INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Polynomial Reconstruction Based Cryptography

by

Aggelos Kiayias

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

2002

This manuscript has been read and accepted for the Graduate Faculty in Computer Science in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

9/20/2002
Date

Stathis Zachos
Chair of Examining Committee

9/20/02
Date

Ted Brown
Executive Officer

Supervisory Committee
Victor Pan, Lehman College
Rohit Parikh, Brooklyn College
Moti Yung, Columbia University

THE CITY UNIVERSITY OF NEW YORK

Abstract

Polynomial Reconstruction Based Cryptography

by

Aggelos Kiayias

Advisers: Moti Yung and Stathis Zachos

An important direction in Cryptographic Research is the identification and investigation of hard computational problems upon which the security of cryptographic primitives and protocols can be based. This is important because: (i) the properties of a certain cryptographic primitive depend to a great extent on the properties of the underlying hard computational problem that is used as an intractability assumption. Employing new plausible intractability assumptions with rich algebraic structure would presumably give rise to cryptographic primitives with unique and novel properties; (ii) the listing of "hard" computational problems that has been utilized in the design of cryptographic primitives is not very large; this motivates the investigation of novel cryptographic assumptions, as different problems will most likely react differently in stronger adversarial models that become feasible with technical/technological advances.

In this thesis, the Problem of Reed-Solomon Codes Decoding - a well-known computational problem in the theory of Error-Correcting Codes, also known as Polynomial Reconstruction (PR) - is considered from the cryptographic hardness perspective. Following the standard methodology in the cryptographic literature which includes the investigation of a Decisional Intractability Assumption related to PR as well as the establishment of results such as Hardness of Partial Information Extraction and

Pseudorandomness, we lay out the theoretical framework for the exploitation of PR in Cryptography.

Subsequently, we present a wide array of concrete applications of PR in Cryptography: (i) basic primitives, such as a probabilistic one-way function with strong concealment properties which gives rise to commitment schemes with unique properties; (ii) symmetric encryption methods with strong security properties such as forward secrecy, computational perfect secrecy, and key-equivalence; (iii) a generic protocol for efficient secure function evaluation over the domain of polynomial expressions. We also consider applications of our work in the Coding Theoretic setting: we investigate the solvability of Interleaved Reed-Solomon Codes Decoding in the presence of random errors, and we discover interesting hardness/self-reducibility tradeoffs in Decoding Problems (including the PR problem). Finally we describe directions for future research that are spawned from the results of this thesis.

Acknowledgements

First, I would like to thank my two advisors, Moti Yung and Stathis Zachos.

I would like to thank my advisor Stathis Zachos for his thoughtful guidance and support throughout the years of my graduate studies. I am grateful to Stathis for transforming me from a Mathematician who liked computers to a *Computer Scientist.* His teaching and research style and methodology had a great impact on my scientific development and I will be proud to follow his example in my career. Stathis has also been a good friend and we shared a lot in the years I worked under his supervision. I will always admire his courage, enthusiasm and dedication to teaching and I hope that in my academic career I will have the opportunity to "return the favor" by serving other students the way he served me. Finally, I am indebted to Stathis for encouraging me to come to New York following my drive to do research in Cryptography, and for directing me to work with Moti Yung.

I would like to thank my advisor Moti Yung for introducing me to cryptographic research. His influence in the three years we collaborated has been immense and we co-author all the publications that result from this thesis' research. Moti is a brilliant mentor, who shaped and matured my perspective in all aspects of Computer Science research through his unique guidance (always challenging, never spoonfeeding). I consider my collaboration with Moti as the undoubted highest peak of my scientific development. On top of all that, Moti is the greatest person to work with; his sense of humor along with his rich ideas and culture lead to frequent digressions from technicalities and to long discussions about the true nature of things. I will be indebted to Moti not only for sharing with me his vast experience, brilliant ideas and research directions, but also for providing me with a unique example of a true *Scientist*, philosophical but also technical, profound thinker but also practical and productive.

Additionally I would like to express my gratitude to Rohit Parikh, for being a brilliant teacher who enlightened my graduate studies and for serving in my defense committee. I am indebted to Ted Brown for his assistance in many issues and his support. A special thanks is due to my dear friend Joe Driscoll whose help in many administrative issues (and quite frequently help that went much beyond his administrative duties) proved more than crucial; our conversations were very uplifting at difficult times. I am indebted to Victor Pan for serving in my dissertation committee and to Melvin Fitting for being a patient listener and for helping me out with several issues.

Regarding the results of this thesis, I would like to thank Daniel Bleichenbacher for sharing his insights about some aspects of this work and additionally I am thankful to Alexander Barg and Yevgeniy Dodis for helpful discussions.

I also thank Aris Pagourtzis, my first research partner in Greece and dear friend. Our collaboration and time together had always being a joy for me. I would also like to thank all the graduate students and researchers I met during my studies and I had great time with, including Christodoulos, Chih, Euripides, Katerina, Antonis, Costas, John, Simina.

I would like to express my gratitude to my mother Heleni for the support, courage and determination she gave me, and to my father Yiannis for his support and inspiration he provided. Finally, I thank Georgia for her support and all the good times we had together and last but not least thanks to Gorky and Mpempi for being what they are (that is an *aratinga jandaya* and an *aratinga erythrogenys* with unique personalities).

New York, **Aggelos Kiayias**

2002

Contents

List of Figures

Chapter 1

Introduction

1.1 Literary Preface

Cryptography is the science of secret communications. From ancient times people tried to find ways of communicating securely. In the military setting such ability proved to be crucial in many occasions. Herodotus describes a legend dated in 480BC, when the Persians were ready to start a military expedition against Greece. At that time, an exostracised Greek living in Susa named Demaratus, decided to send a message warning the homeland of the upcoming invasion. The problem that he faced was how to write the message in such a way so that it would be undetected by the Persian guards. Herodotus writes:

> The Lacedaemonians learned first the preparations of Xerxes against Greece [...]. And they learned it in a strange way [...] The time that Xerxes decided to do the expedition against Greece, Demaratus who was in Susa, learned about it and decided to contact the Lacedaemonians. He had no other way to contact them, because of the danger that he would be caught. And he devised the following: he took a pair of folding tablets and scraped the wax off them, and then, he carved the decision of the king in the wooden surface; he did this and then he melted wax over the letters so that the

> tablets appeared to be unused; so the one who would transfer them would not have trouble with the guards. When the tablets reached Lacedaemon, the Lacedaemonians understood nothing, until in the end, as they have described it to me, Gorgo, Cleomenes' daughter, alone she divined and gave them the advice; she told them to scrape the wax and that they will find a message in the wooden surface underneath. They heard her, and the message was revealed and read, and afterwards it was communicated to the other Greeks. And so they say it happened. [Herodotus, Histories, Book 7 - Polymnia, Chapter 239; Greek to English translation by A. Kiayias]

Demaratus' technique is ingenious in its simplicity and it captures in a natural way the meaning of what is a "secure envelope", a basic cryptographic primitive. Note that the steganographic property in Demaratus' technique is essential for its security: the "envelope" itself is inconspicuous: it does not raise any suspicion.

An important issue raised in the above story is something crucial in traditional/ symmetric key Cryptography: the two parties that communicate should share some information that is hard to guess by other parties. In this particular case the secret information is the method of concealment: the wax covers the message in the tablets. The fact that it is difficult for third parties who encounter the tablets to guess the method of concealment is part of the success of the above secure communication mechanism in that occasion. Interestingly in the above situation the two parties do not share this information. This defect puts the receiver in (almost) the same position as the other third parties who observed the tablets: Cleomenes had no idea of the purpose of the tablets. It was by divine intervention[1] (through Gorgo) that Cleomenes learned the crucial step that they had to scrape the wax from the tablets. This, according to the legend, revealed the invaluable piece of information that would allow Greeks to strip the element of surprise from the strategic plan of Xerxes and

[1] As a sidenote, I remark that the role of the divine is up to interpretation in Gorgo's discovery (who was a very clever and opinionated woman). In my exposition, I will adopt divine intervention as the explanation for the unexpected discovery.

ultimately defeat the Persians.

It is clear from the above that symmetric Cryptography would require two channels of communication between the sender and the receiver: a "dangerous", or "insecure" channel over which the actual message is sent and a safe, untapable one that is used for the exchange of the concealment method. In Demaratus' case the safe channel was actually implemented by the Gods (alternatively, for agnostic readers, it could also have been the well above average IQ of Gorgo). Typically the safe channel is very "expensive" to use and the information to be exchanged has to be *short* in length. On the other hand, the insecure channel is "cheap" and can be used for long messages or repeatedly. This is nicely illustrated in the above legend: the insecure channel is relatively cheap: it merely requires Demaratus to ship the tablets through some means of transportation to Greece. The safe channel (implemented by the divine) which transmits the key to invert the concealment method is expensive to use: Greek Gods are notorious for being unpredictable and egoistic. They might enthusiastically help on one occasion (even by fighting on the mortals' side and risking their pride), go against their followers if they get insulted for some reason or even worse: simply ignore their followers' pleas and let the mortals do their business any way they know best. In this occasion though, valuing the ingenuity and cunningness of Demaratus they gladly interfere and pass the concealment information to the receivers.

An important observation at this point is that even if it is assumed that there is a secure channel that can be used for the distribution of the concealment information, this is not sufficient to ensure secure communication: ingenuity is required to use the insecure channel in a safe manner. As the Greek proverb puts it: "Συν Αθηνά και χείρα κίνει" freely translated to: "Even with God's (Athena's) help, you have to put some of your own personal effort into something." It is not enough that Gods wish to help Demaratus (to implement the secure channel), he found himself a cunning way of using the insecure channel to send the secret message.

So Cryptography is the "art" of designing concealment methods. Although as an art form it is enchanting, in order to apply cryptographic methods with relative confidence what we need is in fact a *Science*. Even though everything that man designs with thoughtfulness might be considered an art expression what we need here is a solid foundation for cryptographic applications especially in the present times where Internet applications mandate large-scale cryptographic usage.

Let us scrutinize carefully the basic concealment paradigm that was illustrated above. The concealment method employed by the sender is in fact an *algorithm*[2], which can be called the encryption algorithm. The technique used by the receiver to unveil the hidden message is also an algorithm that can be called the decryption algorithm. Clearly the natural context in which Cryptography can be formalized is Theoretical Computer Science. It should be clear that the encryption and decryption algorithms as employed by the sender and the receiver should be *efficient* in the sense that they should not require a large number of steps to be completed. Measuring the efficiency of algorithms has been a major endeavor of Computer Scientists since the 60's and there is a wide variety of tools available even to a novice student that enable them to categorize algorithms depending on their efficiency. Formalizing the efficiency of cryptographic algorithms for the sender and the receiver would not require much ingenuity.

There is another party though, that is actively involved in our setting: the guards in the case of Demaratus; the reason why the insecure channel got its name in the first place! They are the enemies, or the *adversaries* as they are called in standard cryptographic terms. Reading the concealed message should be *hard* for the adversaries. In other words, there should be no efficient algorithm that can be used by an adversary to defeat the concealment method. While Demaratus' success relies on the fact that the tablets do not raise any suspicion, it is much more practical to con-

[2] An explicitly described sequence of steps that operate on a given object in finite time.

sider the stronger adversarial setting where the guards are in fact aware that secure communication is taking place and on top of that they are aware of the encryption and decryption algorithms (as they might be in public-domain). There is a piece of information though that they should not know — the key used by the encryption algorithm and is shared by the sender and the receiver. As a result, one should be able to argue that it is hard for the adversaries to read the encrypted messages because they lack knowledge of the shared private information. In this direction one finds a major hurdle: in Computer Science, proving the hardness of problems is a notoriously hard research area. In fact the great majority of computational problems lacks a non-trivial *lower bound proof* of the amount of resources (usually time and memory) required in order to be solved by some algorithm.

Does the above mean that the hope for concrete mathematical foundation of Cryptography is lost? The answer is no, provided that we compromise for the next best thing that we can reach. Computer Science provides a powerful tool to compare two computational problems from a hardness perspective: *reductions*. Given two computational problems A and B, a reduction of A to B involves the design of a procedure that is capable of solving A using an algorithm that solves B as a subroutine. Provided that this procedure is relatively efficient, we can claim as a consequence of the reduction that problem B cannot be substantially easier to solve than problem A (taking into account the resources required by the reduction procedure).

In this spirit, even though we are not able to prove that there are no algorithms that can be used by an adversary to compromise a concealment method, we can provide a reduction of some *well-known* "hard" computational problem to the computational problem faced by the adversary. "Hard" in this case does not stand for a mathematically provable property but rather to a popular belief held among experts. A computational problem would qualify as "hard" if substantial effort has been spent in providing an efficient algorithm for it and yielded no significant results. Of course

this narrows the design methodology and the resulting properties of the concealment method: *the design, efficiency and special properties of the encryption and decryption algorithms depend to a great extent on the selected computational problem to be used in order to claim hardness.*

Studying Computational Problems from a cryptographic hardness perspective, i.e. whether they can be used to claim hardness in concealment methods, is an important area of cryptographic research. This is crucial for two main reasons:

- The properties of a certain cryptographic primitive depend to a great extent on the properties of the underlying hard computational problem that is used as an intractability assumption. Employing new plausible intractability assumptions with rich algebraic structure would presumably give rise to cryptographic primitives with unique and novel properties.

- The listing of "hard" computational problems that has been utilized in the design of cryptographic applications is not very large. This is mainly due to the fact that well studied hardness concepts in computational complexity such as NP-Hardness do not capture the breadth of properties a candidate "hard" problem should satisfy in order to be useful from a cryptographic viewpoint. This motivates further the investigation of novel cryptographic assumptions, as different problems will most likely react differently in stronger adversarial models that become feasible with technical/technological advances.

This brings us to the topic of this thesis, to be explained informally in the next section.

1.2 The Problem Addressed in this Thesis

In this work our goal is to investigate the possibility of cryptographic primitives whose security is based on the problem of *Polynomial Reconstruction* (PR). The problem of Polynomial Reconstruction is defined informally as follows:

> Given n pairs of points over a (large) finite field $\mathbb{F}$, such that at least t of them belong to the graph of a polynomial p of degree less than k, recover such a polynomial.

A graphical representation of Polynomial Reconstruction is presented in figure 1.1.

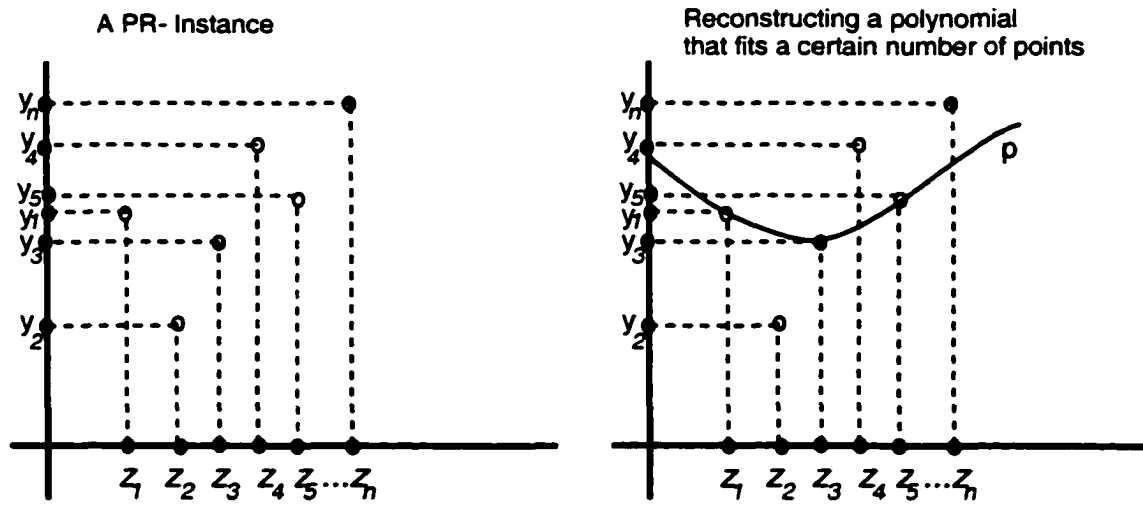

Figure 1.1: An instance of the Polynomial Reconstruction Problem and one possible solution.

