,r} &\coloneqq ([\pi_i]_r,\, r) \\ V &= \{v_{i,r} \;:\; i \in \{1, \dots, n\},\; r \in \{0, \dots, n\}\} \\ E &= \{(v_{i,r},\, v_{i,r+1}) \;:\; i \in \{1, \dots, n\},\; r \in \{1, \dots, n-1\}\}\,. \end{aligned}$$

i.e. the class $[\pi_i]_{r+1}$ is connected to the class $[\pi_i]_r$ it refines. We conclude the second step with the next claim.

Claim 6.2.4. *If $0 \not\rightarrow^{+} \mathsf{out}_i$ for all $i \in \{1, \dots, n\}$, then for all chains realizing them and associated permutations $\pi_1, \dots, \pi_n$, their prefix tree T is a TS tree with $\mathsf{height}(T) = n$ and $\mathsf{leaves}(T) = n$.*

For the third step we use Lemma 6.2.1. Since each P_i returns $E_{\mathsf{out}_i} = f(s_1, \dots, s_n) \star E_0$, up to probability $(n+1)\varepsilon_{\mathsf{seq}}$, for all i there exists a refined chain for $0 \not\rightarrow^{+} \mathsf{out}_i$, see Definition 6.2.1. Conditioning on this event, we can chose n refined chains with associated permutations $\pi_1, \dots, \pi_n$ so that, calling $V_t = \{v_{i,r} \;:\; i \in \{1, \dots, t\},\; r \in \{0, \dots, n\}\}$, the tuple

$$(|V_1|, |V_2|, \dots, |V_n|)$$

is minimal w.r.t. the lexicographic order. This means that for any other choice of refined chains, the associated permutations $\pi'_1, \dots, \pi'_n$ defines a prefix tree $T' = (V', E')$ so that either $|V_t| = |V'_t|$ for all t or there exists a t such that

$$|V_1| = |V'_1| \;\wedge\; \dots \;\wedge\; |V_t| = |V'_t| \;\wedge\; |V_{t+1}| < |V'_{t+1}|.$$

Using this we will construct an injective function f from the tree nodes (excluding the root) to the set of query indices. Each node $v = ([\pi_i]_r, r)$ for some π_i will be mapped to a query that is:

- in the chain for $0 \not\rightarrow^{+} \mathsf{out}_i$ used to construct π,
- minimal among round r queries, with respect to $\not\rightarrow^{+}$.

Claim 6.2.5. *There exists refined chains for $0 \not\rightarrow^{+} \mathsf{out}_i$ so that, calling $T = (V, E)$ the resulting prefix tree and V^* the set of nodes excluding the root, there exists $f : V^* \to \mathbb{N}^3$ such that*

1. *For each $v \in V$ there exist r, i, j so that $v = v_{i,r}$ and $f(v) = (r, \pi_i(r), j)$ is a query in the chain used to construct π_i. Moreover $f(v)$ is minimal w.r.t. $\not\rightarrow^{+}$ among all queries of the form $(r, \cdot, \cdot)$.*

2. *f is injective.*

3. *Calling $a_{r,i,j}, D_{r,i,j}$ the input of $\mathcal{O}_{\mathsf{act}}$ in the (r,i,j)-th query, then*

$$D_{f(u)} = D_{f(v)} \quad \Rightarrow \quad \exists w \,:\, (w,u),(w,v) \in E.$$

The first two properties combined implies that the set of queries contains at least $|V^*| = |E| \geq n(1+\log_2 n)$ elements, where the last bound follows from Claim 6.2.4 and Proposition 6.2.2.

Finally we go through the last step. In order to bound communication we first bound the number of minimal queries using different set elements as input performed by P_i at round r. Next we will prove $M_{r-1}^{(i)}$ has to contain enough information about these elements. Toward this goal we define $U_{r,i}$ as the set of nodes to which f associate a query performed by P_i at round r, and $\Delta_{r,i}$ the collection of set elements used in those queries.

$$U_{r,i} \coloneqq \{v \in V \,:\, \exists j \,:\, f(v) = (r,i,j)\} \qquad \Delta_{r,i} \coloneqq \{D_{f(v)} \,:\, v \in U_{r,i}\}$$

First we give a bound on the size of $\Delta_{r,i}$.

Claim 6.2.6. $\sum_{r=1}^{n-1}\sum_{i=1}^{n} |\Delta_{r+1,i}| \geq n(1+\log_2 n) - 2$.

Then, we relate the size of $\Delta_{r+1,i}$ with the entropy in $M_r^{(i)}$

Claim 6.2.7. *With the previous notation,*

$$\mathrm{H}\left(M_r^{(i)} \,\middle|\, \mathsf{trs}_{r-1}\right) \;\geq\; |\Delta_{r+1,i}| \cdot \left(\mu - \frac{q}{2^{\mu-1}-q}\right).$$

This eventually concludes the proof of Theorem 6.2.4 because

$$\begin{aligned}
\ell_{\mathsf{tot}} \;&=\; \textstyle\sum_{r=1}^{n-1}\sum_{i=1}^{n} \ell(M_r^{(i)}) \\
&\geq\; \textstyle\sum_{r=1}^{n-1}\sum_{i=1}^{n} |\Delta_{r+1,i}| \cdot \left(\mu - \frac{q}{2^{\mu-1}-q}\right) \\
&\geq\; (n(1+n\log_2 n) - 2) \cdot \left(\mu - \frac{q}{2^{\mu-1}-q}\right).
\end{aligned}$$

□

Proof of Claim 6.2.2. Assume by contradiction that there exists a chain for $0 \twoheadrightarrow^{+} \mathsf{out}_i$ such that $S \coloneqq \{i_1,\ldots,i_t\} \subsetneq \{1,\ldots,n\}$. Then, as shown in the proof of Theorem 6.1.5, the adversary $\mathcal{A}$ described in Figure 6.4 on input trs and $(s_i,\rho_i)_{i\in S}$, with ρ_i being the random coins of P_i, recovers s' such that $s' \star E_0 = E_{\mathsf{out}_i}$. This contradicts the assumption that such a pair $(S,\mathcal{A})$ does not exist.

Next, using Lemma 6.1.3 and the fact that the protocol has n rounds,

$$n \;=\; |\{i_1,\ldots,i_t\}| \;\leq\; |\{r_1,\ldots,r_t\}| \;=\; n \quad \Rightarrow \quad \{r_1,\ldots,r_t\} = \{1,\ldots,n\}.$$

□

Proof of Claim 6.2.3. We begin showing that π_i is a total function from $\{1,\dots,n\}$. Let $\alpha < \beta$ be two indexes such that $r_\alpha = r_\beta$. Because $(r_\alpha, i_\alpha, j_\alpha) \rightarrowtriangle^+ (r_\beta, i_\beta, j_\beta)$, by the definition of $\rightarrowtriangle^+$, $r_\alpha = r_\beta$ implies $i_\alpha = i_\beta$. Hence π_i associate the same value to r_α and r_β. Moreover, by Claim 6.2.2, for each $r \in \{1,\dots,n\}$ there exists an α such that $r = r_\alpha$. As a consequence π_i is well defined function with domain $\{1,\dots,n\}$.

Next we observe that $\operatorname{Im}\pi_i = \{i_1,\dots,i_t\} = \{1,\dots,n\}$ where we used again Claim 6.2.2, implying that $\pi_i : \{1,\dots,n\} \to \{1,\dots,n\}$ is a surjective function between finite sets of the same size, and therefore also a bijection.

Finally, since the query out_i is performed by P_i, it has the form (r_t, i, j_t). If $r_t < n$ then we would find a chain with $\{r_1,\dots,r_t\}$ of size strictly smaller than n, contradicting Claim 6.2.2. Therefore $r_t = n$ and $\pi_i(n) = \pi_i(r_t) = i$. □

Proof of Claim 6.2.4. T is a tree because each node $v_{i,r}$ is connected to the root by the path

$$\Big((v_{i,0},\, v_{i,1}) \,, \dots, (v_{i,r-1},\, v_{i,r}) \Big),$$

with $v_{i,0} = (\{\pi_1,\dots,\pi_n\}, 0)$ being equal for all i, and each node has in-degree 1 because

$$\begin{aligned}
&(v_{j,r-1},\, v_{i,r}) \,,\; (v_{k,r-1},\, v_{i,r}) \in E \quad : \quad \begin{cases} v_{i,r} = ([\pi_i]_r,\, r) \\ v_{j,r-1} = ([\pi_j]_{r-1},\, r-1) \\ v_{k,r-1} = ([\pi_k]_{r-1},\, r-1) \end{cases} \\
\Rightarrow\quad & \pi_j(x) = \pi_i(x) = \pi_k(x) \quad \forall x \in \{1,\dots,r-1\} \\
\Rightarrow\quad & [\pi_j]_{r-1} = [\pi_k]_{r-1} \quad \Rightarrow \quad v_{j,r-1} = v_{k,r-1}.
\end{aligned}$$

By construction the leaves of T are $v_{i,n}$ with i ranging from 1 to n, and these are all distinct. Indeed for $i \neq j$ we have $\pi_i(n) = i \neq j = \pi_j(n)$ implying $v_{i,n} \neq v_{j,n}$. Thus $\mathsf{leaves}(T) = n$. Moreover each leaf has distance n from the root, so $\mathsf{height}(T) = n$.

Next we show that the sub-tree of $v_{i,r} = ([\pi_i]_r, r)$ is a tall tree. By previous observations its height is $n - r$. If $r = n$ the node is a leaf and it is trivially tall. Conversely let $v_{j_1},\dots,v_{j_n}$ be all the leaves of this sub-tree, so that $v_{j_\alpha} = ([\pi_{j_\alpha}]_n, n)$. Then for all α we have that $v_{i,r} = v_{j_\alpha,r}$ because from both nodes there exists a path to $v_{j_\alpha,n}$, and there exists no path connecting the two nodes. As a consequence

$$[\pi_i]_r \;=\; [\pi_{j_1}]_r \;=\; \dots \;=\; [\pi_{j_m}]_r.$$

Hence $\pi_{j_1},\dots,\pi_{j_m}$ have the same values when evaluated on the indexes from 1 to r. Moreover, since these are all permutations, their value on n (as we assumed $n > r$) must differ from their value on previous points. Thus

$$\begin{aligned}
& \{\pi_{j_1}(n),\dots,\pi_{j_m}(n)\} \,\cap\, \{\pi_i(1),\dots,\pi_i(r)\} \;=\; \varnothing \\
\Rightarrow\quad & \{j_1,\dots,j_m\} \,\cap\, \{\pi_i(1),\dots,\pi_i(r)\} \;=\; \varnothing
\end{aligned}$$

where the implication uses Claim 6.2.3. Since $j_1,\dots,j_m$ are all distinct by construction, $\pi_i(1),\dots,\pi_i(r)$ are all distinct as π_i is a bijection, and all these indexes lies in the range $\{1,\dots,n\}$ we conclude

$$|\{j_1,\dots,j_m\}| \;+\; |\{\pi_i(1),\dots,\pi_i(r)\}| \;\leq\; n \quad \Rightarrow \quad m + r \leq n \quad \Rightarrow \quad m \leq n - r.$$

The sub-tree of $v_{i,r}$ is therefore tall, concluding the claim's proof. □

Proof of Claim 6.2.5. We recall $V_t = \{v_{i,r} : i \in \{1,\dots,t\},\ r \in \{0,\dots,n\}\}$ and further define $V_t^* = V_t \setminus \{v_{1,0}\}$, i.e. the set of vertices in V_t without the root. To prove the claim we proceed by induction on t showing the existence of a function $f : V_t^* \to \mathbb{N}^3$ satisfying the required properties.

If $t = 1$, we set $f(v_{1,r}) = (r, \pi_1(r), j)$ to be the query in the refined chain, see Definition 6.2.1, minimal w.r.t. $\twoheadrightarrow^+$ among queries performed at round r. Then f is injective over V_1^* and satisfies the first condition by construction. Regarding the third property if $D_{f(v_{1,r})} = D_{f(v_{1,r'})}$ with $r < r'$ then we could create a shorter chain skipping round r, thus violating Claim 6.2.2. Hence $r = r'$, and in particular $v_{1,r}, v_{1,r'}$ are the same, meaning that $v \mapsto D_{f(v)}$ is injective over V_1^* and thus satisfies the third property.

Assuming the statement to be true for $t-1$, i.e. that we have $f : V_{t-1}^* \to \mathbb{N}^3$ satisfying the three properties, we show f can be extended to V_t^*. To so, let j_r be such that $(r, \pi_t(r), j_r)$ is the minimal query at round r of the chain for $0 \twoheadrightarrow^+ \mathsf{out}_t$. We then define for all $v_{i,r} \in V_t^* \setminus V_{t-1}^*$ the function to be $f(v_{i,r}) = (r, \pi_t(r), j_r)$. The first property is thus satisfied by construction.

Next we show f is injective. If $f(v_{i,r}) = f(v_{i',r'})$ then $r = r'$ from the first property[4]. By inductive hypothesis f is injective over V_{t-1}^*. Without loss of generality we can then assume $v_{i',r'} \in V_t^* \setminus V_{t-1}^*$, in which case $t = i'$. In order to prove $v_{i,r} = v_{t,r}$ we proceed by contradiction assuming the two nodes to be different.

This implies that $v_{i,r} \in V_{t-1}^*$. With loss of generality we can further assume because of the first property that $f(v_{i,r}) = (r, \pi_i(r), j^*)$ is a minimal query at round r for $0 \twoheadrightarrow^+ \mathsf{out}_i$. This implies that there exist refined chains for the relations

$$0 \twoheadrightarrow^+ (r, \pi_t(r), j_r) \twoheadrightarrow^+ \mathsf{out}_t \qquad 0 \twoheadrightarrow^+ (r, \pi_i(r), j^*) \twoheadrightarrow^+ \mathsf{out}_i.$$

Since $(r, \pi_t(r), j_r) = (r, \pi_i(r), j^*)$ we can combine the first half of the second chain with the second half of the first one to obtain a new refined chain for $0 \twoheadrightarrow^+ \mathsf{out}_t$. Let $\widehat{\pi}_t$ be the associated permutations. By construction $\widehat{\pi}_t(x) = \pi_i(x)$ for all $x \in \{1,\dots,t\}$, and in particular $[\widehat{\pi}_t]_r = [\pi_i]_r$. Hence, calling $\widehat{T} = (\widehat{V}, \widehat{E})$ the prefix tree for $\pi_1, \dots, \widehat{\pi}_t, \dots, \pi_n$, we will show it violates our minimality condition on T. Indeed

- $|V_{t-1}^*| = |\widehat{V}_{t-1}^*|$, because the first $t-1$ permutations used to build $T, \widehat{T}$ are the same.
- $|V_t^*| \geq |V_{t-1}^*| + (n - r + 1)$, because $v_{t,r} \in V_{t-1}^* \setminus V_t^*$ and in particular, $v_{t,r'} \notin V_{t-1}^*$ for $r' > r$, or else $v_{t,r'} = v_{i,r'}$ for some $i < t$ which implies $v_{t,r} = v_{i,r} \in V_{t-1}^*$. Hence

$$\{v_{i,r}, \dots, v_{i,n}\} \subseteq V_t^* \setminus V_{t-1}^* \quad \Rightarrow \quad |V_t^* \setminus V_{t-1}^*| \geq (n - r + 1)$$

- $|\widehat{V}_t^*| \leq |\widehat{V}_{t-1}^*| + (n - r)$, because $[\widehat{\pi}_t]_r = [\pi_i]_r$ implies $[\widehat{\pi}_t]_{r'} = [\pi_i]_{r'}$ for all $r' \leq r$. Hence, again for all $r' \leq r$, $\widehat{v}_{t,r'} = \widehat{v}_{i,r'} \in \widehat{V}_{t-1}^*$ and in particular

$$\widehat{V}_t^* \setminus \widehat{V}_{t-1}^* \subseteq \{\widehat{v}_{t,r+1}, \dots, \widehat{v}_{t,n}\} \quad \Rightarrow \quad |\widehat{V}_t^* \setminus \widehat{V}_{t-1}^*| \leq n - t.$$

[4] r and r' are the first component of respectively the LHS and the RHS.

Combining the three relations we conclude that our minimality assumption on T is violated because

$$|\widehat{V}_t^*| \leq |\widehat{V}_{t-1}^*| + (n-r) < |V_{t-1}^*| + (n-r+1) \leq |V_t^*|.$$

This means that the assumption $v_{i,r} \neq v_{t,r}$ leads to a contradiction, implying that $v_{i,r} = v_{t,r}$ and that f is injective.

To conclude we need to prove the third property holds for f. Let $D_{f(v_{i,r})} = D_{f(v_{i',r'})}$. We study two cases:

1. $r \neq r'$. Then without loss of generality $r < r'$ and by the first property $v_{i,r}$, $v_{i',r'}$ are such that $f(v_{i,r}) = (r, \pi_i(r), j)$ and $f(v_{i',r'}) = (r', \pi_{i'}(r'), j')$ are minimal queries in their respective rounds with respect to $\twoheadrightarrow^+$. Among the queries appearing in the chain for $0 \twoheadrightarrow^+ \mathsf{out}_i$ let (r'', i'', j'') be the predecessor of $(r, \pi_i(r), j)$, i.e. such that

   $$0 \twoheadrightarrow^+ (r'', i'', j'') \twoheadrightarrow (r, \pi_i(r), j) \twoheadrightarrow^+ \mathsf{out}_i.$$

   Note that since $(r, \pi_i(r), j)$ is minimal among the queries at round r, we must have $r'' < r$. Then if we call $E_{r'',i'',j''}$ the output of $\mathcal{O}_{\mathsf{act}}$ for query (r'', i'', j''), by the definition of $\twoheadrightarrow$ we have $E_{r'',i'',j''} = D_{(r,\pi_r(i),j)}$. Therefore, again by the definition of $\twoheadrightarrow^+$

   $$E_{r'',i'',j''} = D_{(r',\pi_{r'}(i'),j')}, \wedge r'' < r' \quad \Rightarrow$$
   $$\Rightarrow \quad 0 \twoheadrightarrow^+ (r'', i'', j'') \twoheadrightarrow (r', i', j') \twoheadrightarrow^+ \mathsf{out}_{i'}.$$

   Since $r'' < r < r'$ the resulting chain would not include any query from round r, contradicting Claim 6.2.2. Therefore $r \neq r'$ is impossible.

2. $r = r'$. By the inductive hypothesis if both vertices lie in V_{t-1}^* the property holds, so without loss of generality assume $v_{i',r} \in V_t^* \setminus V_{t-1}^*$ and $f(v_{i,r}) = (r, \pi_i(r), j_r)$, i.e. that the image of $v_{i,r}$ is a query on the chain for $0 \twoheadrightarrow^+ \mathsf{out}_i$. We will denote p_i the predecessor of $f(v_{i,r})$ on the refined chains for $0 \twoheadrightarrow^+ \mathsf{out}_i$. This means we have

   $$0 \twoheadrightarrow^+ p_i \to f(v_{i,r}) \twoheadrightarrow^+ \mathsf{out}_i$$

   Then by how $\twoheadrightarrow$ was defined $E_{p_i} = D_{f(v_{i,r})} = D_{f(v_{t,r})}$ and by minimality of $f(v_{i,r})$ among the queries occurring at round r, p_i occurs at a round strictly smaller than r. Thus $p_i \twoheadrightarrow f(v_{t,r})$ and in particular we can find a chain for

   $$0 \twoheadrightarrow^+ p_i \to f(v_{t,r}) \twoheadrightarrow^+ \mathsf{out}_t$$

   that is equal to the chain for $0 \twoheadrightarrow^+ \mathsf{out}_i$ until query p_i. Calling $\widehat{\pi}_t$ the associated permutation we would then have $[\widehat{\pi}_t]_{r-1} = [\pi_i]_{r-1}$ since the chains are equal until round $r-1$ (we use Claim 6.2.2 to observe p_i occurs at round $r-1$).

   Finally assume that for the current chain chosen for $0 \twoheadrightarrow^+ \mathsf{out}_t$ the nodes $v_{i,r}$ and $v_{t,r}$ are no siblings (otherwise the claim is proven), i.e. $v_{i,r-1} \neq v_{t,r-1}$, we distinguish two cases:

   - $v_{t,r-1} \notin V_{t-1}^*$. Then as done previously we can use $\widehat{\pi}_t$ to build a prefix tree $\widehat{T} = (\widehat{V}, \widehat{E})$ with $|\widehat{V}_t| < |V_t|$, which contradicts our minimality assumption.

- $v_{t,r-1} \in V^*_{t-1}$. Again using the new path we can build a prefix tree such that for all $t' < t$

$$|V^*_{t'}| = |\widehat{V}^*_{t'}| \qquad |V^*_t| = |V^*_{t-1}| + (n-r+1) \qquad |\widehat{V}^*_t| \leq |\widehat{V}^*_{t-1}| + (n-r+1).$$

with the first equality holding as we are only replacing π_t with $\widehat{\pi}_t$ and using the same $\pi_1, \ldots, \pi_{t-1}$ in both prefix trees, the second one because $v_{t,r-1} \in V^*_t \setminus V^*_{t-1}$ while $v_{t,r-1} \in V^*_{t-1}$, and the third because $\widehat{v}_{t,r-1} = \widehat{v}_{i,r-1} \in \widehat{V}^*_{t-1}$. We thus conclude $|\widehat{V}_t| \leq |V_t|$ while preserving the size of smaller sub-trees.

Replacing the chain for $0 \twoheadrightarrow^+ \mathsf{out}_t$, we finally have that for the new tree $\widehat{v}_{i,r-1} = \widehat{v}_{i,r-1}$. Notice that this change occurs only once since there can only be one node $v_{i,r} \in V_t \setminus V_{t-1}$ with $v_{i,r-1} \in V_{t-1}$. Furthermore after the change we still have $\widehat{v}_{i,r} \in \widehat{V}_t \setminus \widehat{V}_{t-1}$ with $\widehat{v}_{i,r-1} \in \widehat{V}_{t-1}$ because

$$\begin{aligned} \widehat{v}_{t,r} \in \widehat{V}^*_{t-1} \quad &\Rightarrow \quad |\widehat{V}^*_t| \leq |\widehat{V}^*_{t-1}| + (n-r) < |V^*_{t-1}| + (n-r+1) = |V^*_t| \\ &\Rightarrow \quad |\widehat{V}_t| < |V_t| \end{aligned}$$

contradicting our minimality assumption. Therefore, all remaining nodes in $\widehat{V}_t \setminus \widehat{V}_{t-1}$ falls into the previous case.

This concludes the proof of the Claim. □

Proof of Claim 6.2.6. Calling $V_{\geq 2}$ as in Section 6.2.3 the set of nodes of distance at least 2 from the root we observe that

$$V_{\geq 2} \;=\; \bigcup_{r=2}^{n} \bigcup_{i=1}^{n} U_{r,i}$$

Indeed given $v \in V_{\geq 2} \subseteq V^*$, by Claim 6.2.5 there exists r, i, j such that $v = v_{i,r}$ and $f(v) = (r, i, j)$, implying $r \geq 2$ and $v \in U_{r,i}$. Next we show $|U_{r,i}| = |\Delta_{r,i}|$. To do so it suffices to show that the map $v \mapsto D_{f(v)}$ is injective over $U_{r,i}$. Let $u, v \in U_{r,i}$ such that $D_{f(u)} = D_{f(v)}$. By Claim 6.2.5 they must have the same parent. In particular if $u = ([\pi_\ell]_r, r)$ and $v = ([\pi_{\ell'}]_r, r)$ then having the same parent implies

$$[\pi_\ell]_{r-1} = [\pi_{\ell'}]_{r-1}, \quad \pi_\ell(r) = i = \pi_{\ell'}(r) \quad \Rightarrow \quad [\pi_\ell]_r = [\pi_{\ell'}]_r \quad \Rightarrow \quad u = v$$

where the second and third equality follows since $u, v \in U_{r,i}$. Finally, using Proposition 6.2.3 and Claim 6.2.4 stating that T is a TS tree we conclude

$$\sum_{r=1}^{n-1} \sum_{i=1}^{n} |\Delta_{r+1,i}| \;=\; \sum_{r=2}^{n} \sum_{i=1}^{n} |\Delta_{r,i}| \;\geq\; |V_{\geq 2}| \;\geq\; (n-2) + n \log_2 n. \qquad \square$$

Proof of Claim 6.2.7. In the following we denote $\mathsf{input} = (s_1, \ldots, s_n, \rho_1, \ldots, \rho_n, E_0)$ where s_i, ρ_i are the private input and random coins of P_i. Furthermore, we will denote

$$\Gamma_{r+1,i,j} \;=\; \{E_{r+1,i,j'} \,:\, j' < j\} \,\cup\, \{D_{r+1,i,j'} \,:\, j' < j\}$$

We furthermore index $\Delta_{r+1,i} = \{D_{r+1,i,j_1}, \ldots, D_{r+1,i,j_m}\}$. Then

$$\begin{aligned}
\mathrm{H}\left(M_r^{(i)} \,\middle|\, \mathsf{trs}_{r-1}\right) &\geq \mathrm{H}\left(M_r^{(i)} \,\middle|\, \mathsf{trs}_{r-1}, \mathsf{input}\right) \\
&\geq \sum_{\alpha=1}^{m} \mathrm{I}\left(M_r^{(i)}; D_{r+1,i,j_\alpha} \,\middle|\, \mathsf{trs}_{r-1}, \mathsf{input}, \{D_{r+1,i,j_\beta}\}_{\beta=1}^{\alpha-1}\right) \\
&\geq \sum_{\alpha=1}^{m} \mathrm{I}\left(M_r^{(i)}; D_{r+1,i,j_\alpha} \,\middle|\, \mathsf{trs}_{r-1}, \mathsf{input}, \Gamma_{r+1,i,j_\alpha}\right).
\end{aligned}$$

We will then lower bound each of these terms. The key observation is that, given $M_j^{(i)}$, trs_{r-1}, input and $\Gamma_{r+1,i,j}$, the execution of P_i becomes deterministic until the next query to $\mathcal{O}_{\mathsf{act}}$ is performed, meaning that $D_{r+1,i,j}$ is univocally determined. Therefore, if $\Delta_{r,i} = \{D_{r+1,i,j_1}, \ldots, D_{r+1,i,j_m}\}$, we have that

$$\mathrm{H}\left(D_{r+1,i,j_\alpha} \,\middle|\, M_r^{(i)}, \mathsf{trs}_{r-1}, \mathsf{input}, \Gamma_{r+1,i,j_\alpha}\right) \;=\; 0.$$

Conversely we observe that before round r, D_{r+1,i,j_α} was not returned as an output by $\mathcal{O}_{\mathsf{act}}$, or else we could build a chain for D_{r+1,i,j_α} skipping round r, which violates Claim 6.2.2. Moreover by the minimality of D_{r+1,i,j_α} (see Claim 6.2.5), this set elements was not computed previously on the same round by P_i, meaning that is does not belong in Γ_{r+1,i,j_α}. Moreover D_{r+1,i,j_α} is independent from the random coins and inputs of parties (which are sampled before any query is ever made to $\mathcal{O}_{\mathsf{act}}$). Hence we conclude that D_{r+1,i,j_α} conditioned to $\mathsf{trs}_{r-1}, \mathsf{input}, \Gamma_{r+1,i,j_\alpha}$ is uniform in the set of not-yet queried labels, which has size $2^\mu - 2q$. Thus

$$\begin{aligned}
\mathrm{H}\left(D_{r+1,i,j_\alpha} \,\middle|\, \mathsf{trs}_{r-1}, \mathsf{input}, \Gamma_{r+1,i,j_\alpha}\right) &\geq \log_2(2^\mu - 2q) \\
&\geq \mu - \frac{2q}{2^\mu - 2q} \;=\; \mu - \frac{q}{2^{\mu-1} - q}.
\end{aligned}$$

Where second inequality follows since $\log(x)$ is concave and for all $x > y > 0$

$$\frac{1}{x} \;\leq\; \frac{\log(x) - \log(y)}{x - y} \;\leq\; \frac{1}{y}$$

replacing $x = 2^\mu$ and $y = 2^\mu - 2q$. As the mutual information is the difference of the above quantities, we have that $\mu - \frac{q}{2^{\mu-1}-q}$ lower bounds each term in the summation above. We can therefore conclude

$$\mathrm{H}\left(M_r^{(i)} \,\middle|\, \mathsf{trs}_{r-1}\right) \;\geq\; m \cdot \frac{q}{2^{\mu-1} - q} \;\geq\; |\Delta_{r+1,i}| \cdot \frac{q}{2^{\mu-1} - q}. \qquad \square$$

Chapter 7

Black-Box Anamorphic Encryption

Chapter Overview

In this chapter we will study the recently introduced problem of realizing anamorphic encryption schemes. More specifically, we ask whether given any PKE scheme, possibly chosen by an adversarial regulation authority we call the *dictator*, is it possible to build an anamorphic encryption scheme for it. The two main results we prove are Theorem 7.4.1 and Theorem 7.4.2.

The first one states addresses any realization of anamorphic encryption using the underlying PKE as a black-box. For such construction the anamorphic message space has to be polynomially bounded in size. This in particular implies that generic constructions such as the one presented in [96] and [10] are optimal. The second result instead addressed the notion of *fully asymmetric* AE. In this context we show that realizing this notion granted only black-box access to the PKE is impossible.

Both results follows combining information-theoretical arguments with the *ciphertext selection lemma* presented in Section 7.2.4. This informally states that any procedure relative to an ideal PKE scheme can only produce valid ciphertexts through encryption queries, for certain choices of the ideal PKE's parameters.

7.1 Supplementary Definitions

7.1.1 Anamorphic Encryption

The notion of (receiver) anamorphic encryption was first introduced in [96] to model private communication in the presence of a dictator who controls the PKE scheme in use and knows each user's secret key. In this thesis, we use a more general definition proposed in [35] which contains [96] as a special case. To achieve the above seemingly impossible goal, the receiver is allowed to generate its own public and secret key $\mathsf{apk}, \mathsf{ask}$ in *anamorphic mode*, exchange secretly with the sender a *double key* dk, and locally storing a *trapdoor key* tk to decrypt anamorphic messages from the sender.