Below we summarize the contributions of this thesis:

- **Framework.** We introduce the necessary framework that is required for the exploitation of Polynomial Reconstruction in Cryptography. This includes a

Decisional Intractability Assumption that is related to Polynomial Reconstruction (termed Decisional Polynomial Reconstruction Assumption — DPR). Also we consider a generalization of the problem, called Multisample Polynomial Reconstruction, that is useful in the cryptographic setting.

- **Cryptographic Properties of PR.** We show various results about PR instances that are based on the Decisional Polynomial Reconstruction Assumption. In particular we show that PR-instances, for choices of the parameters where DPR holds, hide their solution in a cryptographically strong manner.

- **Cryptographic Primitives based on PR.** We introduce, design and prove basic facts about fundamental cryptographic primitives that are based on Polynomial Reconstruction such as a probabilistic one-way function with strong security properties, and a PR-based commitment scheme.

- **Symmetric Encryption.** We introduce a semantically secure private-key cryptosystem that possesses unique security properties such as forward secrecy, computational perfect secrecy, super-polynomially large messages in the security parameter etc.

- **Secure Function Evaluation.** We introduce a novel protocol construction which solves a general family of two-player "games" that correspond to a subclass of secure function evaluation protocols where the function is selected to be a jointly constructed polynomial expression. Our protocol has many applications in games where players interact obliviously.

- **Relations to Coding.** We present a novel decoding technique for interleaved Reed-Solomon Codes, motivated from the Multisample Polynomial Reconstruction Problem. Also we investigate random self-reducibility properties of the

decoding problems for general linear and Reed-Solomon codes, and we discover interesting self-reducibility/hardness tradeoffs.

- **Public-Key Methods.** As a part of future directions for research we introduce a framework for building a key-exchange protocol based on Polynomial Reconstruction. Key-exchange protocols enable two parties who share no secret information a-priori, to exchange a common secret-key so that no eavesdropper is capable of extracting partial information about the output of the two parties given the public protocol transcript.

Some of the results of this thesis appeared in [KY01b, KY01c, KY02], while others are currently under submission.

1.3 Informal Introduction to PR-Based Cryptography

Polynomial Reconstruction debuted as an intractability assumption in the context of protocol design by Naor and Pinkas [NP99] and this work motivated further investigation of Polynomial Reconstruction from a cryptographic hardness viewpoint.

Polynomial reconstruction is essentially the problem of decoding Reed-Solomon Codes (which constitute a very successful family of codes used in settings such as CD readers and satellite communications) and as a result has received a lot of attention from a "positive" perspective (how to solve it efficiently).

In the coding theoretic setting, we have two parties, the sender and the receiver. The goal is to establish error-free communication when the channel used for the transmission of data is not error-free. The strategy used is introducing some redundancy in the data that will help detect or preferably correct errors introduced during transmission. The preparation of data to be send is referred to as the encoding process

whereas the removal of errors when data are received is referred to as the decoding process. Reed-Solomon codes can be described as follows: the sender encodes his message as k elements of a finite field and defines a polynomial of degree less than k using his message as the coefficients. Subsequently the sender transmits $n > k$ points in the graph of this polynomial. Basic facts about polynomials suggest that it is enough for k out of the n points to be transmitted without errors; in this case the original message is still recoverable. Nevertheless the problem of the decoding process is more complicated as it has to distinguish the "good" points that belong in the polynomial solution from the points that were altered due to the introduced noise during the data transmission.

As we will see in more detail later depending on the number of "good" points received (denoted by t) the problem of Polynomial Reconstruction accepts straightforward solutions when t is very close to n (few errors), and then it becomes more and more challenging when t becomes smaller and smaller.

This is of interest in the cryptographic setting. The basic idea of employing Polynomial Reconstruction in Cryptography is the following: select some message in more or less the same way as in the encoding procedure in the Coding Theoretic setting. Then, select a certain number of points in the graph of this polynomial and engulf them randomly with random points over the finite field. This is illustrated in figure 1.2

By selecting the parameters carefully the assumption will be that it would be impossible for any practical (polynomial-time bounded) observer to understand what is the encoded message (due to the combinatorial explosion of the number of possibilities). As we will see, the current state of the art presents explicit bounds and relatively convincing numerical values for the number of "good" points for which the PR-problem is solvable, something that allows us to proceed cautiously but with confidence in the cryptographic exploitation of this computational problem.

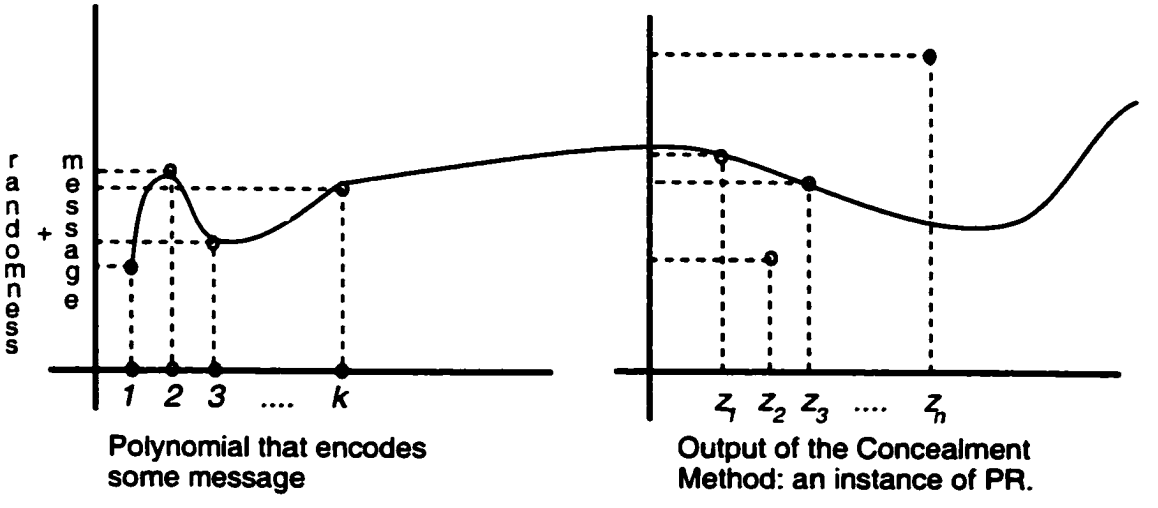

Figure 1.2: Using PR in Cryptography: concealment of a message

Chapter 2

Background

The typical model that is used in computer science to materialize the concept of an algorithm is the Turing Machine (TM). A Turing Machine involves an input tape, some work tapes, and also an output tape. Every Turing Machine defines a (so called computable) function $\mathcal{A}$ so that $\mathcal{A}(x)$ is the string found in the output-tape of the TM if the TM runs on input x. We assume that the alphabet used throughout for input and output is $\{0,1\}$ (tapes may carry additional symbols). If it holds that the range of $\mathcal{A}$ is $\{0,1\}$ we will refer to $\mathcal{A}$ as a *predicate*. The time and space complexity of a TM for a certain input size n are defined respectively as the maximum number of steps performed and the maximum number of cells written in the work tapes given an input of size n. A computable function $\mathcal{A}$ will be called polynomial-time if its corresponding TM has polynomial in n time complexity. Background on TMs can be found in any computer science theory text e.g. [Pap94, Sip97]. Background on probabilistic Turing Machines and their properties can be found in [Zac82].

Frequently, we assume additionally that the TM has access to a coin tossing device. In this case we are dealing with a Probabilistic TM. Without loss of generality we can assume that for any input of size n the number of coin tosses performed by the TM is the same. A probabilistic TM defines a function $\mathcal{A}$ of two arguments one of which is the input and the other is the coin tosses that came up in the course of the

computation. We will use the notation PPT to stand for a "probabilistic polynomial-time." In general, for convenience and whenever we feel that this is not confusing, we will refer to $\mathcal{A}$ both as a function and as the corresponding (probabilistic) Turing Machine.

Let $n \in \mathbb{N}$ be a parameter. Let D_n be a set of objects so that there is an encoding $f : D_n \to \{0,1\}^{p(n)}$ for some polynomial p. Through such encodings we will assume without loss of generality that functions defined through Turing Machines can be extended to include functions with alternative domains and ranges that do not contain bitstrings. As a convention we will assume that D_n contains objects of size n.

For any PPT $\mathcal{A}$ with input in the set D_n and output in the set R_n there is a polynomial q so that for any input $x \in D_n$, it holds that $\mathcal{A}$ performs $q(n)$ coin tosses. If $y \in R_n$ (and y might be also a function of x) we define the probability that $\mathcal{A}$ returns y for any input of size n to be

$$\mathbf{Prob}_{r \in_U \{0,1\}^{q(n)}, x \in_U D_n}[\mathcal{A}(r,x) = y] := \frac{\#\{\langle r,x\rangle \mid \mathcal{A}(r,x) = y\}}{2^{q(n)} \cdot \#D_n}$$

the notation $\#D_n$ denotes the number of elements of the set D_n.

A probability distribution $\mathcal{D}$ over some domain D is an assignment of a real value $\mathbf{Prob}_{\mathcal{D}}[x]$ between 0 and 1 (inclusively) to each element $x \in D$, with the property that $\sum_{x \in D} \mathbf{Prob}_{\mathcal{D}}[x] = 1$. Every TM $\mathcal{A}$ defines a probability distribution $\mathcal{D}_{\mathcal{A}}$ over its range R_n that is defined by $\mathbf{Prob}_{\mathcal{D}_{\mathcal{A}}}[y] := \frac{\#\{x \mid \mathcal{A}(x)=y; x \in D_n\}}{\#D_n}$. A probability distribution that can be defined through some TM will be called *samplable*. If additionally the TM is polynomial-time then we call the probability distribution *polynomial-time samplable*. Without loss of generality, since we deal with polynomial-time TMs, we will use the term samplable throughout. The uniform distribution over some domain D is a probability distribution U so that $\mathbf{Prob}_U[x] = 1/\#D$ for all $x \in D$. If x is a random variable distributed according to some distribution $\mathcal{D}$ over the domain D we will use

the notation $x \in_D D$.

A function $\alpha(n) : \mathbb{N} \to \mathbb{R}$ is negligible if for all $c \in \mathbb{N}$ it holds that $\alpha(n) < n^{-c}$ for sufficiently large n. A function $\beta(n) : \mathbb{N} \to \mathbb{R}$ is called "non-negligible" if it is not negligible for all large enough inputs, namely there is a c s.t. $\beta(n) \geq n^c$ for all n sufficiently large. When the probability of an event is greater equal to $1 - \epsilon(n)$ where $\epsilon(n)$ is negligible, then we write that the event happens "with overwhelming probability." The reader is referred to [Lub96, Gol98] for further reference and definitions of probabilistic TMs and basic probability facts.

We denote by $\mathbb{F}$ a finite field of prime order. We will use the notation $|\mathbb{F}|$ to denote the number of elements of the field. We use bold-face to denote a vector in $\mathbf{x} \in \mathbb{F}^n$ where $(\mathbf{x})_i$ denotes the i-th component of $\mathbf{x}$.

2.1 Indistinguishability of Families of Sets

Let $\{\mathcal{F}_n\}_{n \in \mathbf{N}}$ be a family of sets, such that $\mathcal{F}_n$ contains *all* possible choices of elements of size n. Also assume that the uniform distribution over $\mathcal{F}_n$ is samplable. The following definition is a rendering of the standard indistinguishability definition in the context of families of sets.

Definition 1 *Two families of sets with $A_n, B_n \subseteq \mathcal{F}_n$ are* (polynomial-time, computationally) indistinguishable *if for any* PPT *predicate $\mathcal{A}$.*

$$| \mathbf{Prob}_{X \in_U A_n; r \in_U \mathcal{R}}[\mathcal{A}(r, X) = 1] - \mathbf{Prob}_{X \in_U B_n; r \in_U \mathcal{R}}[\mathcal{A}(r, X) = 1] |$$

is negligible in n.

If on the other hand there is an $\mathcal{A}$ for which the probability above is non-negligible in n, we will say that $\mathcal{A}$ is a distinguisher *for A_n, B_n.*

Pseudorandomness is defined in a straightforward manner using indistinguishability:

Definition 2 *A family of sets A_n is called* pseudorandom *if it is indistinguishable from $\mathcal{F}_n$.*

The notion of indistinguishability is related to the "statistical closeness" or statistical indistinguishability between probability distributions defined as follows:

Definition 3 *Two probability distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ over D_n are called statistically close if the following function in n:*

$$\frac{1}{2}\sum_{x\in D_n} \mid \mathbf{Prob}_{\mathcal{D}_1}[x] - \mathbf{Prob}_{\mathcal{D}_2}[x] \mid$$

is negligible in n.

We note here that it is easy to see that statistical indistinguishability implies (computational) indistinguishability but the opposite direction does not hold. For more details the interested reader is referred to e.g. [Gol98].

2.2 One-Way Functions and Secure Envelopes

A "one-way function" is the most basic cryptographic primitive. Intuitively, an injective function is "one-way" if it can be computed only in a single direction:

Definition 4 *An injective function $f : A_n \rightarrow B_n$ is one-way if (i) f is polynomial-time computable and (ii) for any* PPT *$\mathcal{A}'$ it holds that $\mathbf{Prob}_{r\in_U\mathcal{R};a\in_U A_n}[\mathcal{A}'(r, f(a)) = a]$ is negligible in n.*

From the point of view of protecting information efficiently the existence of a one-way function is not sufficient. Usually more elaborate constructions are necessary in order to build more advanced cryptographic primitives such as a cryptographically strong pseudorandom number generator [HILL99].

To illustrate this point further imagine that one agent wants to use a one-way function f to hide some personal information x. If the agent publishes $f(x)$ the definition above does not guarantee that an adversary who obtains x will be unable to extract any information about x. In fact all one-way functions guarantee that there will be at least one "property" of x that will be hidden (more specifically all one-way functions have a hard-core predicate associated with them — there is a predicate $\mathcal{P}$ that given $f(x)$ is hard to guess $\mathcal{P}(x)$ — see e.g. [Lub96]). But this property will most likely fail to capture all the concealment properties that the agent is interested to sustain in the presence of an eavesdropper. Additionally the element x that the agent wishes to hide is usually not randomly distributed over the domain but rather follows a specific probability distribution that is different compared to the uniform (e.g. compare the distribution of pieces of English text of a certain length encoded in ASCII code to the distribution of random bitstrings of the same length — the two distributions are clearly very different statistically).

As a result a one-way function cannot play the role of a concealment method. Essentially what we would like to simulate here is the effect of a "secure envelope", at least with respect to any eavesdropping (or passive) adversary. Alternatively, we can call this property "hardness of partial information extraction":

Definition 5 *Let $f : A_n \to B_n$ be some computable (probabilistic) function. We say that f is a "secure envelope" (or hides partial information) if for all samplable probability distributions $\mathcal{D}$ over A_n, and for any computable function $g : A_n \to R_n$, if $\mathcal{A}$ is a* PPT *that given $f(x)$ where x is distributed according to $\mathcal{D}$, $\mathcal{A}$ computes $g(x)$, then there exists a* PPT *$\mathcal{A}'$ with the same "functionality" as $\mathcal{A}$ that operates without seeing $f(x)$. Formally, the distance of the probabilities,*

$$\left| \mathbf{Prob}_{r \in_U \mathcal{R}, x \in_{\mathcal{D}} A_n}[\mathcal{A}(r, f(x)) = g(x)] - \mathbf{Prob}_{r' \in_U \mathcal{R}', x \in_{\mathcal{D}} A_n}[\mathcal{A}'(r') = g(x)] \right|$$

is negligible in n, where the probability is taken also over the coin-tosses of f (if f is

probabilistic).