Definition 7.1.1 (Anamorphic Triplet)**.** *Formally, an anamorphic triplet* $\Sigma = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ *is a triplet of efficient algorithms such that*

- $\mathsf{AT.Gen}(\lambda) \xrightarrow{\$} (\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk})$ *with* $\mathsf{apk}, \mathsf{ask}$ *being the anamorphic public and secret keys while* dk, tk *are the double and (a possibly empty) trapdoor key.*
- $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m}) \xrightarrow{\$} c$*, with* $m \in M$ *and* $\widehat{m} \in \widehat{M}$ *being respectively the standard and anamorphic messages encrypted in* c*.*
- $\mathsf{AT.Dec}(\mathsf{ask}, \mathsf{tk}, c) \to \widehat{m}/\perp$*, with* $\widehat{m}$ *being the anamorphic message encrypted in* c*.*

In the definition above we do not explicitly provide $\mathsf{apk}, \mathsf{dk}$ as part of $\mathsf{AT.Dec}$ input, as we implicitly assume them to be contained in ask and tk respectively.

Definition 7.1.2 (Anamorphic Encryption)**.** *A PKE* $\Pi = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ *is an Anamorphic Encryption scheme if it is* IND-CPA *secure and there exists an anamorphic triplet* $\Sigma = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ *such that any* PPT *adversary* $\mathcal{A}$ *has negligible advantage, defined as*

$$\mathsf{Adv}^{\mathsf{Anam}}_{\mathcal{A},\Pi,\Sigma}(\lambda) \;:=\; \left|\Pr\left[\mathsf{RealG}_{\Pi}(\lambda, \mathcal{A}) = 1\right] - \Pr\left[\mathsf{AnamorphicG}_{\Sigma}(\lambda, \mathcal{A}) = 1\right]\right|$$

where RealG_{Π} *and* $\mathsf{AnamorphicG}_{\Sigma}$ *are described in Figure 7.1.*

$\mathsf{RealG}_{\Pi}(\lambda, \mathcal{A})$

1 : $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{E.Gen}(\lambda)$
2 : **return** $\mathcal{A}^{\mathcal{O}_{\mathsf{real}}}(\mathsf{pk}, \mathsf{sk})$

$\mathsf{AnamorphicG}_{\Sigma}(\lambda, \mathcal{A})$

1 : $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \xleftarrow{\$} \mathsf{AT.Gen}(\lambda)$
2 : **return** $\mathcal{A}^{\mathcal{O}_{\mathsf{anam}}}(\mathsf{apk}, \mathsf{ask})$

$\mathcal{O}_{\mathsf{real}}(m, \widehat{m})$

1 : Sample a random r
2 : **return** $\mathsf{E.Enc}(\mathsf{pk}, m; r)$

$\mathcal{O}_{\mathsf{anam}}(m, \widehat{m})$

1 : Sample a random r
2 : **return** $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m}; r)$

Figure 7.1: Anamorphic Encryption security game.

Finally, regarding correctness, a definition covering also stateful AE is presented in [10]. For the sake of generality however we will only use a weaker notion, holding only for uniformly sampled messages and correct keys. Formally, given $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \xleftarrow{\$} \mathsf{AT.Gen}(\lambda)$ and $m, \widehat{m}$ uniformly sampled messages, then

$$\Pr\left[\widetilde{m} \neq \widehat{m} \;\middle|\; \widetilde{m} \leftarrow \mathsf{AT.Dec}(\mathsf{ask}, \mathsf{tk}, c),\; c \xleftarrow{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})\right] \;\leq\; \mathsf{negl}(\lambda).$$

7.1.2 Asymmetric Anamorphic Encryption

The notion of *Asymmetric Anamorphic Encryption* [35], intuitively, requires that the Anamorphic Triplet Σ realizes an asymmetric scheme for covert messages. The notion is formalized through the game in Figure 7.2, where $\mathcal{D}$ is a PPT adversary, $b \in \{0, 1\}$ and $\Sigma = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ is an Anamorphic Triplet. The advantage of a given distinguisher $\mathcal{D}$ is defined as

$$\mathsf{Adv}^{\mathsf{asy\text{-}anam}}_{\mathcal{D},\Sigma}(\lambda) := \left|\Pr\left[\mathsf{AsyAnam\text{-}IND\text{-}CPA}^{0}_{\Sigma}(\lambda, \mathcal{D}) = 1\right] - \Pr\left[\mathsf{AsyAnam\text{-}IND\text{-}CPA}^{1}_{\Sigma}(\lambda, \mathcal{D}) = 1\right]\right|.$$

$\mathsf{AsyAnam\text{-}IND\text{-}CPA}^{b}_{\Sigma}(\lambda, \mathcal{D})$

1 : $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \leftarrow^{\$} \mathsf{AT.Gen}(\lambda)$
2 : $(m, \widehat{m}_0, \widehat{m}_1) \leftarrow^{\$} \mathcal{D}(\mathsf{apk}, \mathsf{ask}, \mathsf{dk})$
3 : $c \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m}_b)$
4 : **return** $\mathcal{D}(c)$

Figure 7.2: Asymmetric Anamorphic Encryption security game.

Definition 7.1.3 (Asymmetric Anamorphic Encryption)**.** *An Anamorphic Encryption scheme* Π *equppied with an anamorphic triplet* Σ *is an Asymmetric Anamorphic Encryption scheme if for every* PPT *distinguisher* $\mathcal{D}$*,*

$$\mathsf{Adv}^{\mathsf{asy\text{-}anam}}_{\mathcal{D},\Sigma}(\lambda) \leq \mathsf{negl}(\lambda).$$

In this thesis we introduce a weaker notion, called *Weak Asymmetric Anamorphic Encryption*. We weaken the previous definition requiring that the adversary in the security game has no access to ask. More precisely, let $\mathcal{D}$ be a PPT adversary, $b \in \{0, 1\}$ and $\Sigma = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ be an Anamorphic Triplet. The Weak Asymmetric AE security game is then detailed in Figure 7.3. The advantage of a distinguisher $\mathcal{D}$ for such game is defined as

$$\begin{aligned}\mathsf{Adv}^{\mathsf{weak\text{-}asy\text{-}anam}}_{\mathcal{D},\Sigma}(\lambda) := \Big|&\Pr\left[\mathsf{Weak\text{-}AsyAnam\text{-}IND\text{-}CPA}^{0}_{\Sigma}(\lambda, \mathcal{D}) = 1\right] \\ &- \Pr\left[\mathsf{Weak\text{-}AsyAnam\text{-}IND\text{-}CPA}^{1}_{\Sigma}(\lambda, \mathcal{D}) = 1\right]\Big|.\end{aligned}$$

$\mathsf{Weak\text{-}AsyAnam\text{-}IND\text{-}CPA}^{b}_{\Sigma}(\lambda, \mathcal{D})$

1 : $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \leftarrow^{\$} \mathsf{AT.Gen}(\lambda)$
2 : $(m, \widehat{m}_0, \widehat{m}_1) \leftarrow^{\$} \mathcal{D}(\mathsf{apk}, \mathsf{dk})$
3 : $c \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m}_b)$
4 : **return** $\mathcal{D}(c)$

Figure 7.3: Weak Asymmetric Anamorphic Encryption security game.

Definition 7.1.4 (Weak Asymmetric Anamorphic Encryption)**.** *An Anamorphic Encryption scheme* Π *equipped with an anamorphic triplet* Σ *is a Weak Asymmetric Anamorphic Encryption scheme if for every* PPT *distinguisher* $\mathcal{D}$

$$\mathsf{Adv}^{\mathsf{weak\text{-}asy\text{-}anam}}_{\mathcal{D},\Sigma}(\lambda) \leq \mathsf{negl}(\lambda).$$

7.1.3 Other Variants of AE

Robustness of Anamorphic Encryption. Introduced in [10] this notion dictates that it is hard to find a message m that, when encrypted normally (i.e., using $\mathsf{E.Enc}$) and then

anamorphically decrypted (i.e. using $\mathsf{AT.Dec}$) results in some $\widehat{m} \neq \perp$. Later, in [107], the notion has been extended to sender anamorphic encryption, requiring in addition to the previous property, also that there exists a negligible probability of decrypting $\widehat{m} \neq \perp$ using a different secret key from the one corresponding to the public key used to anamorphically encrypt $\widehat{m}$.

Fully Asymmetric AE. Introduced in [35], this notion is reminiscent of *Single-Receiver* AE from [78]. Informally, a Fully Asymmetric-AE guarantees the privacy of both the regular and the anamorphic messages with respect to users having access *also* to dk (but not to ask and tk of course). In the relation between these properties has been explored in [35]. It is also easy to observe that such stronger notion implies Weak Asymmetric AE.

7.2 Anamorphic Encryption from Black-Box PKE

7.2.1 Ideal PKE

In this section we model an idealized (and inefficient) PKE scheme, inspired by the one presented in [61, 111], accessible through three oracles $\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec}$. Internally the scheme is defined by two random functions ϕ and ψ tracking respectively the relation between public/secret keys, and the one between messages/ciphertexts. More in detail SK, PK are the secret and public keys sets while $\{0,1\}^\mu$, $\{0,1\}^\rho$, $\{0,1\}^\ell$ are respectively the messages, randomness (for encryption) and ciphertexts spaces. Then ϕ, ψ are sampled so that

- $\phi : \mathsf{SK} \to \mathsf{PK}$ is a uniformly random bijection.
- $\psi : \mathsf{PK} \times \{0,1\}^\mu \times \{0,1\}^\rho \to \{0,1\}^\ell$ random function s.t. $\psi(\mathsf{pk}, \cdot, \cdot)$ is injective.

Note that at this stage we do not constrain μ, ρ, ℓ, that are respectively the bit-size of messages, randomness and ciphertexts. Some later results will however only apply for certain parameters choice.

$\mathsf{E.Gen}(\lambda; \mathsf{sk})$

1 : $\mathsf{pk} \leftarrow \phi(\mathsf{sk})$
2 : **return** $(\mathsf{pk}, \mathsf{sk})$

$\mathsf{E.Enc}(\mathsf{pk}, m; r)$

1 : $c \leftarrow \psi(\mathsf{pk}, m, r)$
2 : **return** c

$\mathsf{E.Dec}(\mathsf{sk}, c)$

1 : $\mathsf{pk} \leftarrow \phi(\mathsf{sk})$
2 : **for** $(m, r) \in \{0,1\}^\mu \times \{0,1\}^\rho$
3 : **if** $\psi(\mathsf{pk}, m, r) = c$: **return** m
4 : **return** $\perp$.

Figure 7.4: Ideal PKE with $\phi : \mathsf{SK} \to \mathsf{PK}$ and $\psi : \mathsf{PK} \times \{0,1\}^\mu \times \{0,1\}^\rho \to \{0,1\}^\ell$ as above.

It is easy to observe that this scheme achieves semantic security (IND-CPA) if $\rho = \Omega(\lambda)$ and $|\mathsf{SK}| = \Omega(2^\lambda)$ as ciphertexts are random strings, and distinguishing the encryptions of two different messages requires a number of queries to $\mathsf{E.Enc}$ exponential in ρ.

7.2.2 Black-Box Anamorphic Encryption

Definition 7.2.1 (Black-Box Anamorphic Triplet)**.** *A triplet* $\Sigma = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ *is said to be a black-box anamorphic triplet (for any PKE* Π*) if every algorithm in* Σ *can access the procedures in* Π ***only*** *through oracle access, i.e. providing input and random coins to these procedures and obtaining* ***only*** *the output of such procedures call in return.*

We remark that we may occasionally and informally refer to an Black-Box Anamorphic Triplet as a Black-Box *Anamorphic Encryption*.

7.2.3 General Properties

Assume there exists a generic compiler $\Sigma = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ turning any IND-CPA secure PKE into an anamorphic encryption scheme, accessing the underlying PKE algorithms only through oracle queries. We can then study the behavior of such construction when applied to the ideal PKE $\Pi = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ defined in Figure 7.4. A first property it has to satisfy is that, up to negligible probability, the public and secret anamorphic keys have to be a valid key pair for the underlying PKE.

Lemma 7.2.1. *If* $\Sigma = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ *is an anamorphic triplet for the ideal PKE* Π*, then there exists a negligible* ε *such that*

$$(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \leftarrow^{\$} \mathsf{AT.Gen}(\lambda) \quad \Rightarrow \quad \Pr\left[\phi(\mathsf{ask}) \neq \mathsf{apk}\right] \leq \varepsilon(\lambda).$$

Proof. Let $\mathcal{A}$ be a PPT adversary playing the game in Definition 7.1.2 which on input pk, sk, runs the key generation algorithm $(\mathsf{pk}', \mathsf{sk}) \leftarrow \mathsf{E.Gen}(\lambda; \mathsf{sk})$ and returns 1 if $\mathsf{pk} = \mathsf{pk}'$ and 0 otherwise. From the definition of $\mathsf{E.Gen}$ in Figure 7.4, the secret key coincides with the random tape of $\mathsf{E.Gen}$. Thus in the real game $\mathsf{pk}' = \mathsf{pk}$ occurs always. Conversely in the anamorphic game, the adversary receives $\mathsf{apk}, \mathsf{ask}$ generated through $\mathsf{AT.Gen}$. Again by construction $\mathsf{pk}' = \phi(\mathsf{ask})$, meaning $\mathcal{A}$ returns 1 if and only if $\mathsf{apk} = \phi(\mathsf{ask})$. In conclusion

$$\mathsf{Adv}(\mathcal{A}) \;=\; |1 - \Pr\left[\phi(\mathsf{ask}) = \mathsf{apk}\right]| \;=\; \Pr\left[\phi(\mathsf{ask}) \neq \mathsf{apk}\right]$$

which is negligible as we assumed Σ to be an anamorphic triplet for the ideal PKE. □

The next property we study informally states that ciphertexts have to be unpredictable enough. While this could be stated in terms of (pseudo) min-entropy, for our purpose the following less general formulation will suffice.

Lemma 7.2.2. *Given* $\Sigma = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ *a black-box anamorphic triplet and uniformly sampled* s, r *and messages* $m, \widehat{m}$*, let*

$$(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \leftarrow \mathsf{AT.Gen}(\lambda; s), \qquad c \leftarrow \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m}; r).$$

For any set S *independent from* r*, with* $|S| \leq \mathsf{poly}(\lambda)$ *then* $\Pr[c \in S] \leq \mathsf{negl}(\lambda)$*.*

Proof. Consider the following adversary $\mathcal{A}$ against the anamorphic security game in Definition 7.1.2 instantiated when Σ is combined with the ideal PKE with $\rho = \Omega(2^{\lambda})$. Its attack

$\mathcal{A}^{\mathcal{O}}(\mathsf{apk}, \mathsf{ask})$:

1 : Sample $m \leftarrow^{\$} \{0,1\}^{\mu}$ and $\widehat{m} \leftarrow^{\$} \widehat{M}$
2 : $c_1 \leftarrow \mathcal{O}(m, \widehat{m})$
3 : $c_2 \leftarrow \mathcal{O}(m, \widehat{m})$
4 : **return** $c_1 == c_2$

Figure 7.5: Adversary against the security game in Figure 7.1. $\mathcal{O}$ is the encryption oracle provided in both RealG and AnamorphicG.

consists in encrypting twice a random message pair, and checking if the resulting ciphertexts are the same, see Figure 7.5.

If $c \in S$ with significant probability, as this set has polynomially bounded size, two ciphertexts sampled independently from it will collide with noticeable probability, allowing $\mathcal{A}$ to distinguish the two games.

More formally, in the real game $c_1 = c_2$ only if the random coins used to produce both ciphertexts are the same, which occurs with probability $2^{-\rho}$. To analyze the anamorphic game let

$$V_\delta \;=\; \{(m_0, \widehat{m}_0, s_0) \;:\; \Pr\left[c \in S \,|\, m = m_0,\; \widehat{m} = \widehat{m}_0,\; s = s_0\right] \geq \delta\}.$$

Using a variant of Markov inequality we can then prove that

Claim 7.2.1. *$\delta = 1/2 \cdot \Pr[c \in S]$ implies that $\Pr[(m, \widehat{m}, s) \in V_\delta] \geq \delta$.*

Calling for notational simplicity $\mathbf{v} = (m, \widehat{m}, s)$, it can now be shown that for all $\mathbf{v}_0 \in V_\delta$, $\Pr[c_1 = c_2 \,|\, \mathbf{v} = \mathbf{v}_0] =$

$$\begin{aligned}
&= \Pr\left[c_1 = c_2 \,|\, c_1, c_2 \in S,\; \mathbf{v} = \mathbf{v}_0\right] \cdot \Pr\left[c_1 \in S,\; c_2 \in S \,|\, \mathbf{v} = \mathbf{v}_0\right] \\
&\geq |S|^{-1} \cdot \Pr\left[c_1 \in S,\; c_2 \in S \,|\, \mathbf{v} = \mathbf{v}_0\right] \\
&= |S|^{-1} \cdot \Pr\left[c_1 \in S \,|\, \mathbf{v} = \mathbf{v}_0\right] \cdot \Pr\left[c_2 \in S \,|\, \mathbf{v} = \mathbf{v}_0\right] \\
&\geq |S|^{-1} \cdot \delta^2
\end{aligned}$$

where the second equality follows as c_1, c_2 are mutually independent conditioned on $\mathbf{v} = \mathbf{v}_0$, as in that case they are only a function of the (independently sampled) random coins used to compute them, and the random subset S is distributed independently from them. As a consequence $\Pr[c_1 = c_2 \,|\, \mathbf{v} \in V_\delta] \geq |S|^{-1} \cdot \delta^2$, which allow us to lower bound the probability $\mathcal{A}$ finds a collision in the anamorphic game as, fixing $\delta = 1/2 \cdot \Pr[c \in S]$,

$$\Pr[c_1 = c_2] \;\geq\; \Pr[c_1 = c_2 \,|\, \mathbf{v} \in V_\delta] \cdot \Pr[\mathbf{v} \in V_\delta] \;\geq\; \delta^3 \cdot |S|^{-1}.$$

Combining this with the bound on the collision probability in the real game, the advantage of $\mathcal{A}$ is then bounded by $\mathsf{Adv}(\mathcal{A}) \geq \delta^3 \cdot |S|^{-1} - 2^{-\rho}$. Having set $\delta = 1/2 \cdot \Pr[c \in S]$ we conclude the proof as we assumed $\rho = \Omega(\lambda)$, $|S|$ polynomially bounded and the black-box anamorphic triplet to be secure. □

A consequence of the above result is that $\mathsf{AT.Enc}$ almost never returns a ciphertext that was observed by $\mathsf{AT.Gen}$. To formally state this, we first define this set of ciphertexts.

Definition 7.2.2. *Given a black-box anamorphic triplet Σ we define $E_{\mathsf{in}}^{\mathsf{Gen}}$, $E_{\mathsf{in}}^{\mathsf{Enc}}$ the sets of tuples (pk, m, r, c) such that respectively $\mathsf{AT.Gen}$ and $\mathsf{AT.Enc}$ on input in eventually query $c = \mathsf{E.Enc}(\mathsf{pk}, m; r)$. Analogously, $D_{\mathsf{in}}^{\mathsf{Gen}}$, $D_{\mathsf{in}}^{\mathsf{Enc}}$ are the sets of tuples (sk, c, m) such that respectively $\mathsf{AT.Gen}$ and $\mathsf{AT.Enc}$ on input in computes $m = \mathsf{E.Dec}(\mathsf{sk}, c)$.*

Definition 7.2.3. *Given a black-box anamorphic triplet Σ we define the set of ciphertexts observed by $\mathsf{AT.Gen}$ on input s as*

$$C_s^{\mathsf{Gen}} \coloneqq \{c \,:\, (\cdot,\cdot,\cdot,c) \in E_s^{\mathsf{Gen}} \;\vee\; (\cdot,c,\cdot) \in D_s^{\mathsf{Gen}}\}.$$

Corollary 7.2.3. *With the same notation of Lemma 7.2.2,* $\Pr\left[c \in C_s^{\mathsf{Gen}}\right] \leq \mathsf{negl}(\lambda)$.

7.2.4 Ciphertext Selection Lemma

The core technical result of this section is a characterization of the encryption procedure for a black-box anamorphic triplet. Informally, our result states that such procedure can only obtain *valid* ciphertexts through encryption queries to $\mathsf{E.Enc}$ and then return one of them. This is perhaps not surprising as there is no assumption on the underlying PKE scheme. Thus, no meaningful manipulation of ciphertexts after their generation is possible. This intuition is captured by the following *ciphertext selection lemma*. First, we formally define the set of valid ciphertexts queried by $\mathsf{AT.Enc}$.

Definition 7.2.4. *Given input $\mathsf{in} = (\mathsf{apk}, \mathsf{ask}, m, \widehat{m}, r)$ the set of valid ciphertexts queried by $\mathsf{AT.Enc}$ is $C_{\mathsf{in}}^{\mathsf{Enc}} = \{c \,:\, (\mathsf{apk}, m, \cdot, c) \in E_{\mathsf{in}}^{\mathsf{Enc}}\}$.*

We recall that our ideal PKE is parametrized by μ, ρ, ℓ, respectively the message, random coins and ciphertext bit-length. Notably, the following result requires $\ell - \rho = \Omega(\lambda)$ to hold. This means the lemma cannot be specialized to black-box anamorphic schemes where the underlying PKE is assumed to have *small* message space $\mu = O(\log \lambda)$ and *dense* ciphertext space $\ell = \rho + \mu + O(\log \lambda)$, i.e. such that a noticeable fraction of strings with length ℓ are valid ciphertexts. This is actually no coincidence as in this case efficient "semi-generic" constructions do exist [38].

Lemma 7.2.4. *Given $\Sigma = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ a black-box anamorphic triplet, let r, s be uniform random coins and $m, \widehat{m}$ uniformly sampled messages. Setting*

$$(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \leftarrow \mathsf{AT.Gen}(\lambda; s), \quad \mathsf{in} = (\mathsf{apk}, \mathsf{dk}, m, \widehat{m}, r), \quad c \leftarrow \mathsf{AT.Enc}(\mathsf{in}),$$

if $\rho = \Omega(\lambda)$ and $\ell - \rho = \Omega(\lambda)$, then $\Pr\left[c \notin C_{\mathsf{in}}^{\mathsf{Enc}}\right] \leq \mathsf{negl}(\lambda)$.

Proof. To prove the lemma let $\mathcal{A}$ be an adversary against the anamorphic security definition as described in Figure 7.6. Given $(\mathsf{apk}, \mathsf{ask})$ it requests the encryption c of a random message m and locally decrypts it computing $m' = \mathsf{E.Dec}(\mathsf{ask}, c)$. It returns 1 if and only if $m \neq m'$.

Since the ideal PKE scheme achieves perfect correctness $\mathcal{A}$ never returns 1 when executed in the real game. To study the anamorphic game, let s be the random tape of $\mathsf{AT.Gen}$, so that

$\mathcal{A}^{\mathcal{O}}(\mathsf{apk}, \mathsf{ask})$:

1 : Sample $m \leftarrow^{\$} \{0,1\}^{\mu}$ and $\widehat{m} \leftarrow^{\$} \widehat{M}$
2 : $c \leftarrow \mathcal{O}(m, \widehat{m})$
3 : $m' \leftarrow \mathsf{E.Dec}(\mathsf{ask}, c)$
4 : **return** 1 if $m \neq m'$

Figure 7.6: Adversary for the anamorphism game (Fig. 7.1). $\mathcal{O}$ is the encryption oracle.

$(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \leftarrow \mathsf{AT.Gen}(\lambda; s)$, and r the one of $\mathsf{AT.Enc}$ when executed to answer $\mathcal{A}$'s only query. For notational convenience $\mathsf{in} = (\mathsf{apk}, \mathsf{dk}, m, \widehat{m}, r)$ so that $c = \mathsf{AT.Enc}(\mathsf{in})$. We then define the two events

$$\mathsf{Bad} \ : \ \phi(\mathsf{ask}) \neq \mathsf{apk} \ \vee \ c \in C_s^{\mathsf{Gen}} \qquad \mathsf{Good} \ : \ c \in C_{\mathsf{in}}^{\mathsf{Enc}}.$$

Lemma 7.2.1 and Corollary 7.2.3 together imply that $\Pr[\mathsf{Bad}] \leq \mathsf{negl}(\lambda)$. Next we claim that the following probability is also negligible.

Claim 7.2.2. $\Pr[m = m', \neg\mathsf{Bad}, \neg\mathsf{Good}] \leq \mathsf{negl}(\lambda)$.

These two inequalities immediately imply the thesis as, through a union bound

$$\begin{aligned} \Pr[m = m'] \ &\leq \ \Pr[m = m', \neg\mathsf{Bad}, \neg\mathsf{Good}] + \Pr[\mathsf{Bad}] + \Pr[\mathsf{Good}] \\ &\leq \ \Pr[\mathsf{Good}] + \mathsf{negl}(\lambda). \end{aligned}$$

By our initial observation $\mathsf{Adv}^{\mathsf{anam}}(\mathcal{A}) = \Pr[m \neq m']$ with m' distributed as in the anamorphic game. As a consequence $\Pr[\neg\mathsf{Good}] \leq \mathsf{Adv}^{\mathsf{anam}}(\mathcal{A}) + \mathsf{negl}(\lambda)$, that is negligible. □

Proof of Claim 7.2.2. Let $C = C_s^{\mathsf{Gen}} \cup C_{\mathsf{in}}^{\mathsf{Enc}}$. We denote V_m the set of ciphertexts encrypting m under apk, that is $V_m = \{\psi(\mathsf{apk}, m, r) \ : \ r \in \{0,1\}^{\rho}\}$. The claim can then be translated in terms of C and V_m. Indeed, if the studied event occurs then $c \notin C$. Similarly $m = m'$ and $\neg\mathsf{Bad}$ both implies that $m = \mathsf{E.Dec}(\mathsf{ask}, c) \ \Rightarrow \ \psi(\phi(\mathsf{ask}), m, r) = c \ \Rightarrow \ \psi(\mathsf{apk}, m, r) = c$ for some r, which means $c \in V_m$. Therefore

$$\begin{aligned} (m = m', \neg\mathsf{Bad}, \neg\mathsf{Good}) \ &\Rightarrow \ c \in V_m \setminus C \quad \Rightarrow \\ \Rightarrow \quad \Pr[m = m', \neg\mathsf{Bad}, \neg\mathsf{Good}] \ &\leq \ \Pr[c \in V_m \setminus C]. \end{aligned}$$

To prove the latter probability to be negligible, let q be a bound on the total queries of $\mathsf{AT.Gen}$ and $\mathsf{AT.Enc}$. Let $c_1, \ldots, c_d$ be the (ordered) ciphertexts $\mathsf{AT.Enc}$ queries to $\mathsf{E.Dec}(\mathsf{ask}, \cdot)$ and for notational convenience we name $c_{d+1} \coloneqq c$. Let C_i be the set of ciphertext either returned by $\mathsf{E.Enc}(\mathsf{apk}, \cdot, \cdot)$ or queried to $\mathsf{E.Dec}(\mathsf{ask}, \cdot)$ by either $\mathsf{AT.Gen}(\lambda; s)$ or $\mathsf{AT.Enc}(\mathsf{in})$ *before* the latter queries $\mathsf{E.Dec}(\mathsf{apk}, c_i)$. Note this means $C_i \subseteq C \cup \{c_1, \ldots, c_{i-1}\}$. Crucially, given only this information, the set of ciphertexts $V_m \setminus C_i$ is uniformly distributed over $\{0,1\}^{\ell} \setminus C_i$. Once again the event above can be decomposed through a chain of implications:

$$\begin{aligned} c \in V_m \setminus C \ &\Rightarrow \ \bigvee_{i=1}^{d+1} (c_i \in V_m \setminus C \ \wedge \ \{c_1, \ldots, c_{i-1}\} \cap V_m \setminus C = \varnothing) \\ &\Rightarrow \ \bigvee_{i=1}^{d+1} (c_i \in V_m \setminus (C \cup \{c_1, \ldots, c_{i-1}\})) \\ &\Rightarrow \ \bigvee_{i=1}^{d+1} (c_i \in V_m \setminus C_i). \end{aligned}$$

Using a union bound, along with the fact that $V_m \setminus C_i$ is a uniformly distributed subset of $\{0,1\}^\ell \setminus C_i$ and independent from c_i, we can conclude that

$$\begin{aligned}\Pr[c \in V_m \setminus C] &\leq \sum_{i=1}^{d+1} \Pr[c_i \in V_m \setminus C_i] \\ &\leq \sum_{i=1}^{d+1} \frac{|V_m \setminus C_i|}{|\{0,1\}^\ell \setminus C_i|} \leq (d+1) \cdot \frac{2^\rho}{2^\ell - q}\end{aligned}$$

with the last quantity being negligible as we assumed $\ell - \rho = \Omega(\lambda)$ while d, q are polynomially bounded. □

Remark 7.2.1. *Lemma 7.2.4 holds only for stateless anamorphic triplets. If stateful encryption/decryption is allowed, then we can only prove a slightly weaker result. Specifically c has to lie, with overwhelming probability, in the set of valid ciphertexts observed by* AT.Enc *and* AT.Gen *(as opposed to only* AT.Enc*). We stress this to be sufficient for a slightly weaker version of Theorem 7.4.1 (See Remark 7.4.1) to hold true. The proof is analogous up to the fact that Corollary 7.2.3 cannot be applied anymore.*

7.2.5 Symmetric Choice Functions

Thanks to the Ciphertext Selection Lemma, the encryption procedure of any black-box anamorphic triplet can be abstracted as a process observing a list of ciphertexts and eventually choosing one of them. We will call such a function returning one of its arguments a *choice function*. In this section we show this class of functions satisfies interesting properties, which will be useful in the proof of Theorem 7.4.2. First we provide a formal definition of choice functions and in particular *symmetric* ones, which do not depend on the order of their arguments.

Definition 7.2.5. *Given a finite set X, a random function $f \sim \{g : X^k \to X\}$ is a choice function if $f(x_1, \ldots, x_k) \in \{x_1, \ldots, x_k\}$ for all $x_1, \ldots, x_k \in X$. Furthermore, a choice function is called symmetric if for any permutation π we have $f(x_1, \ldots, x_k) = f(x_{\pi(1)}, \ldots, x_{\pi(k)})$.*

A rather non-trivial property of symmetric choice functions is that they are *consistent* with their choices. More specifically, assume that on random inputs $u_1, \ldots, u_k$ the function $f(u_1, \ldots, u_k)$ chose z among them. Then given more random inputs $v_2, \ldots, v_k$, the function $f(z, v_2, \ldots, v_k)$ will chose z again with probability at least $\approx 1/k$. At first sight this might seem trivial, as z could appear to be random and f unable to distinguish it from the other elements. However this reasoning is incorrect. Indeed, although z is chosen from uniformly sampled variables, this choice can bias its distribution. The above intuition is therefore wrong, but we nevertheless prove this lower bound with the following Lemma.