The above definition suggests that if someone given $f(x)$ can extract some information about x then he can also do it without seeing $f(x)$. This captures the intuitive notion that $f(x)$ is a "secure envelope" that engulfs x in such a way so that no polynomial-time bounded probabilistic observer can extract some non-trivial information about x (in this setting "trivial" information is the length of x, etc.).

Note that the above definition clearly suggests that there would be probability distributions that f is incapable of hiding. But this is very natural: for example if the adversary knows that $x \in \{a, b\}$ and x is distributed according to the probability distribution $\mathcal{D}$ with $\mathbf{Prob}_{\mathcal{D}}[x = a] = 1$ and $\mathbf{Prob}_{\mathcal{D}}[x = b] = 0$ it is clear that given $f(x)$ he will guess successfully that $x = a$, regardless of the cryptographic properties of f. The point of the definition above is that the adversary *does not gain any advantage in guessing a property of* x by obtaining $f(x)$.

Definition 5 is still not sufficient to argue that a certain function is useful from a cryptographic viewpoint: this is because it merely formalizes concealment, and does not enforce the recoverability of any property of the input. To capture this we say that a probabilistic secure envelope is *non-trivial*, if: given the output of the secure envelope and the coin tosses that were used in its computation, it is possible to recover the input efficiently. Non-trivial secure envelopes are very useful Cryptographic objects.

2.3 Random Self-Reducibility

A computational problem is a function $\Pi : A_n \to B_n$ where A_n denotes the instance-space and B_n denotes the solution space. A computational problem is solvable by the PPT $\mathcal{A}$ with $\mathcal{A} : A_n \to B_n$ if it holds that $\mathbf{Prob}_{r \in_U \mathcal{R}; X \in_U A_n}[\mathcal{A}(r, X) = \Pi(X)]$ is non-negligible in n.

Definition 6 *A computational problem is strongly randomly self-reducible if there exists a re-randomizing procedure* $R : \mathcal{R}' \times A_n \rightarrow A_n$ *so that*

1. *For any* $X \in A_n$ *the uniform distribution over* A_n *and distribution defined by* $R(\cdot, X)$ *are statistically indistinguishable.*
2. *There is a deterministic polynomial-time procedure that given* r' *and* $\Pi(R(r', X))$ *it returns* $\Pi(X)$.

Intuitively, random self-reducibility suggests that the specific choice of an instance does not affect the probability of successfully finding its solution. In most settings the following weaker property is implied by (strong) random self-reducibility:

Definition 7 *A computational problem* Π *is weakly randomly self-reducible if for any* $\mathcal{A}$ *that solves* Π *there exists a* PPT $\mathcal{A}'$ *so that, for all* $X \in A_n$,

$$| \mathbf{Prob}_{r \in_U \mathcal{R}; X \in_U A_n}[\mathcal{A}(r, X) = \Pi(X)] - \mathbf{Prob}_{r' \in_U \mathcal{R}'}[\mathcal{A}'(r', X)) = \Pi(X)] |$$

is negligible in n.

The property of random self-reducibility is important from an average-case complexity viewpoint since it suggests that all instances are "equally-hard" (up to a polynomial-time transformation). A direct conclusion is that the worst-case hardness of a problem implies "average-case" hardness. Random self-reducibility was studied in several settings, see e.g. [FKN90]. Not many computational problems are known to be randomly self-reducible. To this effect, in many settings one opts for "partial random self-reducibility" properties:

Let $\{(A_n)_i\}_i$ be some parameterization of the instance space A_n with $A_n = \cup_i (A_n)_i$.

Definition 8 *A problem* $\Pi : A_n \rightarrow B_n$ *is type-1 random self-reducible for the parameterization* $\{(A_n)_i\}_i$

1. *There is a re-randomizer procedure* $R : \mathcal{R}' \times (A_n)_i \rightarrow (A_n)_i$ *so that for any* i *and for all* $X \in (A_n)_i$ *the uniform probability distribution over* $(A_n)_i$ *is statistically indistinguishable from the probability distribution defined over* $(A_n)_i$ *by* $R(\cdot, X)$.

2. *There is a deterministic polynomial-time procedure that given* r' *and* $\Pi(R(r', X))$ *it returns* $\Pi(X)$.

A typical example of type-1 random self-reducibility with respect to a parameterization is the discrete-logarithm problem (DLP) over multiplicative groups of order a safe prime[1]. The DLP is defined as follows: the instance space for size n includes all triplets of the form $\langle g, h, p \rangle$ where p is a n-bit prime s.t. $p = 2q+1$ and $g, h \in \mathbb{Z}_p^*$ of order q. The solution of a DLP instance is the least positive (non-zero) integer x so that $g^x \equiv h \pmod{p}$. It is not very hard to show that for a parameterization of the instance space with respect to the selected prime, one can easily randomize the DLP instance with respect to this parameterization since given $\langle g, h, p \rangle$, the re-randomization procedure selects r uniformly at random from $\{1, 2, \ldots, q-1\}$ and computes the DLP instance $\langle g', h', p \rangle := \langle g^r, h^r, p \rangle$. Clearly if one finds x s.t. $(g')^x \equiv h' \pmod{p}$ then it holds that $r^{-1} \cdot x \bmod q$ is the solution of the original instance.

Now we define type-2 random self-reducibility with respect to a parameterization:

Definition 9 *A problem* $\Pi : A_n \rightarrow B_n$ *is type-2 random self-reducible with respect to the parameterization* $\{(A_n)_i\}_i$

1. *There is a re-randomizer procedure* $R : \mathcal{R}' \times (A_n)_i \rightarrow A_n$ *so that for any* i, *the uniform probability distribution over* A_n *is statistically indistinguishable from the probability distribution defined over* A_n *by* $R(\cdot, \cdot)$.

2. *There is a deterministic polynomial-time procedure that given* r' *and* $\Pi(R(r', X))$ *it returns* $\Pi(X)$.

[1] A "safe prime" is a prime number p for which it holds that $p = 2q + 1$ and q is also prime.

Type-2 random self-reducibility w.r.t. to a parameterization $\{(A_n)_i\}_i$ is an important property because it suggests that the particular parameterization does not "affect" the hardness behavior of the problem. More specifically, the intuition is that the restriction of the instance space to the given parameterization does not give any significant advantage for solving the problem.

Random self-reducibility is important from a cryptographic perspective because it deals with the "average-case" hardness of computational problems. Even though strong random self-reducibility is the most desirable, it has not been exhibited by many computational problems (a notable exception is related to the shortest vector problem in lattices, see [Ajt96]). Nevertheless type-1 and type-2 random self-reducibility are still partially satisfactory indications for the appropriateness of a certain problem in Cryptography (and do appear more often).

Chapter 3

Employing Polynomial Reconstruction as an Intractability Assumption

Finding new problems based on which we can design cryptographic primitives is an important research area. Given a presumably hard problem it is usually non-trivial to exploit it directly in Cryptography. Many times, in order to serve as the base for secure cryptographic primitives, we need to find related hard decision problems (predicates). This is the fundamental methodology initiated by Goldwasser and Micali in [GM84] where they started the quest for formal notions and proofs of security in Cryptography. The decision problem's hardness, typically seems related to (or at times proved in some sense related or, even better, reducible from) the hardness of the original problem. Hard predicate assumptions allow formal security proofs (in the form of reductions) for advanced cryptographic primitives such as pseudo-randomness and semantically secure encryption. The first example of a decisional assumption is the Quadratic-Residuosity, which is related to (but not known to be reducible from) Factoring and was employed in designing the first semantically secure encryption scheme [GM84]. Another such assumption is the Decisional Diffie-Hellman which implies the security of ElGamal encryption and other advanced cryptographic

primitives (e.g., [NR97]), and is related to (but not known to be reducible from) the Diffie-Hellman problem.

In this chapter, our goal is to investigate the possibility of cryptographic primitives whose security is based on the problem of *Polynomial Reconstruction* (PR). Recall that the problem of Polynomial Reconstruction is defined as follows: Given n points over a (large) finite field $\mathbb{F}$, such that at least t of them belong to the graph of a polynomial p of degree less than k, recover such a polynomial (where $n > t > k$).

We note that Polynomial Reconstruction is essentially equivalent to the decoding problem of Reed-Solomon codes and naturally has received much attention from a "positive" (coding theoretic) perspective: Starting from the classical algorithm of Berlekamp and Welch([BW86]) which solves Polynomial Reconstruction provided that $t \geq \frac{n+k}{2}$ (error correcting bound for Reed-Solomon Codes), to the recent work of Guruswami and Sudan [GS98] which solves it when $t \geq \sqrt{kn}$ (many solutions are possible in the worst case). The current state of knowledge suggests that for values of t below $\sqrt{kn}$ the problem is hard.

Regarding our goal, Polynomial Reconstruction as is, does not seem to be ready for direct cryptographic exploitation: even if presumed hard, it is not at all clear how to build advanced cryptographic primitives whose security can be reduced to it. Indeed, when Naor and Pinkas [NP99] first employed the problem cryptographically in a context of protocol design, they actually introduced a related pseudorandomness assumption. The relation of this assumption to PR also motivates further investigation.

In this chapter, we first identify a decisional problem naturally related to PR. This problem is based on the following basic question: given a PR-instance that contains n points and an index $i \in \{1, \ldots, n\}$, does the i-th point of the instance belong in the graph of the polynomial solution or not? (note that in the range of our parameters, a PR-instance has a unique solution with very high probability). We formalize the

hardness of this predicate for all indices i as the "Decisional-PR-Assumption" (DPR).

Based on the DPR-Assumption we show: (i) *hardness of partial information extraction*: an adversary with access to a PR-instance who wishes to predict the value of some computable function on a new point of the polynomial-solution, gains only negligible advantage compared to an adversary who wishes to predict the same value without seeing the instance — this holds true even if the point follows an adversarially chosen probability distribution; also: (ii) *pseudorandomness*: PR-instances are pseudorandom in the sense that they are indistinguishable from random sets of points, for any poly-time observer. These results suggest that PR is quite robust in the cryptographic sense and is suitable for employment in cryptographic constructions.

There are several possible advantages of the PR problem which can be exploited by cryptographic primitives built on it, for example: (i) The natural dichotomy and independence exhibited between the key-size (index of error locations) and the size of Reed-Solomon encoded message (or concealed information in PR-based systems) allows key-sizes to be selected independently of (and possibly super-polynomially smaller than) the message size; we know of no other problem that allows such a property in the cryptographic literature. (ii) The PR problem enjoys a unique algebraic structure. (iii) The operation of polynomial interpolation which is basic in PR cryptographic primitives can be implemented quite efficiently (especially in special purpose hardware).

3.1 The Problem

Definition 10 Polynomial Reconstruction (PR). *Given* n, k, t *and* $\{\langle z_i, y_i\rangle\}_{i=1}^{n}$ *with* $z_i \neq z_j$ *for* $i \neq j$, *output all* $\langle p(x), I\rangle$ *such that* $p \in \mathbb{F}[x]$, $\text{degree}(p) < k$, $I \subseteq \{1,\ldots,n\}$, $|I| \geq t$ *and* $\forall i \in I(p(z_i) = y_i)$.

PR as a coding theoretic problem asks for all messages that agree with at least

t positions of the received Reed-Solomon codeword. For a general treatment on the subject the interested reader is referred to [Ber68] or [MS77]. Note that $k < n$ since k/n is the message rate of the code, and that we further require that at least one solution $\langle p(x), I \rangle$ exists.

When $t \geq \frac{n+k}{2}$ then PR$[n, k, t]$ has only one solution and it can be found with the algorithm of Berlekamp and Welch [BW86] ($\frac{n+k}{2}$ is the error-correction bound of the Reed-Solomon codes). When t is beyond the error-correction bound then having more than one solution is possible. Sudan proposed an algorithm that solves the PR beyond the error-correction bound when $t \geq \sqrt{2kn}$ in [Sud97] and later in [GS98], Guruswami and Sudan presented an algorithm that solves the PR for $t > \sqrt{kn}$. In [GRS95] it was proven that when $t > \sqrt{kn}$ the number of solutions is bounded by a polynomial. In [GS98] it is pointed out that the possibility of an algorithm that solves instances for smaller values of t might be limited. We note here that the solvability of PR (and related problems) was also studied in the context of lattices, see [BN00]. Consequently the current state of knowledge implies that PR$[n, k, t]$ is hard for the choice of parameters $t < \sqrt{kn}$.

3.1.1 Structure of the Instance Space

An instance of PR will be denoted by $X := \{\langle z_i, y_i \rangle\}_{i=1}^{n}$; the set of all instances with parameters n, k, t will be denoted by $\mathcal{S}_{n,k,t}$. In order to refer to PR with parameters n, k, t we will write PR$[n, k, t]$. Note that unless stated otherwise we assume that n is polynomially related to $\log |\mathbb{F}|$.

Let $I \subseteq \{1, \ldots, n\}$ with $|I| = t$. We denote by $\mathcal{S}_{n,k,t}(I)$ the subset of $\mathcal{S}_{n,k,t}$ so that for any $X \in \mathcal{S}_{n,k,t}(I)$ it holds that X has a solution of the form $\langle p, I \rangle$. It is clear that $\mathcal{S}_{n,k,t} = \cup_{|I|=t} \mathcal{S}_{n,k,t}(I)$, but $\{\mathcal{S}_{n,k,t}(I)\}_{|I|=t}$ does not constitute a partition of $\mathcal{S}_{n,k,t}$. Nevertheless concentrating on instance sets of the form $\mathcal{S}_{n,k,t}(I)$ is helpful in understanding the structure of $\mathcal{S}_{n,k,t}$.

Lemma 11 *For any $I \subseteq \{1,\ldots,n\}$ with $|I| = t$ it holds that*

$$\#\mathcal{S}_{n,k,t}(I) = (|\mathbb{F}|)_n |\mathbb{F}|^{n-t+k}$$

Proof. Straightforward since $n - t + k$ are exactly the degrees of freedom that each element of $\mathcal{S}_{n,k,t}(I)$ has. ■

Clearly if a PR-instance $X \in \mathcal{S}_{n,k,t}$ has two distinct solutions $\langle p_1, I_1\rangle$ and $\langle p_2, I_2\rangle$, it holds that $X \in \mathcal{S}_{n,k,t}(I_1) \cap \mathcal{S}_{n,k,t}(I_2)$. To determine the likelihood that a given PR-instance has a single solution or more, the following lemma is helpful:

Lemma 12 *(i) For all $I_1, I_2 \subseteq \{1,\ldots,n\}$, with $|I_1| = |I_2| = t$, $I_1 \neq I_2$, it holds that $\#(\mathcal{S}_{n,k,t}(I_1) \cap \mathcal{S}_{n,k,t}(I_2)) \leq (|\mathbb{F}|)_n |\mathbb{F}|^{n-t+k-1}$.*
(ii) The total number of PR-instances of $\mathcal{S}_{n,k,t}$ that have more than one solution is less than $\binom{n}{t}^2 (|\mathbb{F}|)_n |\mathbb{F}|^{n-t+k-1}$.