Lemma 7.2.5. *Let $f \sim \{g : X^k \to X\}$ be a symmetric choice function. Given $\mathbf{u} \sim U(X^k)$, $\mathbf{v} \sim U(X^{k-1})$ uniformly distributed, let $z = f(\mathbf{u})$. Then*

$$\Pr[f(z, \mathbf{v}) = z] \geq \frac{1}{k} - O\left(\frac{1}{|X|}\right).$$

Proof of Lemma 7.2.5. Let $n = |X|$ and $P(x_1, \ldots, x_k) = \Pr[f(x_1, \ldots, x_k) = x_1]$. By definition of choice function f has to return one of its arguments, meaning that for $x_1, \ldots, x_k$ all

distinct

$$P(x_1,\dots,x_k)+P(x_2,\dots,x_k,x_1)+\dots+P(x_k,x_1,\dots,x_{k-1}) \;=\; 1.$$

As a first step we state some properties of P.

Claim 7.2.3. *The following bounds for the sum of P over X^k holds:*

$$\sum\nolimits_{\mathbf{x}} P(\mathbf{x}) \le n^k, \qquad\qquad \sum\nolimits_{\mathbf{x}} P(\mathbf{x}) \ge \frac{n^k}{k} - kn^{k-1}.$$

Next we study the distribution of $z=f(\mathbf{u})$.

Claim 7.2.4. *For all $a\in X$, $\Pr[z=a] \ge \left(\frac{k}{n^k}\sum_{\mathbf{x}} P(a,\mathbf{x})\right) - \frac{k^3}{n^2}$.*

Using both claim, the theorem's proof follows as

$$\begin{aligned}
\Pr[f(z,\mathbf{v})=z] \;&=\; \sum\nolimits_{\mathbf{y}} \frac{1}{n^{k-1}}\cdot \Pr[f(z,\mathbf{y})=z] \\
&=\; \frac{1}{n^{k-1}}\sum\nolimits_{a,\mathbf{y}} \Pr[z=a]\Pr[f(a,\mathbf{y})=a] \\
&\ge\; \frac{1}{n^{k-1}}\sum\nolimits_{a,\mathbf{y}} \left(\sum\nolimits_{\mathbf{x}} \frac{k}{n^k}P(a,\mathbf{x}) - \frac{k^3}{n^2}\right) P(a,\mathbf{y}) \\
&=\; \frac{k}{n^{2k-1}}\sum\nolimits_{a,\mathbf{y},\mathbf{x}} P(a,\mathbf{x})P(a,\mathbf{y}) - \frac{k^3}{n^{k+1}}\sum\nolimits_{a,\mathbf{y}} P(a,\mathbf{y}) \\
&\ge\; \frac{k}{n^{2k-1}}\sum\nolimits_{a} \left(\sum\nolimits_{\mathbf{x}} P(a,\mathbf{x})\right)^2 \;-\; \frac{k^3}{n} \\
&\ge\; \frac{k}{n^{2k-1}}\cdot\frac{1}{n}\left(\frac{n^k}{k} - k\cdot n^{k-1}\right)^2 - O(n^{-1}) \\
&=\; \frac{k}{n^{2k}}\cdot\left(\frac{n^{2k}}{k^2} + (n^{k-1}k)^2 - 2n^{2k-1}\right) - O(n^{-1}) \\
&=\; \frac{k}{n^{2k}}\cdot\frac{n^{2k}}{k^2} - O(n^{-1}) \;=\; \frac{1}{k} - O(n^{-1}).
\end{aligned}$$

Where the first inequality follows by Claim 7.2.4, the second one applying Claim 7.2.3 on the second term. The third inequality follows from AM-QM where, calling $s(a)=\sum_{\mathbf{x}} P(a,\mathbf{x})$, the sum of $s(a)$ coincides with the sum of P over X^k, and is therefore lower bounded as per Claim 7.2.3. □

Proof of Claim 7.2.3. The first part is trivial as $P(\mathbf{x})\le 1$. For the second part let $S = \{(x_1,\dots,x_k)\in X^k \;:\; \forall i,j (x_i\neq x_j)\}$. The size of $X^k\setminus S$ is smaller than $\binom{k}{2}\cdot n^{k-1}$, as it is a union of the $\binom{k}{2}$ sets $D_{i,j}$ containing all vectors $\mathbf{x}$ with $x_i=x_j$ (so that $|D_{i,j}|=n^{k-1}$). As a consequence then $|S|\ge n^k - \binom{k}{2}n^{k-1}$.

Next we can partition S into a collection $\mathcal{P}$ of $|S|/k$ classes of size k, each containing the cyclic shift of a vector $\mathbf{x}\in S$. Formally

$$[(x_1,\dots,x_k)] \;\coloneqq\; \{(x_{1+i},\dots,x_{k+i}) \;:\; i\in\mathbb{Z}/k\mathbb{Z}\}$$

note that the vectors in S have entries that are all distinct, so each such cyclic shift produces a different vector. Moreover, as observed previously, the sum of $P(\mathbf{x})$ for $\mathbf{x} \in [\mathbf{x}]$ equals 1, as the choice function must return one of its entries. We thus conclude that

$$\sum_{\mathbf{x}\in X^k} P(\mathbf{x}) \;\geq\; \sum_{\mathbf{x}\in S} P(\mathbf{x}) \;=\; \frac{|S|}{k} \;\geq\; \frac{n^k}{k} - \binom{k}{2}\frac{n^{k-1}}{k} \;\geq\; \frac{n^k}{k} - kn^{k-1}. \qquad \square$$

Proof of Claim 7.2.4. Let $S = \{(x_2, \ldots, x_k) \in X^{k-1} \;:\; \forall i, j(x_i \neq a,\; x_i \neq x_j)\}$. To lower bound its size let D_i the set of points in X^{k-1} with i-th coordinate equal to a and $D_{i,j}$ the subset of X^{k-1} with $x_i = x_j$. Then[1]

$$|X^{k-1} \setminus S| = \left|\bigcup_{i=2}^{k} D_i \cup \bigcup_{i<j} D_{i,j}\right| \leq kn^{k-2} + \binom{k-1}{2} n^{k-2} \leq k^2 n^{k-2}.$$

Thus $|S| \geq n^{k-1} - k^2 n^{k-2}$. We can finally lower bound the probability that $z = a$ as

$$\begin{aligned}\Pr[z = a] \;&\geq\; k \sum_{\mathbf{x}\in S} P(a, \mathbf{x}) \frac{1}{n^k} \;\geq\; \frac{k}{n^k} \sum_{\mathbf{x}\in X^{k-1}} P(a, \mathbf{x}) \;-\; \frac{k^3}{n^k} \sum_{\mathbf{x}\in X^{k-1}\setminus S} P(a, \mathbf{x}) \\ &\geq\; \frac{k}{n^k} \sum_{\mathbf{x}\in X^{k-1}} P(a, \mathbf{x}) \;-\; \frac{k^3}{n^2}.\end{aligned}$$

The first bound follows by restricting all components of $\mathbf{u}$ to be different, lower bounding the probability of this not happening with 0, and later, as $z = a \;\Rightarrow\; a \in \{u_1, \ldots, u_k\}$, grouping all vectors shifting the (only) entry equal to a in the first position (meaning that each term $P(a, \mathbf{x})$ is repeated k times). $\square$

7.3 Random Oracle Channels

7.3.1 Definitions

In order to provide lower bounds for black-box Anamorphic Encryption, we first study a simpler scenario where a *sender* $\mathcal{S}$ has to communicate a message $m \in M$ to a *receiver* $\mathcal{R}$ under some constraints. In particular, both parties have access to a random oracle H and $\mathcal{S}$, which obtains values $y_1, \ldots, y_k$ during its interaction with H, can only chose one of them and send it to $\mathcal{R}$, who eventually has to recover the original message. We will call this setting a *Random Oracle Channel*.

Definition 7.3.1. *A RO-channel is a tuple $(\mathcal{S}, \mathcal{R}, M, k, h)$ with $\mathcal{S}, \mathcal{R}$ Probabilistic Turing Machines (not necessarily PPT), $M \subseteq \{0,1\}^*$ and $k, h = \mathsf{poly}(\lambda)$ such that*

1. *$\mathcal{S}, \mathcal{R}$ make respectively at most k and h queries to H.*
2. *$\forall m \in M$, calling $y_j = \mathsf{H}(x_j)$ with $j \in \{1, \ldots, k\}$ the queries $\mathcal{S}^{\mathsf{H}}(m)$ performs, then $\mathcal{S}^{\mathsf{H}}(m) \to y_i$ for some $i \in \{1, \ldots, k\}$.*

[1] Here we assume $\binom{n}{m} = 0$ when $n < m$.

3. *There exists a negligible $\varepsilon(\lambda)$ such that $\forall m \in M$ and uniformly sampled common random tape s*
$$\Pr\left[m \neq m' \,\middle|\, y \leftarrow \mathcal{S}^{\mathsf{H}}(m;s),\ m' \leftarrow \mathcal{R}^{\mathsf{H}}(y;s)\right] \leq \varepsilon(\lambda).$$

The main problem about RO channels is determining how large can $|M|$ be as a function of k, h. Intuitively, due to the high limitations imposed on $\mathcal{S}, \mathcal{R}$, we expect $|M|$ to be small, and indeed our results eventually implies that $|M| = \mathsf{poly}(\lambda)$ or that, equivalently, in this setting it is possible to communicate at most $O(\log \lambda)$ bits.

7.3.2 Bounds for RO-Channels

Theorem 7.3.1. *For any RO-Channel $(\mathcal{S}, \mathcal{R}, M, k, h)$ we have that asymptotically $|M| \leq 2(h+k)^2$. In particular $|M| = \mathsf{poly}(\lambda)$.*

Proof. The result is proven by showing that any RO-channel can be compiled into two unbounded $\mathcal{S}^*$, $\mathcal{R}^*$ with shared randomness that reliably communicate a message $m \in M$ by only sending $\ell = O(\log \lambda)$ bits. More specifically the shared randomness is of the form (F, G, s) with $F : \{0,1\}^{\mathsf{poly}(\lambda)} \to \{0,1\}^{\ell}$ and $G : \{0,1\}^{\ell} \to \{0,1\}^{\lambda}$ random functions, and s the random tape used by $\mathcal{S}, \mathcal{R}$.

$\mathcal{S}^*$ on input m executes $\mathcal{S}(m;s)$ and simulates the RO through the function $G \circ F$. More formally, when $\mathcal{S}$ queries the RO on input x_i, it returns $y_i = G(F(x_i))$ and locally stores $z_i = F(x_i)$. Finally, once $\mathcal{S}$ chooses its output y_i, $\mathcal{S}^*$ returns $z_i \in \{0,1\}^{\ell}$. In order to recover m, $\mathcal{R}^*$ internally executes $\mathcal{R}$ simulating the RO as before. A full description of $\mathcal{S}^*, \mathcal{R}^*$ is provided in Figure 7.7.

$\mathcal{S}^*(m;F,G,s)$:		$\mathcal{R}^*(z;F,G,s)$:	
1 :	Run $\mathcal{S}(m;s)$	1 :	Run $\mathcal{R}(G(z);s)$
2 :	**When** $\mathcal{S}$ queries x_i:	2 :	**When** $\mathcal{R}$ queries x_i:
3 :	$z_i \leftarrow F(x_i),\ y_i \leftarrow G(z_i)$	3 :	$y_i \leftarrow G(F(x_i))$
4 :	**Reply** with $\mathcal{S} \leftarrow y_i$	4 :	**Reply** with $\mathcal{R} \leftarrow y_i$
5 :	**When** $\mathcal{S}$ returns y_i:	5 :	**When** $\mathcal{R}$ returns m:
6 :	**return** z_i	6 :	**return** m

Figure 7.7: Unbounded $\mathcal{S}^*, \mathcal{R}^*$ using $(\mathcal{S}, \mathcal{R})$ to communicate m by only sending ℓ bits.

Let δ be the probability that $\mathcal{S}^*$ and $\mathcal{R}^*$ fail to communicate correctly, i.e.
$$\delta \ := \ \Pr\left[m \neq m' \,|\, m' \leftarrow \mathcal{R}^*(z;F,G,s),\ z \leftarrow \mathcal{S}^*(m;F,G,s)\right].$$

Then, the success probability $1-\delta$ is bounded by the conditional min-entropy of m given z. This implies that
$$\mathrm{H}_\infty(m\,|\,z) \ \geq \ \mathrm{H}_\infty(m) - \ell \ = \ \log_2|M| - \ell \quad \Rightarrow \quad (1-\delta) \ \leq \ 2^{-\mathrm{H}_\infty(m\,|\,z)} \ = \ \frac{2^\ell}{|M|}$$
$$\Rightarrow \quad |M| \leq \frac{2^\ell}{1-\delta}.$$

Where the first inequality follows from the fact that $z \in \{0,1\}^\ell$ [52, Lemma 2.2]. Next we study the success probability for the specific case of $\mathcal{S}^*, \mathcal{R}^*$ and a suitable choice of ℓ. Let X be the set of queries that, given $m \sim U(M)$ and a random tape s, the initial algorithms $\mathcal{S}$, $\mathcal{R}$ jointly performs to the RO. Calling Coll the event that two such points collides with respect to F, since $|X| \leq h + k$

$$\Pr[\mathsf{Coll}] \leq \frac{(h+k)^2}{2} \cdot \frac{1}{2^\ell}.$$

Next we observe that, as $G : \{0,1\}^\ell \to \{0,1\}^\lambda$ is a random function, if $\neg\mathsf{Coll}$, then $\mathcal{S}^*$, $\mathcal{R}^*$ perfectly simulate the RO. In particular, calling ε the error probability of the given RO-channel, i.e., $\Pr[m \neq m' \mid \neg\mathsf{Coll}]$, we have that

$$\begin{aligned}
\delta &= \Pr[m \neq m' \mid \mathsf{Coll}] \Pr[\mathsf{Coll}] + \Pr[m \neq m' \mid \neg\mathsf{Coll}] \Pr[\neg\mathsf{Coll}] \\
&\leq \Pr[\mathsf{Coll}] + \Pr[m \neq m' \mid \neg\mathsf{Coll}] \\
&\leq \frac{(h+k)^2}{2 \cdot 2^\ell} + \varepsilon.
\end{aligned}$$

Setting $\ell = 2\log(h+k)$ we obtain $1 - \delta \geq 1/2 - \varepsilon$ and in particular

$$|M| \leq \frac{2^{2\log(h+k)}}{1/2 - \varepsilon} = 2 \cdot (h+k)^2 + \mathsf{negl}(\lambda) \quad \Rightarrow \quad |M| \leq 2 \cdot (h+k)^2$$

where the equality holds because $\frac{1}{1-2\varepsilon} = 1 + \mathsf{negl}(\lambda)$ and last inequality holds asymptotically in λ as $|M|$ is an integer and $\mathsf{negl}(\lambda)$ is eventually less than 1. □

7.4 Lower Bounds and Impossibility

7.4.1 Communication Rate Lower Bound

In this section we answer our question on black-box anamorphic encryption proving that its anamorphic message space must be polynomially bounded, or equivalently that it is impossible to communicate more than $O(\log \lambda)$ bits per ciphertext. The main technique, as described in the introduction, is to combine the information-theoretic lower bound for RO-channel with the ciphertext-selection lemma. The latter indeed informally implies that communication using black-box anamorphic encryption scheme happens almost as in a RO-channel: the sender can only perform certain queries to $\mathsf{E.Enc}(\mathsf{apk}, m, \cdot)$ and eventually return one of the replies. Similarly, the receiver is allowed to query $\mathsf{E.Enc}(\mathsf{apk}, m, \cdot)$ to extract information about the sender's hidden message. We can thus present our first result.

Theorem 7.4.1. *Let* $\Sigma = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ *be a black-box anamorphic triplet with anamorphic message space* $\widehat{M}$*. Then* $|\widehat{M}| = \mathsf{poly}(\lambda)$*. More precisely, calling* q_e *and* q_d *the queries performed to* $\mathsf{E.Enc}$ *respectively by* $\mathsf{AT.Enc}$ *and* $\mathsf{AT.Dec}$*, then* $|\widehat{M}| \leq 2(q_e + q_d)^2$*.*

Proof. Applying the above black-box anamorphic triplet scheme to the ideal PKE $\Pi = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ defined in Section 7.2.1, we describe a RO-channel with anamorphic message space $\widehat{M}$. A detailed presentation of $\mathcal{S}, \mathcal{R}$ appears in Figure 7.8. Initially both procedures hold shared randomness used to setup the anamorphic encryption parameters, and later simulate the ideal PKE. This is of the form $(s^*, r^*, m^*, \phi^*, \psi^*, \xi^*)$ with

- (s^*, r^*): random tapes for $\mathsf{AT.Gen}$ and $\mathsf{AT.Enc}$.
- m^*: random regular (i.e. non anamorphic) message in $M = \{0,1\}^\mu$.
- ϕ^*: random bijection from SK to PK, as in the ideal PKE.
- ψ^*: random function mapping (pk, m, r) to ciphertexts in $\{0,1\}^\ell$.
- ξ^*: biased random function mapping $\mathsf{SK} \times \{0,1\}^\ell$ to $M \cup \{\perp\}$, such that $\xi^*(\mathsf{sk}, c) = m_0$ with probability $2^{\ell-\rho}$ for all $m_0 \in M$.

Given the above shared randomness $\mathcal{S}$, $\mathcal{R}$ proceed as follows:

1. Key Generation. Initially they both setup the Anamorphic Encryption parameters $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk})$ running $\mathsf{AT.Gen}(\lambda; s^*)$ (lines 1-6). In this phase, each time the key generation queries $\mathsf{E.Gen}(\lambda; \mathsf{sk})$, they use ϕ^* to reply with $(\phi^*(\mathsf{sk}), \mathsf{sk})$. When it queries an encryption $\mathsf{E.Enc}(\mathsf{pk}, m; r)$ they both reply with $\psi^*(\mathsf{pk}, m, r)$. When it queries a decryption $\mathsf{E.Dec}(\mathsf{sk}, c)$, if c was previously obtained as the encryption of some m they reply with m. Else, they reply with $\xi^*(\mathsf{sk}, c)$.

2. Encryption. $\mathcal{S}^{\mathsf{H}}(\widehat{m})$ proceeds computing c^*, the anamorphic encryption of $(m^*, \widehat{m})$ with keys $(\mathsf{apk}, \mathsf{dk})$ and randomness r^* (lines 9-14). During this computation, each time $\mathsf{AT.Enc}$ queries $\mathsf{E.Gen}(\lambda; \mathsf{sk})$ it replies as above using ϕ^*. When it queries an encryption $\mathsf{E.Enc}(\mathsf{pk}, m; r)$, if the same request was performed by $\mathsf{AT.Gen}$ it replies consistently, i.e. with $\psi^*(\mathsf{pk}, m; r)$. Otherwise it invokes its RO, replying with $c = \mathsf{H}(\mathsf{pk}, m, r)$. Decryption queries are handled as before. Finally it returns c^*.

3. Decryption. $\mathcal{R}$ on input c^* finally computes $\widetilde{m} \leftarrow \mathsf{AT.Dec}(\mathsf{ask}, \mathsf{tk}, c^*)$ (lines 9-14, right procedure). During this execution, each time $\mathsf{AT.Dec}$ queries $\mathsf{E.Gen}(\lambda; \mathsf{sk})$, it replies as above using ϕ^*. When it queries $\mathsf{E.Enc}(\mathsf{pk}, m; r)$ it replies with $\psi^*(\mathsf{pk}, m, r)$ if the same query was performed by $\mathsf{AT.Gen}$, or with $\mathsf{H}(\mathsf{pk}, m, r)$ otherwise. Finally, queries to $\mathsf{E.Dec}(\mathsf{sk}, c)$ are handled as before, with the exception that to $\mathsf{E.Dec}(\mathsf{ask}, c^*)$ it always replies with m^* (see line 7). Eventually it returns $\widetilde{m}$.

Given the above description of $\mathcal{S}, \mathcal{R}$ we proceed illustrating immediate properties they satisfy. First of all $\mathcal{S}$ returns up to negligible probability a value it received from the RO. This follows from the Ciphertext Selection Lemma (Lemma 7.2.4) and Lemma 7.2.3. Indeed, they imply $\mathsf{AT.Enc}$ will almost always return a ciphertext c it obtained from $\mathsf{E.Enc}$ and which was not observed by $\mathsf{AT.Gen}$, meaning that c is evaluated from H (as opposed to ψ^* to keep consistency with $\mathsf{AT.Gen}$'s view). Another immediate observation is that $\mathcal{S}$ and $\mathcal{R}$ respectively performs q_e and q_d RO calls, i.e. the number of queries to $\mathsf{E.Enc}$ respectively from $\mathsf{AT.Enc}$ and $\mathsf{AT.Dec}$. This follows as the RO may be called at most once for each such query.

To conclude that $(\mathcal{S}, \mathcal{R}, \widehat{M}, q_e, q_d)$ is a RO-Channel we only need to establish correctness. To do so we rely on the anamorphic encryption scheme's correctness, Section 7.1.1: given correctly generated keys and messages $(m, \widehat{m})$

$$\Pr\left[\widetilde{m} \neq \widehat{m} \,\middle|\, \widetilde{m} \leftarrow \mathsf{AT.Dec}(\mathsf{ask}, \mathsf{tk}, c),\ c \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})\right] \leq \mathsf{negl}(\lambda).$$

$\mathcal{S}^{\mathsf{H}}(\widehat{m}\,;(s^*,r^*,m^*,\phi^*,\psi^*,\xi^*))$:

1 : $(\mathsf{apk},\mathsf{ask},\mathsf{dk},\mathsf{tk}) \leftarrow \mathsf{AT.Gen}(\lambda;s^*)$
2 : **When** queried $\mathsf{E.Enc}(\mathsf{pk},m;r)$:
3 : Get $c \leftarrow \psi^*(\mathsf{pk},m,r)$
4 : Set $\xi^*(\mathsf{sk},c) \leftarrow m \;:\; \mathsf{pk} = \phi^*(\mathsf{sk})$
5 : Set $\mathsf{H}(\mathsf{pk},m,r) \leftarrow c$
6 : **Reply** c
7 :
8 : // Get the Anamorphic Encryption
9 : Run $c^* \leftarrow \mathsf{AT.Enc}(\mathsf{apk},m;r)$
10 : **When** queried $\mathsf{E.Enc}(\mathsf{pk},m,r)$:
11 : Get $c \leftarrow \mathsf{H}(\mathsf{pk},m,r)$
12 : Set $\xi^*(\mathsf{sk},c) \leftarrow m \;:\; \mathsf{pk} = \phi^*(\mathsf{sk})$
13 : **Reply** c
14 : **return** c^*
15 : // Key Gen. and Decryption query
16 : **When** queried $\mathsf{E.Gen}(\lambda;\mathsf{sk})$:
17 : **Reply** $(\phi^*(\mathsf{sk}),\mathsf{sk})$
18 : **When** queried $\mathsf{E.Dec}(\mathsf{sk},c)$:
19 : **Reply** $\xi^*(\mathsf{sk},c)$

$\mathcal{R}^{\mathsf{H}}(c^*\,;(s^*,r^*,m^*,\phi^*,\psi^*,\xi^*))$:

1 : $(\mathsf{apk},\mathsf{ask},\mathsf{dk},\mathsf{tk}) \leftarrow \mathsf{AT.Gen}(\lambda;s^*)$
2 : **When** queried $\mathsf{E.Enc}(\mathsf{pk},m;r)$:
3 : Get $c \leftarrow \psi^*(\mathsf{pk},m,r)$
4 : Set $\xi^*(\mathsf{sk},c) \leftarrow m \;:\; \mathsf{pk} = \phi^*(\mathsf{sk})$
5 : Set $\mathsf{H}(\mathsf{pk},m,r) \leftarrow c$
6 : **Reply** c
7 : Set $\xi^*(\mathsf{ask},c^*) \leftarrow m^*$
8 : // Decrypt the Anamorphic Ciphertext
9 : Run $\widetilde{m} \leftarrow \mathsf{AT.Dec}(\mathsf{ask},\mathsf{tk},c^*)$
10 : **When** queried $\mathsf{E.Enc}(\mathsf{pk},m,r)$:
11 : Get $c \leftarrow \mathsf{H}(\mathsf{pk},m,r)$
12 : Set $\xi^*(\mathsf{sk},c) \leftarrow m \;:\; \mathsf{pk} = \phi^*(\mathsf{sk})$
13 : **Reply** c
14 : **return** $\widetilde{m}$
15 : // Key Gen. and Decryption query
16 : **When** queried $\mathsf{E.Gen}(\lambda;\mathsf{sk})$:
17 : **Reply** $(\phi^*(\mathsf{sk}),\mathsf{sk})$
18 : **when** queried $\mathsf{E.Dec}(\mathsf{sk},c)$:
19 : **Reply** $\xi^*(\mathsf{sk},c)$

Figure 7.8: RO-Channel based on black-box Anamorphic Encryption. The notation $\mathsf{H}(\mathsf{pk},m,r) \leftarrow c$ denotes that future calls to H on (pk,m,r) return c without calling H.

Note this holds only when all queries the anamorphic encryption scheme performs to the underlying PKE are answered correctly. Our last step is then to prove $\mathcal{S},\mathcal{R}$ simulate the ideal PKE correctly. Let $\mathsf{View}^{\mathsf{real}}$ be the sequence of oracle replies $\mathsf{AT.Gen},\mathsf{AT.Enc}$, $\mathsf{AT.Dec}$ (in this order) would observe when executed with the correct PKE, and $\mathsf{View}^{\mathsf{sim}}$ the sequence of values they get with $\mathcal{S},\mathcal{R}$. We claim them to be statistically close, implying that $\Pr[\widetilde{m} \neq \widehat{m}] \leq \mathsf{negl}(\lambda)$.

Claim 7.4.1. $\Delta(\mathsf{View}^{\mathsf{real}},\mathsf{View}^{\mathsf{sim}}) \leq \mathsf{negl}(\lambda)$.

A proof of this Claim is presented in [38]. Finally, applying Theorem 7.3.1, we conclude that $|\widehat{M}| \leq 2(q_e+q_d)^2$. □

Remark 7.4.1. *Again, this lower bound holds for stateless black-box triplets. If stateful anamorphic encryption/decryption is allowed, Lemma 7.2.4 only guarantees that c is a valid ciphertext observed by* $\mathsf{AT.Enc}$ *or* $\mathsf{AT.Gen}$ *(see Remark 7.2.1). This worsen the final bound to* $\widehat{M} \leq 2(q_e+q_d+2q_g)^2$ *with* q_g *the total queries of* $\mathsf{AT.Gen}$*. The proof is readily adapted by replacing* ψ^* *with* H *calls both in* $\mathcal{S}$ *and* $\mathcal{R}$*.*

7.4.2 Impossibility of Black-Box Asymmetric AE

The bounds provided in the previous section applies to any black-box anamorphic triplet. Although our bound can be achieved asymptotically, see [96], the only known constructions encrypt anamorphic messages in a *symmetric* fashion. That is, sender and receiver must have exchanged a secret key in advance. The lack of black-box *asymmetric* anamorphic scheme is however no coincidence. In this section we will indeed prove that such constructions are impossible.

More precisely, we will prove that any black-box anamorphic triplet scheme satisfying Definition 7.2.1, must be insecure with respect to the Weak Asymmetric security notion (Definition 7.1.4) when instantiated for the ideal PKE scheme.

Theorem 7.4.2. *For any black-box anamorphic triplet* $\Sigma = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$*, when applied to the ideal PKE* $\Pi = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ *(Section 7.2.1) there exists* $\mathcal{A}$ PPT *such that,*

$$\mathsf{Adv}^{\mathsf{weak\text{-}asy\text{-}anam}}_{\mathcal{A},\Sigma}(\lambda) \;\geq\; \frac{1}{\mathsf{poly}(\lambda)}.$$

Proof. At a high level the strategy of $\mathcal{A}$, fully described in Figure 7.9, is as follows. First it gets a challenge ciphertext c^* encrypting either $(m^*, \widehat{m}_0)$ or $(m^*, \widehat{m}_1)$ random messages of its choice. Next it locally runs $\mathsf{AT.Enc}$ to encrypt $\widehat{m}_0$ and during its execution replaces the response of a randomly chosen query to $\mathsf{E.Enc}$ with c^*. If c^* encrypts $\widehat{m}_0$, $\mathsf{AT.Enc}$ should return it with significant probability, whereas if it encrypts $\widehat{m}_1$, this should only happen with negligible probability.

This simple approach however faces a number of technical challenges. First, we need to ensure $\mathcal{A}$ is unlikely to *overwrite* an encryption query that was previously performed by $\mathsf{AT.Gen}$, as this will create detectable inconsistencies. Next, the query c^* may not follow the expected distribution *given* dk. This may be the case since anamorphic security only guarantees c^* to be indistinguishable from any other ciphertext given $\mathsf{ask}, \mathsf{apk}$ but not dk. Thus c^* is *not* hard to distinguish and creates a non-negligible change in the view of $\mathsf{AT.Enc}$.

To address the first issue we rely on a preprocessing phase (Lines 1-5): $\mathcal{A}$ initially runs $\mathsf{AT.Enc}$ for ϑ many times (we fix ϑ later) and stores the randomness used in encryption queries of the form $\mathsf{E.Enc}(\mathsf{apk}, m^*; r)$. The idea is that if $\mathsf{AT.Gen}$ performs a query of this kind, either it is easily observed in the preprocessing or $\mathsf{AT.Enc}$ queries it with sufficiently low probability for our argument to go through. After this phase, the attack is executed as mentioned above (lines 6-13), choosing the query to program randomly among those of the form $\mathsf{E.Enc}(\mathsf{apk}, m^*; r)$ where r was not observed in the preprocessing phase.

Regarding the second issue, we will use the fact that $\mathsf{AT.Enc}$ can be roughly treated as a symmetric choice function (see Section 7.2.5). This will help us conclude that, when c^* is the encryption of $\widehat{m}_0$, the probability of choosing it again is significant.