Proof. (i) Let $|I_1 \cap I_2| = m$; note that $m \in \{0,\ldots,t-1\}$. The $\langle z_1,\ldots,z_n\rangle$ values contribute $(\mathbb{F})_n$ choices. The "free" (noise) points contribute $|\mathbb{F}|^{n-2t+m}$ choices. It remains to find the number of choices due to the y-elements that correspond to the positions $I_1 \cup I_2$. There first solution contributes $|\mathbb{F}|^k$ choices, whereas the second solution, if $m < k$, it contributes $|\mathbb{F}|^{k-m}$. If $m \geq k$ no second solution is feasible. So we have two cases: $m < k$, where $\#(\mathcal{S}_{n,k,t}(I_1) \cap \mathcal{S}_{n,k,t}(I_2)) = (|\mathbb{F}|)_n |\mathbb{F}|^{n-2t+2k}$, and $m \geq k$, where $\#(\mathcal{S}_{n,k,t}(I_1) \cap \mathcal{S}_{n,k,t}(I_2)) = (|\mathbb{F}|)_n |\mathbb{F}|^{n-2t+m+k}$, with $m \in \{k,\ldots,t-1\}$. As a result, independently of the choice of I_1, I_2, $\#(\mathcal{S}_{n,k,t}(I_1) \cap \mathcal{S}_{n,k,t}(I_2)) \leq (|\mathbb{F}|)_n |\mathbb{F}|^{n-t+k-1}$ (recall that $t > k$).
(ii) it follows easily from the fact that the set of all instances of $\mathcal{S}_{n,k,t}$ that have more than one solution is a subset of $\cup_{I_1 \neq I_2} \mathcal{S}_{n,k,t}(I_1) \cap \mathcal{S}_{n,k,t}(I_2)$. ■

The following lemma compares the number of elements of $\mathcal{S}_{n,k,t}$ and $\mathcal{S}_{n,k,t}(I)$ and in combination with the previous lemma it provides an estimate to the number of elements of $\mathcal{S}_{n,k,t}$.

Lemma 13 *Suppose* $\log|\mathbb{F}| \geq 3n$. *For any* $I \subseteq \{1,\ldots,n\}$, $|I| = t$, *it holds that* $\binom{n}{t} - 2^{-n} \leq \frac{\#\mathcal{S}_{n,k,t}}{\#\mathcal{S}_{n,k,t}(I)} \leq \binom{n}{t}$.

Proof. By definition it holds that $\mathcal{S}_{n,k,t} = \cup_{|I|=t}\mathcal{S}_{n,k,t}(I)$. It follows from lemma 11 that $\#\mathcal{S}_{n,k,t}(I) = \#\mathcal{S}_{n,k,t}(I')$ for all I, I'. Now fix some $I \subseteq \{1,\ldots,n\}$, $|I| = t$. It follows that.

$$\binom{n}{t}\#\mathcal{S}_{n,k,t}(I) - \sum_{I_1 \neq I_2} \#(\mathcal{S}_{n,k,t}(I_1) \cap \mathcal{S}_{n,k,t}(I_2)) \leq \#\mathcal{S}_{n,k,t} \leq \binom{n}{t}\#\mathcal{S}_{n,k,t}(I)$$

Next using the upper bound on $\sum_{I_1 \neq I_2} \#(\mathcal{S}_{n,k,t}(I_1) \cap \mathcal{S}_{n,k,t}(I_2))$ that follows from lemma 12, it follows that (using the facts $\log|\mathbb{F}| \geq 3n$, $\binom{n}{t} < 2^n$)

$$\sum_{I_1 \neq I_2} \#(\mathcal{S}_{n,k,t}(I_1) \cap \mathcal{S}_{n,k,t}(I_2)) < \frac{\binom{n}{t}^2 \#\mathcal{S}_{n,k,t}(I)}{|\mathbb{F}|} < \frac{\#\mathcal{S}_{n,k,t}(I)}{2^n}$$

It follows that

$$(\binom{n}{t} - \frac{1}{2^n})\#\mathcal{S}_{n,k,t}(I) \leq \#\mathcal{S}_{n,k,t} \leq \binom{n}{t}\#\mathcal{S}_{n,k,t}(I)$$

which completes the proof. ■

As a result we can draw the following corollary:

Corollary 14 *The number of elements of* $\mathcal{S}_{n,k,t}$ *can be approximated (within negligible error) by* $\binom{n}{t}(|\mathbb{F}|)_n|\mathbb{F}|^{n-k+t}$.

Clearly sampling the uniform distribution over $\mathcal{S}_{n,k,t}(I)$ is straightforward (based on the fact that the uniform distribution over the finite field $\mathbb{F}$ can be sampled — something that can be shown easily). Next we proceed to show that the uniform distribution of PR instances is actually *samplable* (with negligible statistical error). We start with a standard definition:

Definition 15 *A probability distribution $\mathcal{D}$ over some space R of objects of size polynomial in n is called (polynomial time) samplable if there is a* PPT $S_{\mathcal{D}} : \mathcal{R}_{\mathcal{D}} \to R$ *so that the probability assigned to any $y \in R$ by $\mathcal{D}$ is* $\mathbf{Prob}_{\mathcal{D}}[y] = \mathbf{Prob}_{x \in_U \mathcal{R}_{\mathcal{D}}}[S_{\mathcal{D}}(x) = y]$.

Consider the following procedure S that samples $\mathcal{S}_{n,k,t}$: first select n random distinct elements of $\mathbb{F}$, $z_1, \ldots, z_n$. Then, select a random I such that $|I| = t$ and then select a random polynomial p of degree less than k (e.g. by selecting k random elements of $\mathbb{F}$ as its coefficients). Set $y_i := p(z_i)$ for $i \in I$ and select the remaining y_i for $i \notin I$ at random. The output of S is $\{\langle z_i, y_i \rangle\}_{i=1}^{n}$. The following lemma suggests that the described procedure S essentially samples the uniform distribution over $\mathcal{S}_{n,k,t}$.

Lemma 16 *Let* $\log |\mathbb{F}| \geq 3n$. *The probability distribution defined by* S *is statistically indistinguishable from the uniform over $\mathcal{S}_{n,k,t}$. More specifically,* $A := \sum_{X \in \mathcal{S}_{n,k,t}} |\ \mathbf{Prob}[\mathsf{S}(1^n) = X] - \frac{1}{\#\mathcal{S}_{n,k,t}}\ | < 2n2^{-n}$.

Proof. Fix some $X \in \mathcal{S}_{n,k,t}$. If only a single solution $\langle p, I \rangle$ with $|I| = t$ exists in X then it follows easily that there is a unique assignment of the random choices of S that yields X. As a result in this case it holds that $\mathbf{Prob}[\mathsf{S}(1^n) = X] = \frac{1}{\binom{n}{t} \cdot (|\mathbb{F}|)_n |\mathbb{F}|^{n-t+k}}$. Let us partition $\mathcal{S}_{n,k,t}$ to the set $\mathcal{S}_1$ that contains instances X with a single solution as above and let $\mathcal{S}_2 := \mathcal{S}_{n,k,t} - \mathcal{S}_1$. If $\frac{1}{2}A$ is the statistical distance between the two distributions then it follows that:

$$A = A_1 + A_2 =$$

$$= \sum_{X \in \mathcal{S}_1} \left| \frac{1}{\binom{n}{t}(|\mathbb{F}|)_n |\mathbb{F}|^{n-t+k}} - \frac{1}{\#\mathcal{S}_{n,k,t}} \right| + \sum_{X \in \mathcal{S}_2} \left| \mathbf{Prob}[\mathsf{S}(1^n) = X] - \frac{1}{\#\mathcal{S}_{n,k,t}} \right|$$

From lemma 13 it holds that

$$\left| \frac{1}{\binom{n}{t}(|\mathbb{F}|)_n |\mathbb{F}|^{n-t+k}} - \frac{1}{\#\mathcal{S}_{n,k,t}} \right| = \frac{1}{\binom{n}{t}\#\mathcal{S}_{n,k,t}} \left| \frac{\#\mathcal{S}_{n,k,t}}{(|\mathbb{F}|)_n |\mathbb{F}|^{n-t+k}} - \binom{n}{t} \right| < \frac{1}{2^n \binom{n}{t} \#\mathcal{S}_{n,k,t}}$$

It follows that:

$$A_1 = \sum_{X \in \mathcal{S}_1} | \frac{1}{\binom{n}{t}(|\mathbb{F}|)_n |\mathbb{F}|^{n-t+k}} - \frac{1}{\#\mathcal{S}_{n,k,t}} | < \frac{1}{2^n \binom{n}{t}}$$

and as a result A_1 is negligible. Next we proceed to show that A_2 is also negligible. Note that this will follow immediately by the following two facts:

(i) $\sum_{X \in \mathcal{S}_2} \mathbf{Prob}[\mathsf{S}(1^n) = X] < (n-t)2^{-n}$. To see this, let n_X be such that

$$\mathbf{Prob}[\mathsf{S}(1^n) = X] = \frac{n_X}{\binom{n}{t} \cdot (|\mathbb{F}|)_n |\mathbb{F}|^{n-t+k}}$$

It follows that $\sum_{X \in \mathcal{S}_2} n_X = \binom{n}{t} \cdot (|\mathbb{F}|)_n |\mathbb{F}|^{n-t+k} - \#\mathcal{S}_1$. Since $\mathcal{S}_1$ contains all those PR instances that contain exactly one solution it follows easily that

$$\#\mathcal{S}_1 > \binom{n}{t}(|\mathbb{F}|)_n |\mathbb{F}|^k (|\mathbb{F}| - \binom{n}{k})^{n-t}$$

As a result (using the facts $\log |\mathbb{F}| \geq 3n$, $\binom{n}{k} < 2^n$)

$$\sum_{X \in \mathcal{S}_2} n_X < \binom{n}{t} \cdot (|\mathbb{F}|)_n |\mathbb{F}|^{n-t+k} \left(1 - \left(\frac{|\mathbb{F}| - \binom{n}{k}}{|\mathbb{F}|}\right)^{n-t}\right) \Longrightarrow$$

$$\sum_{X \in \mathcal{S}_2} \mathbf{Prob}[\mathsf{S}(1^n) = X] < 1 - \left(1 - \frac{\binom{n}{k}}{|\mathbb{F}|}\right)^{n-t} < 1 - \left(1 - \frac{1}{2^n}\right)^{n-t} = \sum_{i=1}^{n-t} \binom{n-t}{i} \frac{(-1)^{i+1}}{2^{ni}}$$

the sum on the right hand side is easily shown to be less than $(n-t)2^{-n}$.

(ii) $\sum_{X \in \mathcal{S}_2} \frac{1}{\#\mathcal{S}_{n,k,t}} < 2^{-n}$. Indeed the sum equals to $\frac{\#\mathcal{S}_2}{\#\mathcal{S}_{n,k,t}}$ and the stated result follows from lemma 12(ii).

Finally we conclude that $A < \frac{1}{2^n \binom{n}{t}} + \frac{n-t}{2^n} + \frac{1}{2^n} < 2n2^{-n}$. ■

Lemma 17 *Suppose that* $\log |\mathbb{F}| \geq 2n$. *The ratio of the number of PR-instances of* $\mathcal{S}_{n,k,t}$ *with more than one solution, over* $\#\mathcal{S}_{n,k,t}$ *is less than* 2^{-n}.

Proof. Because of lemma 13 it holds that $(\binom{n}{t} - 2^{-n})(|\mathbb{F}|)_n |\mathbb{F}|^{n-t+k} \leq \#\mathcal{S}_{n,k,t} \leq \binom{n}{t}(|\mathbb{F}|)_n |\mathbb{F}|^{n-t+k}$. The number of PR-instances of $\mathcal{S}_{n,k,t}$ with more than one solution

is less than $\binom{n}{t}^2(|\mathbb{F}|)_n|\mathbb{F}|^{n-t+k-1}$ (from lemma 12ii). It follows that the ratio is less than $\binom{n}{t}^2(\binom{n}{t}-2^{-n})^{-1}|\mathbb{F}|^{-1} < 2^{-n}$. ■

It is an immediate corollary from the above lemma that any PPT which samples the uniform distribution over $\mathcal{S}_{n,k,t}$ will select an instance X that has a unique solution with overwhelming probability $1-2^{-n}$. Consequently any instance $X \in \mathcal{S}_{n,k,t}$ uniquely defines a polynomial p (with overwhelming probability) such that degree(p) $< k$. We denote this polynomial by s_X (for solution of X). The set of indices that corresponds to the graph of p which we call "the index-solution set" is denoted by $I(X)$. Obviously, the recovery of s_X implies the recovery of $I(X)$ and vice-versa.

3.1.2 Security Parameters

In our exposition we will use n as be the security parameter. The parameters k, t are functions in n, so that $k < t < n$ and $t < \sqrt{nk}$. The straightforward brute-force algorithm for solving PR$[n,k,t]$ requires checking all possibilities and as a result has complexity proportional to $\min(\binom{n}{k},\binom{n}{t})$. The parameters $[n,k(n),t(n)]$ are called *sound* for PR$[n,k,t]$ if $k(n)$ and $t(n)$ are chosen so that $t < \sqrt{kn}$ and $\min(\binom{n}{k},\binom{n}{t})$ is exponential in n. Note that we will suppress (n) in $k(n),t(n)$. Observe that if $[n,k,t]$ are sound parameters then it also holds that $[n,k+1,t]$ are sound parameters (provided that $k+1 < t$). Intuitively this means that allowing the degree of the solution-polynomial to be greater without changing the other parameters it cannot make the problem easier. We will assume sound parameters throughout.

3.1.3 Partial Random Self-Reducibility

As it is noted in [NP99], Polynomial Reconstruction enjoys a partial self-reducibility property, namely that given an $X := \{\langle z_i, y_i\rangle\}_{i=1}^n \in \mathcal{S}_{n,k,t}$ it is possible to randomize the polynomial solution of X: choose a random polynomial p' of degree less than k and compute the instance $Y := \{\langle z_i, y_i + p(z_i)\rangle\}_{i=1}^n$. Nevertheless this is not at all sufficient

to show that the problem is randomly self-reducible. This is because the procedure does not randomize the points that do not lie in the index-solution-set. Polynomial Reconstruction enjoys yet another partial random self-reducibility property, namely that the choice of the index-solution-set is not important. Informally this can be seen by the fact that one can permute the points of PR-instance by applying a random n-permutation. This fact is of importance from a cryptographic viewpoint since in many settings the index-solution-set plays the role of a cryptographic key. This second partial random self-reducibility is formalized and strengthened in the next section.

3.1.4 Altering The Distribution of PR-Instance Solutions

Suppose that some points of a polynomial solution of PR$[n,k,t]$ instance follow a given (non-uniform over $\mathbb{F}$) probability distribution. If h points of the polynomial solution follow a certain probability distribution we will fix these points to be the values of the polynomial over $\{0,\ldots,h-1\}$. Without loss of generality we assume that $0,\ldots,h-1$ are not equal to any of the $\langle z_1,\ldots,z_n\rangle$ values in a PR-instance (this is an event of negligible probability). Note that alternative "base" values $w_1,\ldots,w_h$ can be used instead of $0,\ldots,h-1$.