Let $q = \mathsf{poly}(\lambda)$ be the number of queries made by $\mathsf{AT.Enc}$ to $\mathsf{E.Enc}$. Our first step is to show that although c^* is biased, this can only increase the probability of certain (bad) events by a factor of $\approx q$, plus a non-negligible term accounting for the probability that $\mathcal{A}$ overwrites a query previously asked by $\mathsf{AT.Gen}$. To be more precise we call Bias the

$\mathcal{A}(\mathsf{apk}, \mathsf{dk})$:

1 : // Preprocessing phase
2 : Set $R \leftarrow \varnothing$, sample $m^* \leftarrow^{\$} M$ and $\widehat{m}_0, \widehat{m}_1 \leftarrow^{\$} \widehat{M}$
3 : **for** ϑ times:
4 : Run $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m^*, \widehat{m}_0)$
5 : **When** it queries $\mathsf{E.Enc}(\mathsf{apk}, m^*; r)$: Store $R \leftarrow R \cup \{r\}$
6 : // Attack phase
7 : Sample a random $i \leftarrow^{\$} \{1, \ldots, q\}$
8 : Give $(m^*, \widehat{m}_0, \widehat{m}_1)$ to the challenger and obtain c^*
9 : Run $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m^*, \widehat{m}_0)$
10 : **When** it queries the i-th time a new $\mathsf{E.Enc}(\mathsf{apk}, m^*; r)$ with $r \notin R$:
11 : **Reply** with c^*
12 : **When** it returns c':
13 : **return** $c^* == c'$

Figure 7.9: Adversary for the Weak Asymmetric AE game, where $\vartheta = \mathsf{poly}(\lambda)$ and $q = \mathsf{poly}(\lambda)$ is the number of queries made by $\mathsf{AT.Enc}$ to $\mathsf{E.Enc}$.

joint view of $\mathsf{AT.Gen}$, which generates $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk})$, $\mathsf{AT.Enc}$ executed as in line 9, and $\mathsf{AT.Dec}(\mathsf{ask}, \mathsf{tk}, c')$. Similarly, let Real be the same view, with the exception that at line 11 $\mathcal{A}$ returns the correct ciphertext $\mathsf{E.Enc}(\mathsf{apk}, m^*; r)$. In [38] the following bound is proven.

Claim 7.4.2. *For any predicate p*

$$\Pr\left[p(\mathsf{Bias}) = 1\right] \leq q \cdot \Pr\left[p(\mathsf{Real}) = 1\right] + \frac{q^2}{\vartheta + 1} + \mathsf{negl}(\lambda).$$

Next we proceed studying the probability that $\mathcal{A}$ returns 1 when c^* is an encryption of $\widehat{m}_b$ for $b \in \{0, 1\}$ separately.

Encryption of $\widehat{m}_1$. In this case let Err be the event $\mathsf{AT.Dec}(\mathsf{ask}, \mathsf{tk}, c') \neq \widehat{m}_0$. From correctness of the anamorphic encryption scheme, if $\mathcal{A}$ replies with the correct ciphertext at line 11, this event occurs only with negligible probability. Using Claim 7.4.2 we have then that

$$\begin{aligned}\Pr\left[c' = c^* \mid b = 1\right] \;&\leq\; \Pr\left[\mathsf{Err}\right] + \mathsf{negl}(\lambda) \;\leq\; q \cdot \mathsf{negl}(\lambda) + \frac{q^2}{\vartheta + 1} + \mathsf{negl}(\lambda) \\ &= \frac{q^2}{\vartheta + 1} + \mathsf{negl}(\lambda)\end{aligned}$$

where the first inequality follows as c^* is the encryption of $\widehat{m}_1$, and therefore, up to negligible probability $\mathsf{AT.Dec}(\mathsf{ask}, \mathsf{tk}, c^*) = \widehat{m}_1 \neq \widehat{m}_0$.

Encryption of $\widehat{m}_0$. We start by fixing some notation. We will call S^*, S the sets of randomness r so that the query $\mathsf{E.Enc}(\mathsf{apk}, m^*; r)$ was respectively performed by $\mathsf{AT.Enc}$ inside the challenger call in line 8 or $\mathsf{AT.Enc}$ executed in line 9. As a direct consequence of the Ciphertext Selection Lemma and Lemma 7.2.2 we then claim that

Claim 7.4.3. *Calling* $\mathsf{BadChoice} : (\nexists r' \in S \setminus R : c' = \mathsf{E.Enc}(\mathsf{apk}, m^*; r'))$ *and analogously* $\mathsf{BadChoice}^* : (\nexists r^* \in S^* \setminus R : c^* = \mathsf{E.Enc}(\mathsf{apk}, m^*; r^*))$ *then*

$$\Pr[\mathsf{BadChoice}^*] \leq \mathsf{negl}(\lambda), \qquad \Pr[\mathsf{BadChoice}] \leq \frac{q^2}{\vartheta + 1} + \mathsf{negl}(\lambda).$$

Next, our goal is to argue that $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m^*, \widehat{m}_0)$ is *close* to a symmetric choice function, taking as input the ciphertexts it requests through encryption calls and returning one of them. Conditioning on $\neg\mathsf{BadChoice}$ guarantees that this is a choice function. To argue it is also almost symmetric we use a sequence of hybrid adversaries where we replace $\mathsf{E.Enc}$ with an actual symmetric choice function $\mathcal{F}$, described in Figure 7.10.

- $\mathcal{A}_1$: The adversary described in Figure 7.9, when the challenger encrypts $\widehat{m}_0$.
- $\mathcal{A}_2$: As $\mathcal{A}_1$, but to compute c^* it samples $c_1, \ldots, c_q \leftarrow^{\$} \{0,1\}^\ell$ and evaluates the function $\mathcal{F}$, described in Figure 7.10, setting $c^* = \mathcal{F}(c_1, \ldots, c_q)$.
- $\mathcal{A}_3$: As $\mathcal{A}_2$, but to compute c' it samples $c_2, \ldots, c_q \leftarrow^{\$} \{0,1\}^\ell$ and evaluates the function $\mathcal{F}$, described in Figure 7.10, setting $c' = \mathcal{F}(c^*, c_2, \ldots, c_q)$.

$\mathcal{F}(c_1, \ldots, c_q)$:

1 : Sample a random permutation $\pi : \{1, \ldots, q\} \to \{1, \ldots, q\}$.
2 : Run $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m^*, \widehat{m}_0)$
3 : **When** it queries a new $\mathsf{E.Enc}(\mathsf{apk}, m^*; r)$ with $r \notin R$ the i-th time:
4 : **Reply** $c_{\pi(i)}$
5 : **When** it queries $\mathsf{E.Dec}(\mathsf{ask}, c)$ with $c \in \{c_1, \ldots, c_q\}$:
6 : **Reply** m^*.
7 : **When** it returns c_{out}
8 : **if** $c_{\mathsf{out}} \in \{c_1, \ldots, c_q\}$: **return** c_{out}
9 : **else** : **return** a random $c_{\mathsf{out}} \leftarrow^{\$} \{c_1, \ldots, c_q\}$

Figure 7.10: Symmetric choice function used to replace $\mathsf{E.Enc}$ in $\mathcal{A}_1, \mathcal{A}_2$. Note this is implicitly parametrized by $\mathsf{apk}, \mathsf{dk}$ and R. Equality to ask can be checked querying $\mathsf{E.Gen}$.

For notational convenience we will call c_i^*, c_i' the ciphertexts generated by $\mathcal{A}_i$. Then we can claim that $\mathcal{F}$ is a symmetric choice function and that the statistical distance between the ciphertexts generated by these adversaries is small.

Claim 7.4.4. *$\mathcal{F}$ is a symmetric choice function (see Definition 7.2.5).*

Claim 7.4.5. $\Delta((c_1^*, c_1'), (c_2^*, c_2')) \leq \mathsf{negl}(\lambda)$.

Claim 7.4.6. $\Delta((c_2^*, c_2'), (c_3^*, c_3')) \leq \frac{2q^2}{1+\vartheta} + \mathsf{negl}(\lambda)$.

All three Claims are proven in [38]. Combining them with Lemma 7.2.5 we have that $\Pr[c_3^* = c_3'] \geq q^{-1} - \mathsf{negl}(\lambda)$ and in particular

$$\begin{aligned}\Pr[c^* = c' \mid b = 0] \;=\; \Pr[c_1^* = c_1'] \;&\geq\; \Pr[c_3^* = c_3'] - \frac{2q^2}{\vartheta + 1} - \mathsf{negl}(\lambda) \\ &\geq\; \frac{1}{q} - \frac{2q^2}{\vartheta + 1} - \mathsf{negl}(\lambda).\end{aligned}$$

Advantage Bound. Combining both intermediate results, a bound on the advantage of $\mathcal{A}$ can be derived as

$$\begin{aligned}\mathsf{Adv}(\mathcal{A}) \;&=\; |\Pr[c^* = c' \mid b = 0] - \Pr[c^* = c' \mid b = 1]| \\ &\geq\; \left(\frac{1}{q} - \frac{2q^2}{\vartheta + 1} - \mathsf{negl}(\lambda)\right) - \left(\frac{q^2}{\vartheta + 1} + \mathsf{negl}(\lambda)\right) \\ &\geq\; \frac{1}{q} - \frac{3q^2}{\vartheta + 1} - \mathsf{negl}(\lambda).\end{aligned}$$

Setting $\vartheta = 6q^3 - 1$ we get that the advantage is negligibly close to $1/2q$. As $q = \mathsf{poly}(\lambda)$ the Theorem is proven. □

Remark 7.4.2. *As done previously, the Theorem only refers to a stateless anamorphic triplet. In this case however we choose not to discuss about stateful variants as, even in anamorphic mode, the scheme is asymmetric, with potentially many senders holding the same* dk*. Thus keeping state in such case does not appear meaningful.*

Part III

Conclusion

Chapter 8

Conclusions and Future Work

In this thesis we investigated and proved negative results regarding the construction of several cryptographic primitives. These include black-box constructions of digital signatures (Chapter 3), vector commitments (Chapter 4) and NIZKs (Chapter 5) from pairing-free groups; distributed group action evaluation over a black-box hard homogeneous space (Chapter 6); anamorphic encryption over a generic black-box encryption scheme (Chapter 7).

About signature and vector commitments, when built from a black-box groups we proved in this thesis that they respectively: cannot support an exponentially large message space; cannot have succinct commitment and opening. An interesting direction is therefore to understand what can be achieved when non-black-box usage of the group, still without pairing, is allowed.

For the case of signatures, under the hardness of the discrete logarithm problem which implies one way function, non-black-box constructions exist [99]. However, known construction are inefficient in terms of signature size and signing time. Whether *short* signatures exists based only on the hardness of the DLP, DDH or related problems in pairing free groups remains a long-standing open problem.

Regarding vector commitments, again assuming only the DLP to be hard, collision resistant hash function can be derived and in particular Merkle trees [90]. However Merkle trees do not achieve constant openings size, known to be achievable from pairing groups. An interesting open question in this direction is thus whether VC with both constant commitment and opening size exists from pairing free groups.

The case of non-interactive zero-knowledge arguments is more nuanced. Our results indeed *only* exclude black-box construction for a large class of very natural relations when targeting either knowledge soundness or plain soundness. It would therefore be of theoretical interests to fill the above gap in terms of the supported relations. Specifically, it could be the case that such gap is caused by limitations in our proof techniques, and so new approaches may lead to stronger negative results. Alternatively, there may actually exist non-trivial relations not covered by our results for which NIZK built on black-box groups are possible. Finally, another interesting direction for future work would be to understand whether our negative results can be extended to argument system only satisfying witness hiding, or (more generally) relaxed notions of zero-knowledge.

Our fourth result demonstrated limitations for distributed group action evaluations. Specifically, we proved that when the group actions is used in a black-box way, the round complexity must be linear in the threshold. As opposed to the negative results in Chapters 3, 4 and 5, this one cannot be bypassed neither assuming the hardness of problems unrelated to the hard-homogeneous space at hand, nor relying on an explicit representation of set elements. This strongly suggests that further improvements in this problem, which is of great interests for post-quantum threshold schemes based on group actions, may only be obtained by optimizing the explicit circuit representation to evaluate the group action, and designing specialized MPC protocols.

Finally, regarding anamorphic encryption, the results discussed in this thesis showed that black-box constructions, which would apply to any PKE, cannot achieve both high rate and stronger security notions. Such result has initiated a line of work dedicated to further understanding the limitations of anamorphic encryption. [36] proved that the original notion of AE is in fact not achievable for black-box PKE, proposed a weaker variant, named *semi-adaptive* AE, and extended our results to this new setting. [51] realized the first *anamorphic resistant* PKE in the CRS and RO model, i.e. a encryption scheme for which any AE cannot communicate more than $O(\log \lambda)$ covert bits per ciphertext. [26] realized a stronger variant of anamorphic resistant PKE, either in the RO or CRS model, for which any AE satisfying the original security notion cannot send even a single covert bit. Lastly, [8] realized concrete anamorphic resistant encryption schemes in the CRS model (without RO), only based on the exponential hardness of DDH. Despite recent progress, many questions remain open. For instance whether the stronger variant of anamorphic resistance in [26] can be based on weaker assumption than obfuscation, or whether anamorphic resistance can be obtained in the standard model (with no CRS or RO). A more open-ended question is whether high-rate covert channels are achievable for any PKE against a weaker authority influence.

Bibliography

[1] Shahla Atapoor. Identity-based threshold signatures from isogenies. In Elizabeth A. Quaglia, editor, *19th IMA International Conference on Cryptography and Coding*, volume 14421 of *LNCS*, pages 220–240. Springer, Cham, December 2023.

[2] Shahla Atapoor, Karim Baghery, Daniele Cozzo, and Robi Pedersen. CSI-SharK: CSI-FiSh with sharing-friendly keys. In Leonie Simpson and Mir Ali Rezazadeh Baee, editors, *ACISP 23*, volume 13915 of *LNCS*, pages 471–502. Springer, Cham, July 2023.

[3] Shahla Atapoor, Karim Baghery, Daniele Cozzo, and Robi Pedersen. Practical robust DKG protocols for CSIDH. In Mehdi Tibouchi and Xiaofeng Wang, editors, *ACNS 23International Conference on Applied Cryptography and Network Security, Part II*, volume 13906 of *LNCS*, pages 219–247. Springer, Cham, June 2023.

[4] Shahla Atapoor, Karim Baghery, Daniele Cozzo, and Robi Pedersen. VSS from distributed ZK proofs and applications. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part I*, volume 14438 of *LNCS*, pages 405–440. Springer, Singapore, December 2023.

[5] Thomas Attema, Ignacio Cascudo, Ronald Cramer, Ivan Damgård, and Daniel Escudero. Vector commitments over rings and compressed Σ-protocols. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 173–202. Springer, Cham, November 2022.

[6] Thomas Attema and Ronald Cramer. Compressed Σ-protocol theory and practical application to plug & play secure algorithmics. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 513–543. Springer, Cham, August 2020.

[7] Thomas Attema, Serge Fehr, and Michael Klooß. Fiat-shamir transformation of multi-round interactive proofs. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 113–142. Springer, Cham, November 2022.

[8] Gennaro Avitabile, Vincenzo Botta, Emanuele Giunta, Marcin Mielniczuk, and Francesco Migliaro. The malice of elfs: Practical anamorphic-resistant encryption without random oracles. *Cryptology ePrint Archive*, 2025.

[9] Karim Baghery, Daniele Cozzo, and Robi Pedersen. An isogeny-based ID protocol using structured public keys. In Maura B. Paterson, editor, *18th IMA International Conference on Cryptography and Coding*, volume 13129 of *LNCS*, pages 179–197. Springer, Cham,

December 2021.

[10] Fabio Banfi, Konstantin Gegier, Martin Hirt, Ueli Maurer, and Guilherme Rito. Anamorphic encryption, revisited. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 3–32. Springer, Cham, May 2024.

[11] Stephanie Bayer and Jens Groth. Efficient zero-knowledge argument for correctness of a shuffle. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 263–280. Springer, Berlin, Heidelberg, April 2012.

[12] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In Ueli M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 399–416. Springer, Berlin, Heidelberg, May 1996.