Let $[n,k-h,t]$ be sound parameters for some $0<h<k$. Let $\mathcal{D}_h$ be a samplable probability distribution over $\mathbb{F}^h$. We can extend $\mathcal{D}_h$ to be a samplable probability distribution over $\mathcal{S}_{n,k,t}$ by modifying the sampler S of section 3.1.1 so that it selects h values of the polynomial solution following $\mathcal{D}_h$ (instead of at random). We use the notation $\mathsf{S}_{\mathcal{D}_h}$ to denote this generalized sampler over $\mathcal{S}_{n,k,t}$. Note that we will use the same notation $\mathcal{D}_h$ for both probability distributions (over $\mathbb{F}^h$ and $\mathcal{S}_{n,k,t}$). Defining $\mathcal{D}_h$ over $\mathcal{S}_{n,k,t}(I)$ can be done in a similar manner as above, and the sampler will be denoted by $\mathsf{S}^I_{\mathcal{D}_h}$. If the base values are set to $\{w_1,\ldots,w_h\}$ the derived probability distribution over $\mathcal{S}_{n,k,t}$ and $\mathcal{S}_{n,k,t}(I)$ will be denoted by $\mathcal{D}_h^{w_1,\ldots,w_h}$.

The next lemma reveals that even under such a "modified solution distribution",

the particular choice of the index-solution-set does not affect the output behavior of a certain procedure that operates on PR-instances. The core of the proof below is that given a PR-instance with unknown solution one can randomly permute the points in the instance.

Lemma 18 *Let $\mathcal{D}_h$ be a probability distribution over $\mathbb{F}^h$, with $h \in \{0,\ldots,k\}$. Let $\mathcal{A} : \mathcal{S}_{n,k,t} \to V$ be some PPT. Then it holds that there exists a PPT $\mathcal{A}'$ s.t. for all $v \in V$ and $I \subseteq \{1,\ldots,n\}$ with $|I| = t$.*

$$| \mathbf{Prob}_{X \in_{\mathcal{D}_h} \mathcal{S}_{n,k,t}}[\mathcal{A}(X) = v] - \mathbf{Prob}_{X \in_{\mathcal{D}_h} \mathcal{S}_{n,k,t}(I)}[\mathcal{A}'(X) = v] |$$

is negligible in n.

Proof. Fix some samplable distribution $\mathcal{D}_h$ over $\mathbb{F}^h$, a $v \in V$, and some $I \subseteq \{1,\ldots,n\}$ with $|I| = t$. Let $S_{\mathcal{D}_h} : \mathcal{R}_{\mathcal{D}_h} \to \mathbb{F}^h$ be the PPT that samples $\mathcal{D}_h$. Let $\rho \in \Re$ be the randomness used by $\mathsf{S}_{\mathcal{D}_h}$, to sample an element of $\mathcal{S}_{n,k,t}$, i.e. $\rho := \langle I, z_1, \ldots, z_n, m_1, \ldots, m_{k-h}, r, y_1, \ldots, y_{n-t} \rangle$: it holds that

$$\#\Re = \binom{n}{t} (\mathbb{F})_n |\mathbb{F}|^{k-h} \mathcal{R}_{\mathcal{D}_h} |\mathbb{F}|^{n-t}$$

Similarly denote by $\rho' \in \Re'$ to be the randomness used by $\mathsf{S}^I_{\mathcal{D}_h}$, i.e.

$$\rho' := \langle z_1, \ldots, z_n, m_1, \ldots, m_{k-h}, r, y_1, \ldots, y_{n-t} \rangle$$

it holds that $\#\Re' = (\mathbb{F})_n |\mathbb{F}|^{k-h} \mathcal{R}_{\mathcal{D}_h} |\mathbb{F}|^{n-t}$. It follows $\#\Re = \#\Re' \binom{n}{t}$. Regarding the probability of $\mathcal{A}$ to return v, we have that:

$$\mathbf{Prob}_{X \in_{\mathcal{D}_h} \mathcal{S}_{n,k,t}}[\mathcal{A}(X) = v] = \mathbf{Prob}_{\rho \in_U \Re}[\mathcal{A}(\mathsf{S}_{\mathcal{D}_h}(\rho)) = v]$$

Now consider the PPT $\mathcal{A}'$ that on input X, first it selects a random permutation π, it permutes the pairs of X according to π to obtain X^π and then it simulates $\mathcal{A}$.

$$\mathbf{Prob}_{X\in_{\mathcal{D}_h}\mathcal{S}_{n,k,t}(I);\pi\in_U \mathrm{Perm}(n)}[\mathcal{A}'(\pi, X) = v] = \mathbf{Prob}_{\rho'\in_U\Re';\pi\in_U\mathrm{Perm}(n)}[\mathcal{A}([\mathsf{S}^I_{\mathcal{D}_h}(\rho')]^\pi) = v]$$

Assume that $\mathcal{A}$ does $q(n)$ coin-tosses and define $C := \{\langle b, \rho\rangle \mid \mathcal{A}(b, \mathsf{S}_{\mathcal{D}_h}(\rho)) = v\}$ and $D := \{\langle b, \rho', \pi\rangle \mid \mathcal{A}(b, [\mathsf{S}^I_{\mathcal{D}_h}(\rho)]^\pi) = v\}$, where $b \in \{0,1\}^{q(n)}$.

It follows that

$$\mathbf{Prob}_{\rho\in_U\Re}[\mathcal{A}(\mathsf{S}_{\mathcal{D}_h}(\rho)) = v] = \frac{\#C}{2^{q(n)}\#\Re}$$

and

$$\mathbf{Prob}_{\rho'\in_U\Re';\pi\in_U\mathrm{Perm}(n)}[\mathcal{A}([\mathsf{S}^I_{\mathcal{D}_h}(\rho')]^\pi) = v] = \frac{\#D}{2^{q(n)}n!\#\Re'}$$

Consider a mapping $J : \Re' \times \mathrm{Perm}(n) \to \Re$ so that if $\rho = J(\rho', \pi)$ with

$$\rho = \langle I^\rho, z^\rho_1, \ldots, z^\rho_n, m^\rho_1, \ldots, m^\rho_{k-h}, r^\rho, y^\rho_1, \ldots, y^\rho_{n-t}\rangle$$

and

$$\rho' = \langle z^{\rho'}_1, \ldots, z^{\rho'}_n, m^{\rho'}_1, \ldots, m^{\rho'}_{k-h}, r^{\rho'}, y^{\rho'}_1, \ldots, y^{\rho'}_{n-t}\rangle$$

it holds that $z^\rho_i = z^{\rho'}_i$, $m^\rho_j = m^{\rho'}_j$, $r^\rho = r^{\rho'}$ and $y^\rho_\ell = y^{\rho'}_\ell$, for $i = 1, \ldots, n$, $j = 1, \ldots, k-h$ and $\ell = 1, \ldots, n-t$ and additionally $I^\rho = \{\pi(i) \mid i \in I\}$. It is easy to see that a certain $\rho \in \Re$ has $t!(n-t)!$ pre-images under J. It follows that $\#D = t!(n-t)!\#C$ and as a result:

$$\mathbf{Prob}_{\rho\in_U\Re}[\mathcal{A}(\mathsf{S}_{\mathcal{D}_h}(\rho)) = v] = \mathbf{Prob}_{\rho'\in_U\Re';\pi\in_U\mathrm{Perm}(n)}[\mathcal{A}([\mathsf{S}^I_{\mathcal{D}_h}(\rho')]^\pi) = v]$$

the result of the theorem follows. ■

Note that in the statement of the lemma above the choice of the points $\{0, \ldots, h-1\}$ as the ones that will be distributed according to some probability distribution is arbitrary as it is very easy to reformulate the above result so that some other collection of "base" values is selected. Additionally the value v used above can be generalized to being a function of X in a straightforward manner, without any modifications in the proof.

3.1.5 The Intractability Assumption

A decision problem that relates naturally to the hardness of solving an instance X of PR$[n,k,t]$ is the following: given X and an index $i \in \{1,\ldots,n\}$ decide whether $i \in I(X)$. We postulate that such decision is computationally hard to make whenever PR is hard. Since this has to hold true for all indices we will use a counter-positive argument to formalize the related decisional intractability assumption. In the definition below we describe a pair of predicates that refutes the assumption by "revealing" one of the points that belongs in the graph of the solution-polynomial (note that we formulate probabilities independently of the index-solution-set since given any PR-instance the index-solution-set can be randomized — see lemma 18):

Definition 19 *A pair of* PPT *predicates $\mathcal{A}_1, \mathcal{A}_2$ is called a gap-predicate-pair for the parameters n,k,t if for all $I \subseteq \{1,\ldots,n\}$ with $|I| = t$ it holds that:*

$$|\ \mathbf{Prob}[\mathcal{A}_1(i,X)=1] - \mathbf{Prob}[\mathcal{A}_2(i,X)=1]\ | = \begin{cases} \text{negligible} & \forall i \notin I \\ \text{non-negligible} & \text{for some } i \in I, \\ & i \leq n-k \end{cases}$$

where the probabilities are taken over all choices of $X \in \mathcal{S}_{n,k,t}(I)$ and internal coin-tosses of the predicates $\mathcal{A}_1, \mathcal{A}_2$.

A gap-predicate-pair when given a PR instance X and $i \in \{1,\ldots,n\}$ exhibits a measurable difference for at least one $i \in I(X)$, where at the same time it exhibits no measurable difference for indices outside $I(X)$. Using this, we formulate the Decisional-PR-Assumption as follows:

Decisional-PR-Assumption. (DPR$[n,k,t]$)
For any sound parameters $[n,k,t]$ there does not exist a gap-predicate-pair.

The relation of DPR to the Polynomial Reconstruction problem is revealed in the following two facts which are used to underline the justification for our intractability assumption. The first is straightforward:

Fact 20 *The existence of a polynomial-time algorithm for* PR$[n,k,t]$ *violates the* DPR$[n,k,t]$.

To state the second fact we need a definition: a predicate $\mathcal{A} : \cup_j D_j \to \{0,1\}$ is called *independently samplable* over $\cup_j D_j$ if there is a PPT $S_{\mathcal{A}}$ that given $u \in \mathbb{N}$ and $X \in D_j$, it draws u independently sampled values of $\mathcal{A}$ over the space D_j. In particular, given $X \in D_j$, it holds that $S_{\mathcal{A}}(u,X) := \langle c_1,\ldots,c_u\rangle$ where each c_i is distributed over $\{0,1\}$ according to $\mathcal{A}(Y)$ where Y is uniformly selected over D_j. We denote by $\sum S_{\mathcal{A}}(u,X)$ the sum $\sum_{i=1}^{u} c_i$.

Lemma 21 *If there exists a gap-predicate-pair $\mathcal{A}_1,\mathcal{A}_2$ so that the predicates are independently samplable over the space $\cup_I \mathcal{S}_{n,k,t}(I)$, it follows that* PR$[n,k,t]$ *is solvable with overwhelming probability.*

Proof. First we show how to obtain an $i \in I$ with overwhelming probability. Let $\mathcal{A}_1,\mathcal{A}_2$ be a gap-predicate-pair and denote the non-negligible probability of revealing an index of the index-solution-set by $\alpha(n)$. Suppose we are given some $X \in \mathcal{S}_{n,k,t}$, let $I := I(X)$.

Since $\alpha(n)$ is non-negligible it follows that $\alpha(n) \geq \frac{1}{n^c}$ for some c and sufficiently large n. Let $N := n^{2c+1}$. Consider the following procedure $\mathcal{B}$: first compute the values $a_i := \sum S_{\mathcal{A}_1}(N,i,X)$ and $a'_i := \sum S_{\mathcal{A}_2}(N,i,X)$ for all $i = 1,\ldots,n$ (note that $S_{\mathcal{A}_1}(N,i,X) := \langle \mathcal{A}_1(i,X_1),\ldots,\mathcal{A}_1(i,X_N)\rangle$ and similarly for $S_{\mathcal{A}_2}$). For all $i \in \{1,\ldots,n\}$ check the difference $|a_i - a'_i|$. If it is discovered that for some i, $|a_i - a'_i| \geq \frac{2n^{c+1}}{3}$, output i as a "good" index (i.e. an index that belongs in I). If no such i is discovered the procedure fails.

We show that for any $X \in \mathcal{S}_{n,k,t}$ the above procedure returns an element of $I(X)$ with overwhelming probability. Let A_i and A'_i be the random variables that correspond to the computed values a_i and a'_i. Let μ_i and μ'_i denote the expected values of A_i

and A'_i. By definition it holds that A_i, A'_i follow the Binomial probability distribution over N Bernoulli trials with probability of success $p_i := \mathbf{Prob}_{X \in_U \mathcal{S}_{n,k,t}(I)}[\mathcal{A}_1(i, X) = 1]$ and $p'_i := \mathbf{Prob}_{X \in_U \mathcal{S}_{n,k,t}(I)}[\mathcal{A}_2(i, X) = 1]$ respectively. Using the Chernoff bound we have that for $\epsilon > 0$, $\mathbf{Prob}[|A_i - Np_i| > \epsilon N] \leq 2e^{-2\epsilon^2 N}$ and $\mathbf{Prob}[|A'_i - Np'_i| > \epsilon N] \leq 2e^{-2\epsilon^2 N}$. Now observe that:

$$|Np_i - Np'_i| - |A_i - Np_i| - |A'_i - Np'_i| \leq |A_i - A'_i| \leq |Np_i - Np'_i| + |A_i - Np_i| + |A'_i - Np'_i|$$

Consider the following two facts:

(a) Suppose that $i \notin I$: then it holds that $q_i := \mathbf{Prob}[|A_i - A'_i| \geq \frac{n^{c+1}}{2}]$ is negligible in n. Indeed, $q_i \leq \mathbf{Prob}[|A_i - Np_i| + |A'_i - Np'_i| + |Np_i - Np'_i| \geq \frac{n^{c+1}}{2}]$. Now because $|p_i - p'_i|$ is negligible it follows that for sufficiently large n it holds that $|p_i - p'_i| < \frac{1}{6n^c}$. As a result $q_i \leq \mathbf{Prob}[|A_i - Np_i| + |A'_i - Np'_i| \geq \frac{n^{c+1}}{3}]$. It follows that $q_i \leq \mathbf{Prob}[|A_i - Np_i| \geq \frac{n^{c+1}}{6}] + \mathbf{Prob}[|A'_i - Np'_i| \geq \frac{n^{c+1}}{6}]$ and using the Chernoff bound for $\epsilon := \frac{1}{6n^{c-1}}$ we conclude that $q_i \leq 4e^{-(2/36)n^2}$ which is clearly negligible.

(b) Suppose that $i_0 \in I$ is the index for which $\alpha(n) = |p_{i_0} - p'_{i_0}|$ is non-negligible. The probability $q_{i_0} := \mathbf{Prob}[|A_{i_0} - A'_{i_0}| \leq \frac{2n^{c+1}}{3}]$ is negligible in n: first observe that $q_{i_0} \leq \mathbf{Prob}[|A_{i_0} - Np_{i_0}| + |A'_{i_0} - Np'_{i_0}| \geq N|p_{i_0} - p'_{i_0}| - \frac{2n^{c+1}}{3}]$. We know that $|p_{i_0} - p'_{i_0}| \geq \frac{1}{n^c}$ for sufficiently large n. As a result $q_{i_0} \leq \mathbf{Prob}[|A_{i_0} - Np_{i_0}| + |A'_{i_0} - Np'_{i_0}| \geq \frac{n^{c+1}}{3}]$. This probability was shown in case (a) above to be negligible.

Using the above two facts we deduce the following about the procedure $\mathcal{B}$:

1. The procedure fails with negligible probability. This is because of fact (b).
2. The procedure will report an index that is not in the index-solution set with negligible probability. This is because of fact (a).

It follows that, given *any* $X \in \mathcal{S}_{n,k,t}$, $\mathcal{B}$ reports an index of the index solution set $I(X)$ with overwhelming probability. Moreover for such index i_1 it will hold that $i_1 \leq n - k$ with overwhelming probability (because of the corresponding property of the gap-predicate-pair).

Now we modify the instance X as follows: we substitute the i_1-th point with the n-th point to obtain the altered instance X_2. Subsequently we repeat the procedure $\mathcal{B}$ that will recovers an index of the index-solution-set (different from i_1). By repeating the above k times we obtain k points of the solution polynomial of X and the solution follows by interpolation. This will be done with overwhelming probability. ■

Fact 22 *Violating the* DPR *by an independently samplable gap-predicate-pair with parameters* $[n, k, t]$ *implies that* PR$[n, k, t]$ *is solvable with overwhelming probability.*

3.2 Hardness of Recovering Partial Information of any Specific Polynomial Value

In this section we show that PR$[n, k, t]$ "leaks no partial information" about any specific polynomial value under the DPR-Assumption. In particular, we show that for some fixed value $w \in \mathbb{F}$, given an instance $X := \{\langle z_i, y_i \rangle\}_{i=1}^{n} \in \mathcal{S}_{n,k,t}$ with $w \notin \{z_1, \ldots, z_n\}$, we get no polynomial advantage in predicting the value of *any* function g over the polynomial value $s_X(w)$ for $s_X(w)$ drawn from any polynomially samplable probability distribution $\mathcal{D}$, unless the DPR fails for parameters $[n, k-1, t]$. In the remaining of the section we will fix $w \in \mathbb{F}$ and we will assume that $\mathcal{S}_{n,k,t}$ does not contain instances with w among the z-values (which is a negligible probability event). The generality of the proof stems from the fact that we can map a PR$[n, k-1, t]$-instance X into a PR$[n, k, t]$-instance X' of which we can select the value $s_{X'}(w)$. Then, we can use any algorithm that makes a non-negligible prediction regarding some property of $s_{X'}(w)$ to extract a parameterized predicate that is sensitive to a parameter choice inside the index-solution-set. This predicate yields a gap-predicate-pair that violates DPR$[n, k-1, t]$.

For the rest of the section fix some value $w \in \mathbb{F}$. Next, we formalize the concept of "leaking no partial information." Informally, we can describe the definition as

follows: for any PPT that predicts the value of $g(s_X(w))$ given a PR instance, there is another algorithm with essentially the same functionality that operates *without* the PR instance (cf. definition 5).

Definition 23 PR$[n,k,t]$ *leaks no partial information means that for all poly-time computable* $g : \mathbb{F} \to R$ *and all polynomial-time samplable probability distributions* $\mathcal{D}_1$ *over* $\mathbb{F}$ *it holds: for all* PPT $\mathcal{A}$ *there exists a* PPT $\mathcal{A}'$ *such that the following is negligible in* n*:*

$$|\ \mathbf{Prob}_{r\in_U \mathcal{R}:X\in_{\mathcal{D}_1^w}\mathcal{S}_{n,k,t}}[\mathcal{A}(r,X) = g(s_X(w))] - \mathbf{Prob}_{r'\in_U \mathcal{R}':u\in_{\mathcal{D}_1}\mathbb{F}}[\mathcal{A}'(r') = g(u)]\ |$$

A consequence of lemma 18 is that the definition above can be made more specific so that: for all PPT $\mathcal{A}$ there exists a PPT $\mathcal{A}'$ so that for all $I \subseteq \{1,\ldots,n\}$ with $|I| = t$ it holds that the following is negligible in n:

$$|\ \mathbf{Prob}_{r\in_U \mathcal{R}:X\in_{\mathcal{D}_1^w}\mathcal{S}_{n,k,t}(I)}[\mathcal{A}(r,X) = g(s_X(w))] - \mathbf{Prob}_{r'\in_U \mathcal{R}':u\in_{\mathcal{D}_1}\mathbb{F}}[\mathcal{A}'(r') = g(u)]\ |$$

So, the probability of success of any PPT $\mathcal{A}$ is taken over $\mathcal{S}_{n,k,t}(I)$ following the distribution $\mathcal{D}_1^w$, independently of the index-solution-set I. The core of the proof that PR leaks no partial information is the following lemma:

Lemma 24 *Suppose that there is a poly-time computable* $g : \mathbb{F} \to R$ *and a probability distribution* $\mathcal{D}_1$ *for which* PR$[n,k,t]$ *leaks partial information. Then there exists a* PPT $\mathcal{B}$ *such that for all* $I \subseteq \{1,\ldots,n\}$ *with* $|I| = t$*, if* $\beta_i(n) := \mathbf{Prob}_{\rho\in_U \mathcal{R}:X\in_U \mathcal{S}_{n,k-1,t}(I)}[\mathcal{B}(i,\rho,X) = 1]$ *with* $i \in \{0,\ldots,n\}$ *it holds that*

1. *For all* $i \notin I$ $|\beta_{i-1}(n) - \beta_i(n)|$ *is negligible.*
2. *There exists an* $i_0 \in I$ *such that* $|\beta_{i_0-1}(n) - \beta_{i_0}(n)|$ *is non-negligible and* $i_0 \leq n - k + 1$.

Proof. For simplicity we assume that $\mathcal{D}_1$ is the uniform distribution. The proof is similar in both cases (see below for comments in the case $\mathcal{D}$ is not uniform). Regarding

the success probability $\alpha(n)$ of $\mathcal{A}$ we have that for all I with $|I| = t$, and for all PPT $\mathcal{A}'$ the probability distance below is non-negligible in n:

$$|\ \mathbf{Prob}_{r\in_U\mathcal{R};X\in_U\mathcal{S}_{n,k,t}(I)}[\mathcal{A}(r,X) = g(s_X(w))] - \mathbf{Prob}_{r'\in_U\mathcal{R}';u\in_U\mathbb{F}}[\mathcal{A}'(r') = g(u)]\ |$$

Let $\mathcal{B}$ be the following PPT that operates on $\mathcal{S}_{n,k-1,t}(I)$ with random input string $\rho := \langle u, y, r\rangle \in_U \bar{\mathcal{R}} := \mathbb{F} \times \mathbb{F}^n \times \mathcal{R}$ (in the case $\mathcal{D}$ is not uniform, u is not part of the random input of $\mathcal{B}$ but rather it is sampled using the PPT that samples $\mathcal{D}_1$). Given some $X \in \mathcal{S}_{n,k-1,t}(I) := \{\langle z_i, y_i\rangle\}_{i=1}^n$. The set of pairs $X^* := \{\langle z_i, (z_i - w)y_i + u\rangle\}_{i=1}^n$ is computed. Note that X^* is a random instance of $\mathcal{S}_{n,k,t}(I)$ (similarly if u was distributed according to some non-uniform distribution $\mathcal{D}_1$, then X would follow the corresponding distribution $\mathcal{D}_1$ over $\mathcal{S}_{n,k,t}$). Subsequently the y-part of the first i pairs of X^* is randomized by substituting them with the first i values of the given string $y \in \mathbb{F}^n$. The resulting partially randomized instance is denoted by X_i^*. Then $\mathcal{A}$ is simulated on input (r, X_i^*). If $\mathcal{A}$ returns $g(u)$ (i.e. $\mathcal{A}$ is correct) then $\mathcal{B}$ returns 1 (0 otherwise).

It is easy to see that $\beta_0(n) = \alpha(n)$. When $i = n - k + 1$, $\mathcal{B}$ completely randomizes the first $n-k+1$ positions of the y-part of the constructed $\mathcal{S}_{n,k,t}(I)$ instance. Consider a PPT $\mathcal{A}'$ that first samples a random $Y \in \mathcal{S}_n := (\mathbb{F})_n \times \mathbb{F}^n$ (where $(\mathbb{F})_n$ denotes the set of all n-tuples over $\mathbb{F}$ without repetitions) and then simulates $\mathcal{A}$ on Y. It holds that,

$$\alpha'(n) := \mathbf{Prob}_{u\in_U\mathbb{F}}[\mathcal{A}(\cdot) = g(u)] = \mathbf{Prob}_{r\in_U\mathcal{R};Y\in_U\mathcal{S}_n;u\in_U\mathbb{F}}[\mathcal{A}(r,Y) = g(u)]$$

Let $C' := \{\langle r, Y, u\rangle \mid \mathcal{A}(r, Y) = g(u); Y \in \mathcal{S}_n\}$, it holds that: $\alpha'(n) = \frac{\#C'}{\#\mathcal{R}\times\mathcal{S}_n\times\mathbb{F}}$. We want to compare the probability $\beta_{n-k+1}(n)$ to $\alpha'(n)$. Define the mapping

$$J(i, u, y, X) := \langle X_i^*, u\rangle$$

where X_i^* is defined as in the description of $\mathcal{B}$. Given a certain $\langle Y, u\rangle$ for a $Y \in \mathcal{S}_n$ we want to compute how many pre-images of the form $\langle y, X\rangle$ has, under the mapping

$J(n-k+1,u,\cdot,\cdot)$. Let $h := |I \cap \{n-k+2,\ldots,n\}|$: obviously $h \leq k-1$. Fix h values of the polynomial-solution of X to the corresponding y-positions of Y and $k-1-h$ values of the non-polynomial values of X to the corresponding positions of Y. This leaves a total of $|\mathbb{F}|^{n-t+k-1}$ choices for the pre-images of $\langle Y, u\rangle$. It follows that:

$$\mathcal{B}_{n-k+1}(n) = \frac{|\mathbb{F}|^{n-t+k-1}\#C'}{\#\mathcal{R}\times\mathbb{F}\times\mathbb{F}^n\times\mathcal{S}_{n,k-1,t}(I)} = \frac{|\mathbb{F}|^{n-t+k-1}\#C'}{\#\mathcal{R}\cdot|\mathbb{F}|\cdot(|\mathbb{F}|)_n\cdot|\mathbb{F}|^{2n-t+k-1}} =$$

$$= \frac{\#C'}{\#\mathcal{R}\cdot|\mathbb{F}|\cdot(|\mathbb{F}|)_n\cdot|\mathbb{F}|^n} = \alpha'(n)$$

From the assumption of the theorem it is immediate that $|\alpha(n)-\alpha'(n)|$ is non-negligible and as a result we conclude that $|\mathcal{B}_0(n)-\mathcal{B}_{n-k+1}(n)|$ is non-negligible in n. It follows easily that for some $i_0 \in \{1,\ldots,n-k+1\}$ it should be the case that $|\mathcal{B}_{i_0-1}(n)-\mathcal{B}_{i_0}(n)|$ is non-negligible (by the triangular inequality). It remains to show that it cannot be the case that $i_0 \notin I$.

In particular we will show that for any $i \notin I$ it holds that $|\mathcal{B}_{i-1}(n)-\mathcal{B}_i(n)|$ is negligible.

Let $C_i := \{\langle r,y,X,u\rangle \mid \mathcal{A}(r,X_i^*) = g(u); X_i^* = J(i,X,y,u); X \in \mathcal{S}_{n,k-1,t}(I)\}$. It follows that

$$\mathbf{Prob}_{r\in_U\mathcal{R};y\in_U\mathbb{F}^n;X\in\mathcal{S}_{n,k,t}(I);u\in_U\mathbb{F}}[\mathcal{B}(i,u|y|r,X)=1] = \frac{\#C_i}{\#\mathcal{R}\times\mathbb{F}^n\times\mathcal{S}_{n,k-1,t}(I)\times\mathbb{F}}$$

Suppose $i \notin I$. Next we will compare the number of elements of $\#C_i$ and $\#C_{i-1}$.

Let $\langle r,y^-,X^-,u\rangle$ be an element of $\mathcal{R}\times\mathbb{F}^n\times\mathcal{S}_{n,k-1,t}(I)\times\mathbb{F}$ with the i-th position of y and the i-th y-position of X left "blank." Define $V_{r,y^-,X^-,u} := \{v \mid \mathcal{A}(r,y^-/v/X^-) = g(u)\}$: here $y^-/v/X^-$ denotes the set of pairs $\{\langle z_i,y_i'\rangle\}_{i=1}^n$ such that up to $i-1$ y_i' agrees with y, $y_i' = v$ and from $i+1$ and on $y_i' = (z_i-w)y_i+u$ (where $X^- = \{\langle z_i,y_i\rangle\}_{i=1}^n$). Any $\langle r,y^-,X^-,u\rangle$ together with some $v \in V_{r,y^-,X^-,u}$ can be extended to:

- $|\mathbb{F}|$ tuples $\langle r,y^-_{[v']},X^-_{[v]},u\rangle$ that belong in C_{i-1}; the number of tuples stems from the free choice of $v' \in \mathbb{F}$.

- $|\mathbb{F}|$ tuples $\langle r, y_{[v]}^{-}, X_{[v']}^{-}, u\rangle$ that belong in C_i; the number of tuples stems from the free choice of $v' \in \mathbb{F}$.

It follows that

$$\#C_i = |\mathbb{F}| \sum_{\langle r, y^{-}, X^{-}, u\rangle} \#V_{r, y^{-}, X^{-}, u} = \#C_{i-1}$$

and as a result $\mathcal{Z}_{i-1}(n) = \mathcal{Z}_i(n)$. ■

The proof of this Lemma is a crucial contribution. It exhibits the two main proof-techniques used throughout; one technique involves controlling portions of the instance's solution, whereas the other technique involves a "walking argument" over the points of the instance. Now observe that if $\mathcal{A}_1(i, r, X) := \mathcal{B}(i, r, X)$ and $\mathcal{A}_2(i, r, X) := \mathcal{B}(i-1, r, X)$, it follows easily that $\mathcal{A}_1, \mathcal{A}_2$ is a gap-predicate-pair. As a result,

Theorem 25 *Suppose that there is a poly-time computable $g : \mathbb{F} \to R$ and a probability distribution $\mathcal{D}_1$ for which* PR$[n, k, t]$ *leaks partial information. Then the DPR-Assumption fails for parameters* $[n, k-1, t]$.

Proof. The proof is immediate from lemma 24 and the definition of the DPR assumption. ■

In the rest of the section we present special cases of the above Theorem which appear frequently in cryptographic settings. Let us assume that the distribution $\mathcal{D}_1$ is uniform. Let $g : \mathbb{F} \to R$ be a poly-time computable function. Define $\mathbb{F}_a = \{u \mid g(u) = a; u \in \mathbb{F}\}$ for any $a \in R$. We say that g is *balanced* if for all $a \in R$ and all polynomials q it holds that $\left| \frac{|\mathbb{F}_a|}{|\mathbb{F}|} - \frac{1}{|R|} \right| < \frac{1}{q(\log|\mathbb{F}|)}$ (for sufficiently large $|\mathbb{F}|$). The balanced property means that any image under g corresponds to roughly the same number of pre-images. This is a very general condition that applies to individual bits of elements of $\mathbb{F}$ as well as to various length bit-sequences of elements of $\mathbb{F}$.