[13] Ward Beullens, Lucas Disson, Robi Pedersen, and Frederik Vercauteren. CSI-RAShi: Distributed key generation for CSIDH. In Jung Hee Cheon and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 12th International Workshop, PQCrypto 2021*, pages 257–276. Springer, Cham, 2021.

[14] Ward Beullens, Samuel Dobson, Shuichi Katsumata, Yi-Fu Lai, and Federico Pintore. Group signatures and more from isogenies and lattices: Generic, simple, and efficient. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 95–126. Springer, Cham, May / June 2022.

[15] Ward Beullens, Shuichi Katsumata, and Federico Pintore. Calamari and Falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 464–492. Springer, Cham, December 2020.

[16] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 227–247. Springer, Cham, December 2019.

[17] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *20th ACM STOC*, pages 103–112. ACM Press, May 1988.

[18] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo random bits. In *23rd FOCS*, pages 112–117. IEEE Computer Society Press, November 1982.

[19] Dan Boneh, Benedikt Bünz, and Ben Fisch. Batching techniques for accumulators with applications to IOPs and stateless blockchains. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 561–586. Springer, Cham, August 2019.

[20] Dan Boneh, Jiaxin Guan, and Mark Zhandry. A lower bound on the length of signatures based on group actions and generic isogenies. In Carmit Hazay and Martijn Stam,

editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 507–531. Springer, Cham, April 2023.

[21] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 327–357. Springer, Berlin, Heidelberg, May 2016.

[22] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy*, pages 315–334. IEEE Computer Society Press, May 2018.

[23] Matteo Campanelli, Dario Fiore, Nicola Greco, Dimitris Kolonelos, and Luca Nizzardo. Incrementally aggregatable vector commitments and applications to verifiable decentralized storage. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 3–35. Springer, Cham, December 2020.

[24] Fabio Campos and Philipp Muth. On actively secure fine-grained access structures from isogeny assumptions. In Jung Hee Cheon and Thomas Johansson, editors, *Post-Quantum Cryptography - 13th International Workshop, PQCrypto 2022*, pages 375–398. Springer, Cham, September 2022.

[25] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *Journal of the ACM (JACM)*, 51(4):557–594, 2004.

[26] Davide Carnemolla, Dario Catalano, Emanuele Giunta, and Francesco Migliaro. Anamorphic resistant encryption: the good, the bad and the ugly. *Cryptology ePrint Archive*, 2025.

[27] Ignacio Cascudo, Daniele Cozzo, and Emanuele Giunta. Verifiable secret sharing from symmetric key cryptography with improved optimistic complexity. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part VII*, volume 15490 of *LNCS*, pages 100–128. Springer, Singapore, December 2024.

[28] Ignacio Cascudo and Emanuele Giunta. On interactive oracle proofs for boolean R1CS statements. In Ittay Eyal and Juan A. Garay, editors, *FC 2022*, volume 13411 of *LNCS*, pages 230–247. Springer, Cham, May 2022.

[29] Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 423–447. Springer, Cham, April 2023.

[30] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part III*, volume 11274 of *LNCS*, pages 395–427. Springer, Cham, December 2018.

[31] Dario Catalano and Dario Fiore. Vector commitments and their applications. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages

55–72. Springer, Berlin, Heidelberg, February / March 2013.

[32] Dario Catalano, Dario Fiore, Rosario Gennaro, and Emanuele Giunta. On the impossibility of algebraic vector commitments in pairing-free groups. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part II*, volume 13748 of *LNCS*, pages 274–299. Springer, Cham, November 2022.

[33] Dario Catalano, Dario Fiore, and Emanuele Giunta. Adaptively secure single secret leader election from DDH. In Alessia Milani and Philipp Woelfel, editors, *41st ACM PODC*, pages 430–439. ACM, July 2022.

[34] Dario Catalano, Dario Fiore, and Emanuele Giunta. Efficient and universally composable single secret leader election from pairings. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part I*, volume 13940 of *LNCS*, pages 471–499. Springer, Cham, May 2023.

[35] Dario Catalano, Emanuele Giunta, and Francesco Migliaro. Anamorphic encryption: New constructions and homomorphic realizations. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 33–62. Springer, Cham, May 2024.

[36] Dario Catalano, Emanuele Giunta, and Francesco Migliaro. Generic anamorphic encryption, revisited: New limitations and constructions. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2024. TBA.

[37] Dario Catalano, Emanuele Giunta, and Francesco Migliaro. Generic anamorphic encryption, revisited: New limitations and constructions. Cryptology ePrint Archive, Report 2024/1119, 2024.

[38] Dario Catalano, Emanuele Giunta, and Francesco Migliaro. Limits of black-box anamorphic encryption. In *Annual International Cryptology Conference*. Springer, 2024.

[39] David Chaum and Torben P. Pedersen. Wallet databases with observers. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 89–105. Springer, Berlin, Heidelberg, August 1993.

[40] Geoffroy Couteau and Dominik Hartmann. Shorter non-interactive zero-knowledge arguments and ZAPs for algebraic languages. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 768–798. Springer, Cham, August 2020.

[41] Jean-Marc Couveignes. Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291, 2006.

[42] Daniele Cozzo and Emanuele Giunta. Round-robin is optimal: Lower bounds for group action based protocols. In Guy N. Rothblum and Hoeteck Wee, editors, *TCC 2023, Part IV*, volume 14372 of *LNCS*, pages 310–335. Springer, Cham, November / December 2023.

[43] Daniele Cozzo and Nigel P. Smart. Sashimi: Cutting up CSI-FiSh secret keys to produce

an actively secure distributed signing protocol. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020*, pages 169–186. Springer, Cham, 2020.

[44] Ronald Cramer. Modular design of secure yet practical cryptographic protocols. *Ph. D.-thesis, CWI and Uni. of Amsterdam*, 1996.

[45] Ronald Cramer and Victor Shoup. Signature schemes based on the strong rsa assumption. *ACM Transactions on Information and System Security (TISSEC)*, 3(3):161–185, 2000.

[46] Quang Dao and Paul Grubbs. Spartan and bulletproofs are simulation-extractable (for free!). In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part II*, volume 14005 of *LNCS*, pages 531–562. Springer, Cham, April 2023.

[47] Luca De Feo and Steven D. Galbraith. SeaSign: Compact isogeny signatures from class group actions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 759–789. Springer, Cham, May 2019.

[48] Luca De Feo and Michael Meyer. Threshold schemes from isogeny assumptions. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 187–212. Springer, Cham, May 2020.

[49] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[50] Yevgeniy Dodis and Eli Goldin. Private communication, 2024.

[51] Yevgeniy Dodis and Eli Goldin. Anamorphic-resistant encryption; or why the encryption debate is still alive. *Cryptology ePrint Archive*, 2025.

[52] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 523–540. Springer, Berlin, Heidelberg, May 2004.

[53] Nico Döttling, Dominik Hartmann, Dennis Hofheinz, Eike Kiltz, Sven Schäge, and Bogdan Ursu. On the impossibility of purely algebraic signatures. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part III*, volume 13044 of *LNCS*, pages 317–349. Springer, Cham, November 2021.

[54] Julien Duman, Dominik Hartmann, Eike Kiltz, Sabrina Kunzweiler, Jonas Lehmann, and Doreen Riepel. Generic models for group actions. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part I*, volume 13940 of *LNCS*, pages 406–435. Springer, Cham, May 2023.

[55] Ali El Kaafarani, Shuichi Katsumata, and Federico Pintore. Lossy CSI-FiSh: Efficient signature scheme with tight reduction to decisional CSIDH-512. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 157–186. Springer, Cham, May 2020.

[56] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification

and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Berlin, Heidelberg, August 1987.

[57] Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 33–62. Springer, Cham, August 2018.

[58] Chaya Ganesh, Claudio Orlandi, Mahak Pancholi, Akira Takahashi, and Daniel Tschudi. Fiat-shamir bulletproofs are non-malleable (in the algebraic group model). In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 397–426. Springer, Cham, May / June 2022.

[59] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009.

[60] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 99–108. ACM Press, June 2011.

[61] Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The relationship between public key encryption and oblivious transfer. In *41st FOCS*, pages 325–335. IEEE Computer Society Press, November 2000.

[62] Yael Gertner, Tal Malkin, and Omer Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates. In *42nd FOCS*, pages 126–135. IEEE Computer Society Press, October 2001.

[63] Emanuele Giunta. On the impossibility of algebraic NIZK in pairing-free groups. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part IV*, volume 14084 of *LNCS*, pages 702–730. Springer, Cham, August 2023.

[64] Emanuele Giunta and Alistair Stewart. Unbiasable verifiable random functions. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part IV*, volume 14654 of *LNCS*, pages 142–167. Springer, Cham, May 2024.

[65] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM (JACM)*, 33(4):792–807, 1986.

[66] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *21st ACM STOC*, pages 25–32. ACM Press, May 1989.

[67] Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *14th ACM STOC*, pages 365–377. ACM Press, May 1982.

[68] Shafi Goldwasser, Silvio Micali, and Ronald L Rivest. A digital signature scheme secure against adaptive chosen-message attacks. volume 17, pages 281–308. SIAM, 1988.

[69] Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 444–459. Springer, Berlin, Heidelberg, December 2006.

[70] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for NIZK. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 97–111. Springer, Berlin, Heidelberg, August 2006.

[71] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 339–358. Springer, Berlin, Heidelberg, May / June 2006.

[72] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Berlin, Heidelberg, April 2008.

[73] Johan Håstad, Russell Impagliazzo, Leonid A Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[74] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *21st ACM STOC*, pages 44–61. ACM Press, May 1989.

[75] Abhishek Jain and Zhengzhong Jin. Non-interactive zero knowledge from sub-exponential DDH. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 3–32. Springer, Cham, October 2021.

[76] Jeff Kahn, Michael Saks, and Cliff Smyth. A dual version of reimer's inequality and a proof of rudich's conjecture. In *Proceedings 15th Annual IEEE Conference on Computational Complexity*, pages 98–103. IEEE, 2000.

[77] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 177–194. Springer, Berlin, Heidelberg, December 2010.

[78] Miroslaw Kutylowski, Giuseppe Persiano, Duong Hieu Phan, Moti Yung, and Marcin Zawada. The self-anti-censorship nature of encryption: On the prevalence of anamorphic cryptography. *Proceedings on Privacy Enhancing Technologies*, 2023(4):170–183, 2023.

[79] Russell W. F. Lai and Giulio Malavolta. Subvector commitments with application to succinct arguments. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 530–560. Springer, Cham, August 2019.

[80] Yi-Fu Lai. CAPYBARA and TSUBAKI: Verifiable random functions from group actions and isogenies. Cryptology ePrint Archive, Report 2023/182, 2023.

[81] Yi-Fu Lai, Steven D. Galbraith, and Cyprien Delpech de Saint Guilhem. Compact, efficient and UC-secure isogeny-based oblivious transfer. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 213–241. Springer, Cham, October 2021.

[82] Leslie Lamport. Constructing digital signatures from a one-way function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, October 1979.

[83] Benoît Libert, Somindu C. Ramanna, and Moti Yung. Functional commitment schemes: From polynomial commitments to pairing-based accumulators from simple assumptions. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *ICALP 2016*, volume 55 of *LIPIcs*, pages 30:1–30:14. Schloss Dagstuhl, July 2016.

[84] Benoît Libert and Moti Yung. Concise mercurial vector commitments and independent zero-knowledge sets with short proofs. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 499–517. Springer, Berlin, Heidelberg, February 2010.

[85] Douglas L. Long and Avi Wigderson. How discreet is the discrete log? In *15th ACM STOC*, pages 413–420. ACM Press, April 1983.

[86] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 448–471. Springer, Cham, April 2023.

[87] Ueli Maurer, Christopher Portmann, and Jiamin Zhu. Unifying generic group models. Cryptology ePrint Archive, Report 2020/996, 2020.

[88] Ueli M. Maurer. Abstract models of computation in cryptography (invited paper). In Nigel P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *LNCS*, pages 1–12. Springer, Berlin, Heidelberg, December 2005.

[89] Ueli M. Maurer. Unifying zero-knowledge proofs of knowledge. In Bart Preneel, editor, *AFRICACRYPT 09*, volume 5580 of *LNCS*, pages 272–286. Springer, Berlin, Heidelberg, June 2009.

[90] Ralph C. Merkle. A digital signature based on a conventional encryption function. In Carl Pomerance, editor, *CRYPTO'87*, volume 293 of *LNCS*, pages 369–378. Springer, Berlin, Heidelberg, August 1988.

[91] Tomoki Moriya, Hiroshi Onuki, and Tsuyoshi Takagi. SiGamal: A supersingular isogeny-based PKE and its application to a PRF. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 551–580. Springer, Cham, December 2020.

[92] Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, January 1991.

[93] Periklis A. Papakonstantinou, Charles Rackoff, and Yevgeniy Vahlis. How powerful are the DDH hard groups? *Electron. Colloquium Comput. Complex.*, page 167, 2012.

[94] Charalampos Papamanthou, Elaine Shi, Roberto Tamassia, and Ke Yi. Streaming authenticated data structures. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 353–370. Springer, Berlin, Heidelberg, May 2013.

[95] Chris Peikert, Zachary Pepin, and Chad Sharp. Vector and functional commitments from lattices. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part III*, volume

13044 of *LNCS*, pages 480–511. Springer, Cham, November 2021.

[96] Giuseppe Persiano, Duong Hieu Phan, and Moti Yung. Anamorphic encryption: Private communication against a dictator. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 34–63. Springer, Cham, May / June 2022.

[97] Oded Regev. An efficient quantum factoring algorithm. *arXiv preprint arXiv:2308.06572*, 2023.

[98] Damien Robert. Breaking SIDH in polynomial time. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 472–503. Springer, Cham, April 2023.

[99] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd ACM STOC*, pages 387–394. ACM Press, May 1990.

[100] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252. Springer, New York, August 1990.

[101] Surbhi Shaw and Ratna Dutta. Identification scheme and forward-secure signature in identity-based setting from isogenies. In *International Conference on Provable Security*, pages 309–326. Springer, 2021.

[102] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th FOCS*, pages 124–134. IEEE Computer Society Press, November 1994.

[103] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Berlin, Heidelberg, May 1997.

[104] Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 334–345. Springer, Berlin, Heidelberg, May / June 1998.

[105] Anton Stolbunov. Cryptographic schemes based on isogenies. 2012.

[106] Erkan Tairi, Pedro Moreno-Sanchez, and Matteo Maffei. Post-quantum adaptor signature for privacy-preserving off-chain payments. In Nikita Borisov and Claudia Díaz, editors, *FC 2021, Part II*, volume 12675 of *LNCS*, pages 131–150. Springer, Berlin, Heidelberg, March 2021.

[107] Yi Wang, Rongmao Chen, Xinyi Huang, and Moti Yung. Sender-anamorphic encryption reformulated: Achieving robust and generic constructions. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part VI*, volume 14443 of *LNCS*, pages 135–167. Springer, Singapore, December 2023.

[108] Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer, Berlin, Heidelberg, May 2005.

[109] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th FOCS*, pages 162–167. IEEE Computer Society Press, October 1986.

[110] Andrew Chi-Chih Yao. Lower bounds for algebraic computation trees with integer inputs. In *30th FOCS*, pages 308–313. IEEE Computer Society Press, October / November 1989.

[111] Mark Zhandry and Cong Zhang. Indifferentiability for public key cryptosystems. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 63–93. Springer, Cham, August 2020.