Naturally, guessing an unknown value of a balanced function with a uniformly distributed pre-image cannot be done with probability significantly greater than $1/|R|$:

Fact 26 *Let $g : \mathbb{F} \to R$ be balanced, poly-time computable and let n be polynomially related to $\log|\mathbb{F}|$. Then, for any PPT in n, $\mathcal{A}'$, if $\alpha'(n) := \mathbf{Prob}_{r' \in_U \mathcal{R}' : u \in_U \mathbb{F}}[\mathcal{A}'(r') = g(u)]$ it holds that $|\alpha'(n) - \frac{1}{|R|}|$ is negligible in $\log|\mathbb{F}|$.*

Proof. Let $\mathcal{R}'_a := \{r' \mid \mathcal{A}'(r') = a\}$ for any $a \in R$. Note that it holds that $\cup_{a \in R} \mathcal{R}'_a = \mathcal{R}'$. Let q be any polynomial: now because g is balanced:

$$\alpha'(n) = \frac{\sum_{a \in R} |\mathbb{F}_a||\mathcal{R}'_a|}{|\mathbb{F}||\mathcal{R}'|} < \frac{\sum_{a \in R} |\mathcal{R}'_a|}{|\mathcal{R}'|}\left(\frac{1}{|R|} + \frac{1}{q(\log|\mathbb{F}|)}\right) = \frac{1}{|R|} + \frac{1}{q(\log|\mathbb{F}|)}$$

and

$$\alpha'(n) = \frac{\sum_{a \in R} |\mathbb{F}_a||\mathcal{R}'_a|}{|\mathbb{F}||\mathcal{R}'|} > \frac{\sum_{a \in R} |\mathcal{R}'_a|}{|\mathcal{R}'|}\left(\frac{1}{|R|} - \frac{1}{q(\log|\mathbb{F}|)}\right) = \frac{1}{|R|} - \frac{1}{q(\log|\mathbb{F}|)}$$

consequently $|\alpha'(n) - \frac{1}{|R|}|$ is negligible in $\log|\mathbb{F}|$. ■

The corollary of fact 26 and theorem 25 is the following:

Corollary 27 *For any balanced $g : \mathbb{F} \to R$, the success of any PPT $\mathcal{A}$ that given $X \in \mathcal{S}_{n,k,t}$, computes the value $g(s_X(w))$ is only by a negligible fraction different than $1/|R|$ unless the* DPR*-Assumption fails for parameters $[n, k-1, t]$.*

More specifically we can give the following examples of balanced predicates/ functions that are hard to compute given a $\mathrm{PR}[n, k, t]$-instance:

Proposition 28 *The following problems are hard under the* $\mathrm{DPR}[n, k-1, t]$*:*

1. *Let $\mathrm{BIT}_l(a)$ denote the l-th LSB of $a \in \mathbb{F}$. Given $X \in \mathcal{S}_{n,k,t}$ predict $\mathrm{BIT}_l(s_X(w))$ with non-negligible advantage where l represents any bit, except the $\log\log|\mathbb{F}|$ most significant — in particular l as a function of $\log|\mathbb{F}|$ should satisfy that for any $c \in \mathbb{N}$, $l < \log|\mathbb{F}| - c\log\log|\mathbb{F}|$ for sufficiently large $\log|\mathbb{F}|$.*

2. *Let* $\mathrm{BITS}_l(a)$ *denote the sequence of the* l *least significant bits of* $a \in \mathbb{F}$. *Given* $X \in \mathcal{S}_{n,k,t}$ *predict* $\mathrm{BITS}_l(s_X(w))$ *with probability* $\frac{1}{2^l} + \alpha(n)$ *where* $\alpha(n)$ *is non-negligible.*

3. *Let* $\mathrm{QR}(a)$ *be 1 iff* $a \in \mathbb{F}$ *is a quadratic residue, and assume* $\mathbb{F}$ *is of prime order. Given* $X \in \mathcal{S}_{n,k,t}$ *predict* $\mathrm{QR}(s_X(w))$ *with non-negligible advantage.*

Proof. (1) Let H_v denote the number of elements of $\mathbb{F}$ that their l-th LSB is v (where $v \in \{0,1\}$). We want to show that $\frac{|\mathbb{F}_0|-|\mathbb{F}_1|}{|\mathbb{F}|}$ is negligible in $\log|\mathbb{F}|$. Let $f := |\mathbb{F}| \bmod 2^l$. If is easy to see that $|\mathbb{F}_0|-|\mathbb{F}_1| = f$ if $f \leq 2^{l-1}$ and that $|\mathbb{F}_0|-|\mathbb{F}_1| = 2^l - f$ if $f > 2^{l-1}$. At any rate we would like to show that $\frac{2^{l-1}}{|\mathbb{F}|}$ is negligible in $\log|\mathbb{F}|$, which is easy to see under the condition of the theorem.

(2) For any bitstring $b \in \{0,1\}^l$ (where $l = 1,\ldots,\lfloor\log|\mathbb{F}|\rfloor$) it holds that $|\mathbb{F}_b|$ is either (a) $\lfloor\frac{|\mathbb{F}|}{2^l}\rfloor$ or (b) $\lfloor\frac{|\mathbb{F}|}{2^l}\rfloor + 1$. Case (a): $|\frac{|\mathbb{F}_b|}{|\mathbb{F}|} - \frac{1}{|R|}| = |\frac{\lfloor|\mathbb{F}|/2^l\rfloor}{|\mathbb{F}|} - \frac{1}{2^l}|$ which is easy to see that is negligible in $\log|\mathbb{F}|$. Case (b) is similar.

(3) Straightforward as we assume that $\mathbb{F}$ is a field of prime order. ■

We note that the exclusion of the $\log\log|\mathbb{F}|$ most significant bits from the item (1) above is independent of our treatment as depending on the order of the field they may be easy to guess, and as a result BIT_l might not be balanced. Note that if the finite field is chosen appropriately all bits of $s_X(w)$ will be hard: e.g. if we restrict to finite fields $\mathbb{F}$ such that there is a $c \in \mathbb{N}$: $|\mathbb{F}| - 2^{\lfloor\log|\mathbb{F}|\rfloor} \leq (\log|\mathbb{F}|)^c$ then all bits will be hard (e.g. a field of numbers modulo a Mersenne prime):

Corollary 29 *Under the* DPR*-Assumption with parameters* $[n, k-1, t]$, *predicting any bit in a point of the graph of the solution polynomial of a* $\mathrm{PR}[n,k,t]$ *instance is hard.*

A natural question to ask at this point is whether simultaneously more than one point of the polynomial solution enjoys the hardness of extraction properties showed in

theorem 25. In particular we can extend the definition of leaking partial information to many points at the same time as follows:

Definition 30 *Fix some* $w_1, \ldots, w_h \in \mathbb{F}$ *with* $h \in \{1, \ldots, k-1\}$. *We say that* PR$[n, k, t]$ *leaks no partial information for* h *points simultaneously if for all poly-time computable* $g : \mathbb{F}^h \to R$ *and all polynomial-time samplable probability distributions* $\mathcal{D}_h$ *over* $\mathbb{F}^h$ *it holds: for all* PPT $\mathcal{A}$ *there exists a* PPT $\mathcal{A}'$ *such that the following is negligible in* n*:*

$$\left| \mathbf{Prob}_{r \in_U \mathcal{R}; X \in_{(\mathcal{D}_h^{w_1, \ldots, w_h})} \mathcal{S}_{n,k,t}} [\mathcal{A}(r, X) = g(s_X(w_1), \ldots, s_X(w_h))] \right.$$

$$\left. - \mathbf{Prob}_{r' \in_U \mathcal{R}'; \mathbf{u} \in \mathcal{D}_h \mathbb{F}^h} [\mathcal{A}'(r') = g(\mathbf{u})] \right|$$

By choosing the appropriate parameters for the DPR assumption it is possible to show hardness of partial information extraction even in this extended setting:

Theorem 31 *Suppose that there is a poly-time computable* $g : \mathbb{F}^h \to R$ *and a probability distribution* $\mathcal{D}_h$ *for which* PR$[n, k, t]$ *leaks partial information for* h *points simultaneously. Then the DPR-Assumption fails for parameters* $[n, k-h, t]$.

Proof. The proof of the theorem is a straightforward multidimensional extension of the proof of lemma 24. ■

3.3 Pseudorandomness

In this section we will show that distinguishing instances of PR$[n, k, t]$ from random elements of $\mathcal{S}_n := (\mathbb{F})_n \times \mathbb{F}^n$ is hard under the DPR-Assumption (which essentially amounts to saying that instances of PR$[n, k, t]$ are pseudorandom under the DPR). Recall the definition of indistinguishability:

Definition 32 *Let $\{\mathcal{F}_n\}_{n\in\mathbf{N}}$ be a family of sets, such that $\mathcal{F}_n$ contains* all *possible choices of elements of size n. Two families of sets with $A_n, B_n \subseteq \mathcal{F}_n$ are* (polynomial-time, computationally) indistinguishable *if for any* PPT *predicate $\mathcal{A}$,*

$$|\ \mathbf{Prob}_{r\in_U\mathcal{R};X\in_U A_n}[\mathcal{A}(r,X)=1]-\mathbf{Prob}_{r\in_U\mathcal{R};X\in_U B_n}[\mathcal{A}(r,X)=1]\ |$$

is negligible in n. If on the other hand there is an $\mathcal{A}$ for which the probability above is non-negligible in n, we will say that $\mathcal{A}$ is a distinguisher *for A_n, B_n. A family of sets A_n is called* pseudorandom *if it is indistinguishable from $\mathcal{F}_n$.*

Note that for this section we consider $B_n = \mathcal{F}_n := \mathcal{S}_n = (\mathbb{F})_n \times \mathbb{F}^n$ and $A_n := \mathcal{S}_{n,k,t}$ (the set of PR$[n,k,t]$ instances). Let $\mathcal{A}$ be a distinguisher for $\mathcal{S}_{n,k,t}$ and $\mathcal{S}_n$. Because of lemma 18 it holds that the particular choice of the index-solution set I is independent of the distinguishing probability, i.e. for all $I \subseteq \{1,\ldots,n\}$, $|I| = t$, it holds that the following is non-negligible in n:

$$|\ \mathbf{Prob}_{r\in_U\mathcal{R};X\in_U\mathcal{S}_{n,k,t}(I)}[\mathcal{A}(r,X)=1]-\mathbf{Prob}_{r\in_U\mathcal{R};X\in_U\mathcal{S}_n}[\mathcal{A}(r,X)=1]\ |$$

In other words lemma 18 suggests that any distinguisher between $\mathcal{S}_{n,k,t}$ and $\mathcal{S}_n$ also serves as a distinguisher between $\mathcal{S}_{n,k,t}(I)$ and $\mathcal{S}_n$ for all subsets I.

The core of the pseudorandomness proof is the next lemma that given such distinguisher it shows how to extract a parameterized over $\{0,\ldots,n\}$ predicate $\mathcal{B}$ that its behavior is sensitive to some choice of the parameter that belongs in the index-solution-set of the given instance.

Lemma 33 *Let $\mathcal{A}$ be a* PPT *predicate s.t. for all $I \subseteq \{1,\ldots,n\}$ with $|I| = t$, $\mathcal{A}$ is a distinguisher for $\mathcal{S}_{n,k,t}(I)$ and $\mathcal{S}_n$. Then there exists an* PPT *$\mathcal{B}$, for which it holds that for all $I \subseteq \{1,\ldots,n\}$ with $|I| = t$, there exists a $i_0 \in I$ with $i_0 \leq n-k$, such that if*

$$\beta_i(n) := \mathbf{Prob}_{X\in_U\mathcal{S}_{n,k,t}(I);\rho\in_U\mathcal{R}}[\mathcal{B}(i,\rho,X)=1] \quad \text{for} \quad i\in\{0,\ldots,n\}$$

it holds that $|\beta_{i-1}(n) - \beta_i(n)|$ *is negligible for any* $i \notin I$ *and non-negligible for* i_0.

Proof. Let $\mathcal{R}$ be the set of random strings used by the distinguisher $\mathcal{A}$. $\mathcal{B}$ is the following algorithm: given y, r, i, X where $i \in \{0, \ldots, n\}$ and $y \in \mathbb{F}^n$ substitute the first i y-positions of $X = \{\langle z_i, y_i \rangle\}_{i=1}^n$ by the first i values of y; denote this partially randomized instance by $J(i, X, y)$. Then $\mathcal{B}$ simulates $\mathcal{A}$ on input r and $J(i, X, y)$. Note that the randomness used by $\mathcal{B}$ is $\rho := \langle r, y \rangle \in \Re$ where $\Re := \mathcal{R} \times \mathbb{F}^n$.

Define the probabilities $\alpha_1(n) := \mathbf{Prob}_{r \in_U \mathcal{R}; X \in_U \mathcal{S}_{n,k,t}(I)}[\mathcal{A}(r, X) = 1]$ and $\alpha_2(n) := \mathbf{Prob}_{r \in_U \mathcal{R}; X \in_U \mathcal{S}_n}[\mathcal{A}(r, X) = 1]$. Define the following sets:

- $C_i := \{\langle r, y, X \rangle \mid \mathcal{A}(r, Y) = 1; r \in \mathcal{R}; y \in \mathbb{F}^n; X \in \mathcal{S}_{n,k,t}(I); Y = J(i, X, y)\}$
- $V_1 := \{\langle r, X \rangle \mid \mathcal{A}(r, X) = 1; r \in \mathcal{R}; X \in \mathcal{S}_{n,k,t}(I)\}$
- $V_2 := \{\langle r, X \rangle \mid \mathcal{A}(r, X) = 1; r \in \mathcal{R}; X \in \mathcal{S}_n\}$

It is easy to see that $\beta_i(n) = \frac{\#C_i}{\#\mathcal{S}_{n,k,t}(I) \times \mathcal{R} \times \mathbb{F}^n}$; $\alpha_1(n) = \frac{\#V_1}{\#\mathcal{S}_{n,k,t}(I) \times \mathcal{R}}$ and that $\alpha_2(n) = \frac{\#V_2}{\#\mathcal{S}_n \times \mathcal{R}}$. Moreover from the lemma's hypothesis we know that $|\alpha_1(n) - \alpha_2(n)|$ is non-negligible.

Consider C_0; it is immediate that $\#C_0 = |\mathbb{F}|^n \#V_1$ and as a result $\beta_0(n) = \alpha_1(n)$.

Consider C_{n-k}; let $h := |I \cap \{n-k+1, \ldots, n\}|$. obviously it holds that $h \in \{0, \ldots, k\}$. Let $Y := \{\langle z_i, y_i \rangle\}_{i=1}^n \in \mathcal{S}_n$. It is not difficult to show that Y has $|\mathbb{F}|^{n-t+k}$ pre-images under $J(n-k, \cdot, \cdot)$. It follows that,

$$\beta_{n-k}(n) = \frac{\#C_{n-k}}{\#\mathcal{S}_{n,k,t}(I) \times \mathcal{R} \times \mathbb{F}^n} = \frac{|\mathbb{F}|^{n-t+k} \#V_2}{\#\mathcal{S}_{n,k,t}(I) \times \mathcal{R} \times \mathbb{F}^n} = \alpha_2(n)$$

We conclude that $|\beta_0(n) - \beta_{n-k}(n)|$ is non-negligible. This means that there has to be an $i_0 \in \{1, \ldots, n-k\}$ such that $|\beta_{i_0-1}(n) - \beta_{i_0}(n)|$ is non-negligible (using the triangular inequality).

To complete the proof we show that when $i \notin I$ it holds that $|\beta_{i-1}(n) - \beta_i(n)|$ is negligible.

Fix $i \notin I$. Let y^- denote a $\mathbb{F}^n$ vector with its i-th position "blank" (so essentially a $\mathbb{F}^{n-1}$ vector); in a similar manner define X^- to be an instance of $\mathcal{S}_{n,k,t}(I)$ with its i-th y-position "blank". Denote by $y^-_{[v]}$ the $\mathbb{F}^n$ vector that has v "filled" in its i-th position. Similarly define $X^-_{[v]}$.

Let $V_{r,y^-,X^-} := \{v \mid \mathcal{A}(r, y^-/v/X^-) = 1\}$, where the notation $y^-/v/X^-$ stands for an element Y of $\mathcal{S}_n$ s.t. its y-part is comprised of the first $i-1$ elements of y^-, followed by v, followed by the $n-i$ final elements of the y-part of X^-, and $z(Y) = z(X^-)$ (where $z(\cdot)$ denotes the z-elements of a PR instance). Any $v \in V_{r,y^-,X^-}$ together with $\langle r, y^-, X^-\rangle$ can be extended to:

- $|\mathbb{F}|$ tuples $\langle r, y^-_{[v']}, X^-_{[v]}\rangle \in C_{i-1}$ — the fact that there are $|\mathbb{F}|$ tuples follows from the free choice of v'.
- $|\mathbb{F}|$ tuples $\langle r, y^-_{[v]}, X^-_{[v']}\rangle \in C_i$, (recall: $i \notin I$) — the fact that there are $|\mathbb{F}|$ tuples follows from the free choice of v'.

It follows:

$$\#C_i = |\mathbb{F}| \sum_{\langle r,y^-,X^-\rangle} \#V_{r,y^-,X^-} = \#C_{i-1}$$

as a result $\mathcal{J}_{i-1}(n) = \mathcal{J}_i(n)$. ■

Now observe that if $\mathcal{A}_1(i, r, X) := \mathcal{B}(i, r, X)$ and $\mathcal{A}_2(i, r, X) := \mathcal{B}(i-1, r, X)$, it follows easily that $\mathcal{A}_1, \mathcal{A}_2$ is a gap-predicate-pair. As a result,

Theorem 34 *Under the* DPR-*Assumption for* $[n, k, t]$, *the set of instances* $\mathcal{S}_{n,k,t}$ *is pseudorandom.*

Remark. In fact it is possible to assume Pseudorandomness of PR-instances and prove all the results of section 3.2. So, Pseudorandomness of PR-instances as a computational problem stands in between DPR and Hardness of Partial Information Extraction.

Chapter 4

Basic Cryptographic Primitives based on PR

4.1 Chapter Preface

In this chapter we present some fundamental cryptographic primitives that are based on Polynomial Reconstruction. First we define a one-way function based on Polynomial Reconstruction. Under DPR, we show that our one-way function has very strong partial-information concealment properties that make it suitable for direct usage in designing large-value (super-polynomial in the security parameter) commitment schemes.

Next we deal with the problem of symmetric encryption. Despite the fact that a lot of research efforts have been put to obtaining efficient and secure symmetric encryption schemes, the security treatment frequently relies on heuristics and other "questionable" techniques of enforcing secrecy. The public-key setting has enjoyed much more theoretical investigation with respect to various notions of security. Consider for example the notion of semantic security introduced in [GM84]. Proving that inverting ciphertexts for any (possibly adversarially chosen) plaintext probability distribution is hard under reasonable assumptions is fundamental to the security of any practical cryptosystem. And in practice, semantic security is far more important in

the private-key setting as private-key cryptosystems are usually the ones used for transmission of non-random plaintexts.

Although any public-key scheme can be transformed to a symmetric scheme by merely hiding the public-key, the symmetric-key setting exhibits many intricacies [KY00] with respect to security definitions. For example, the fact that the adversary does not have access to the encryption mechanism by default makes it necessary to consider various types of chosen plaintext attacks (i.e. controlling the ability of the adversary to have access to an encryption oracle). Security investigations in the symmetric encryption setting [Lub96, BDJR97] followed the approaches of the public-key setting (see e.g. [Gol93]) and in many cases revealed very interesting separations [KY00]. For a thorough presentation of security notions in the symmetric encryption setting the reader is referred to [KY00].

Here, we introduce a new stateful cipher based on Polynomial Reconstruction. Our cipher possesses unique properties, which are:

- (i) *Computational perfect secrecy.* Consider the following two attacks against a stateful cryptosystem: an existential attack is a chosen-plaintext attack that reveals an encrypted message whereas a universal attack is a chosen-plaintext attack that reveals the key, and thus all messages (from some point on in a stateful cipher). A cipher for which the two attacks are interreducible is said to satisfy "computational perfect secrecy." This property is motivated by Shannon's early work and was introduced and achieved in the computational sense by the remarkable cryptosystem of Blum and Goldwasser [BG85] (where they show that violating semantic security implies factoring of the composite key).

- (ii) *Short key-size*: this property suggests that the plaintext can be *superpolynomial* in the key-size (the security parameter). This property is important since it saves key space in cases it is considered a scarce resource.

- (iii) *Built-in error correction.*
- (iv) *Forward Secrecy*: this property suggests that if a total security breach occurs at a certain time, this affects the security only of future messages while the previously sent messages remain semantically secure for the perpetrator.

We complete this chapter with section 4.5 where we present a generic treatment of stateful ciphers. In particular we show how one can employ a "pseudorandom expander" to simulate the most fundamental security properties of the one-time pad.

4.2 One-Way Function with Built-in Semantic Security

In this section we present a one-way function based on polynomial reconstruction that acts as a "secure envelope" under the DPR-Assumption and can be used to build commitment schemes. Note that there are generic ways [Gol90, Nao91, HILL99] for obtaining such cryptographic primitives based on the results we presented in sections 3.2 and 3.3, however describing a direct construction with improved concealment properties is interesting in its own right for efficiency and applicability purposes.

Fix some parameters $[n, k, t]$. The probabilistic function $F_{n,k,t} : \mathbb{F}^* \to \mathcal{S}_{n,k,t}$ operates as follows: given $\mathbf{x} \in \mathbb{F}^*$, it samples a random element $Y := \{\langle z_i, y_i \rangle\}_{i=1}^{n}$ of $\mathcal{S}_{n,k,t}$ so that (i) Y has a solution s_X that satisfies $s_X(0) = (\mathbf{x})_0, \ldots, s_X(k-1) = (\mathbf{x})_{k-1}$, and (ii) $\{z_1, \ldots, z_n\} \cap \{0, \ldots, k-1\} = \emptyset$.

We note here that $F_{n,k,t}$ is not an injection as it could be the case that $F_{n,k,t}(\mathbf{x}) = F_{n,k,t}(\mathbf{x}')$ for $\mathbf{x} \neq \mathbf{x}'$. This happens when the randomness selected to engulf the polynomial derived from $\mathbf{x}$ happens to correspond to several points of the graph of the polynomial defined by the vector $\mathbf{x}'$. Nevertheless this means that the PR instance generated by $F_{n,k,t}$ has two distinct solutions something that happens with negligible probability as shown in lemma 17 (given that $\log |\mathbb{F}| \geq 2n$ and $t > k$). As a result

we consider $F_{n,k,t}$ to be an injection for all purposes of definition 4. Nevertheless it is important to point out that some user of $F_{n,k,t}$ may deliberately embed more than one polynomial-solution into the output of the function $F_{n,k,t}$. As a result $F_{n,k,t}$ thought of as an encryption function enjoys a natural "ambiguous commitment" property.

Theorem 35 *Under* DPR$[n,k,t]$ *the function* $F_{n,k,t}$ *is a one-way function.*

Proof. Suppose that there is $\mathcal{A}$ with $\mathbf{Prob}[\mathcal{A}(F_{n,k,t}(\mathbf{x})) = \mathbf{x})]$ non-negligible, where the probability is taken over all $\mathbf{x} \in \mathbb{F}^k$ and the internal coin tosses of $\mathcal{A}$ and $F_{n,k,t}$. Obviously it holds that $\mathcal{A}$ solves the PR with non-negligible probability. Let $\mathcal{A}'$ be a PPT that first permutes the pairs on the input (instance of PR) and then simulates $\mathcal{A}$ on the permuted pairs. It is easy to show (cf. lemma 18) that for all $I \subseteq \{1,\ldots,n\}$, $|I| = t$, $\mathbf{Prob}_{X \in \mathcal{S}_{n,k,t}(I)}[\mathcal{A}'(X) = s_X]$ is non-negligible in n.

Now we show how to use $\mathcal{A}'$ to construct a gap-predicate-pair. Let $\mathcal{B}$ be a PPT that given $X \in \mathcal{S}_{n,k,t}(I)$ and $i \in \{0,\ldots,n\}$ it does the following: first it randomizes the first i y-positions of X and then simulates $\mathcal{A}'$ on this instance. If $\mathcal{A}'$ returns the correct answer (something that is checkable in polynomial-time — a proposed solution for a PR instance can be verified in poly-time), $\mathcal{B}$ returns 1, otherwise $\mathcal{B}$ returns 0.

It is easy to verify that $\mathbf{Prob}_{X \in \mathcal{S}_{n,k,t}(I)}[\mathcal{B}(0,X) = 1]$ is non-negligible function in n, whereas $\mathbf{Prob}_{X \in \mathcal{S}_{n,k,t}(I)}[\mathcal{B}(n-k,X) = 1]$ is negligible function in n since $\mathcal{A}'$ cannot predict a polynomial which has been completely randomized (cf. lemma 33). It follows that

$$|\mathbf{Prob}_{X \in_U \mathcal{S}_{n,k,t}(I)}[\mathcal{B}(0,X) = 1] - \mathbf{Prob}_{X \in_U \mathcal{S}_{n,k,t}(I)}[\mathcal{B}(n-k,X) = 1]|$$

is non-negligible in n and by the triangular inequality it follows that for some $i_0 \in \{1,\ldots,n-k\}$ it holds that

$$|\mathbf{Prob}_{X \in_U \mathcal{S}_{n,k,t}(I)}[\mathcal{B}(i_0-1,X) = 1] - \mathbf{Prob}_{X \in_U \mathcal{S}_{n,k,t}(I)}[\mathcal{B}(i_0,X) = 1]|$$

is non-negligible in n. Using a similar argument as in proof of lemma 33 it can be shown that i_0 should be an element of I. It follows that $\mathcal{A}_1(i,\cdot) := \mathcal{B}(i-1,\cdot)$ and $\mathcal{A}_2(i,\cdot) := \mathcal{B}(i,\cdot)$ constitute a gap-predicate-pair and as a result the Decisional-PR assumption is violated. ■

Based on the results of section 3.2, we draw the following corollary:

Corollary 36 *Under* DPR$[n,k-1,t]$, $F_{n,k,t}$ *is a one-way function so that: if* $g:\mathbb{F}\to R$ *is some computable function, an adversary given* $V_x := F_{n,k,t}(\langle x, r_1,\ldots,r_{k-1}\rangle)$, *with* $r_1,\ldots,r_{k-1}$ *are selected at random over* $\mathbb{F}$, *gains no advantage in computing* $g(x)$ *even if* x *follows an adversarially chosen probability distribution.*

The above corollary suggests that V_x is a "secure envelope" for the value x. In fact it is possible to increase the ratio of concealed information as the following theorem reveals:

Theorem 37 *Let* $h\in\{1,\ldots,k-1\}$. *Under* DPR$[n,k-h,t]$, *the function* $F_{n,k,t}$ *defined on inputs from* $\mathbb{F}^h$, *with the remaining of its input selected at random from* $\mathbb{F}^{k-h}$ *(i.e. given* $x_0,\ldots,x_{h-1}\in\mathbb{F}$, *we compute* $F_{n,k,t}(\langle x_0,\ldots,x_{h-1},r_1,\ldots,r_{k-h}\rangle)$, *where* $r_1,\ldots,r_{k-h}$ *are random elements of* $\mathbb{F}$*), is a* secure envelope *(see definition 5).*

Proof. The proof follows closely the arguments of lemma 24. Let $\mathcal{A}$ be a PPT and $\mathcal{D}_h$ a probability distribution over $\mathbb{F}^h$ for which the commitment of some values $\mathbf{x} := \langle x_0,\ldots,x_h\rangle$ leaks some partial information: i.e. for some poly-time computable function $g:\mathbb{F}^h\to R$, $\mathcal{A}$ computes the value $g(\mathbf{x})$ with non-negligible advantage. As a result and due to lemma 18 we can formulate the success probability of $\mathcal{A}$ as follows: for all $I\subseteq\{1,\ldots,n\}$, $|I|=t$,

$$\alpha(n) := \mathbf{Prob}_{r\in_U\mathcal{R};X\in\mathcal{D}_h\mathcal{S}_{n,k,t}(I)}[\mathcal{A}(r,X) = g(\langle s_X(0),\ldots,s_X(h-1)\rangle)]$$

The proof follows directly from theorem 31. ■

Observe that $F_{n,k,t}$ is a *non-trivial secure envelope*: given the coin-tosses that were used for the generation of a certain output it is easy to see that the input to the function can be computed (by polynomial interpolation since the coin tosses will reveal the error-locations).

An interesting property of the above secure envelope is that the hidden value $\mathbf{x}$ can be superpolynomial size in the security parameter n. This is because the size of $\mathbf{x}$ is proportional to $\log|\mathbb{F}|$ which can be selected to be superpolynomial in the security parameter n without affecting the security of the primitive. The "secure envelope" properties of $F_{n,k,t}$ suggest that the PR-based one-way function can be used directly in the design of commitment schemes. More details about the use of $F_{n,k,t}$ in commitment schemes are presented in the next section.

4.3 Value Commitment

A value commitment scheme involves two players A and B that act in two phases: the commitment phase where A commits to some private input $\mathbf{x}$. The output of this phase denoted by $V_{\mathbf{x}}$ is transmitted to player B. The decommitment or "open" phase where A transmits the decommitment witness U to player B. Player B applies U on $V_{\mathbf{x}}$ (a process that reveals $\mathbf{x}$) and either accepts or rejects the commitment. A commitment scheme should be (i) binding: player A should not be able to "open" $V_{\mathbf{x}}$ to a value $\mathbf{x}' \neq \mathbf{x}$: (ii) hiding: player B should not be able to extract any partial information about $\mathbf{x}$ given $V_{\mathbf{x}}$. A commitment scheme is called "non-interactive", if no interaction is required from the two players (the communication flow is only from player A to player B).

We point here that using generic techniques ([Nao91]) it is possible to derive a PR-based commitment scheme based on our pseudorandomness results of section 3.3. Nevertheless such generic techniques are typically expensive to implement and it is

of interest to pursue more direct designs.

Theorem 37 suggests that the function $F_{n,k,t}$ can be used to commit to an element $\mathbf{x} \in \mathbb{F}^h$ by publishing $V_{\mathbf{x}}$ as the commitment value. The decommitment witness is defined to be the index-solution-set I of $V_{\mathbf{x}}$. This scheme is non-interactive and hiding under the DPR$[n, k-h, t]$. Nevertheless the scheme is not binding for the committer since player A might embed more than one solution in the instance $V_{\mathbf{x}}$ and open one of them at her choice; as a result the scheme applies only to the "honest committer" case. By coupling the PR-based non-binding commitment with a binding commitment scheme we derive a scheme with a unique property:

Commitment with Sublinear Decommitment Witness. Typically in commitment schemes the size of the decommitment witness is of the same size as the committed value (or larger). For example in Pedersen's non-interactive scheme [Ped91], that is based on the discrete-logarithm assumption, the commitment to some $x < Q$ is a value $g^r h^x$ that belongs to $\mathbf{Z}_P^*$ (where $P = 2Q + 1$ with P, Q large primes, and $g, h \in \mathbf{Z}_P^*$ public parameters which are quadratic residues modulo P) and the decommitment information is $\langle r, x \rangle$ (note that $r < Q$ is selected at random). Clearly the size of the decommitment witness is linear in the size of the committed information. In many settings it is of great interest to minimize the size of the decommitment information for private storage space saving.

In the case of PR-based commitment, we can use an alternative commitment scheme with which player A commits to the index-solution-set I of $V_{\mathbf{x}}$. The combined scheme becomes binding. Because of the fact that the size of the committed value $\mathbf{x}$ (which is proportional to $\log |\mathbb{F}|$) can be much larger (even superpolynomially) compared to the size of the index-solution-set (which is n) this turns a binding/hiding commitment to a bitstring of small length (n) to a binding/hiding commitment of a large value of length $\log |\mathbb{F}|$. Note that this does not compromise security since

$\min\{\binom{n}{t}, \binom{n}{k}\}$ (which is the number of steps required for a brute-force attack against PR) can be chosen to be superpolynomial in $\log|\mathbb{F}|$ even if $\log|\mathbb{F}|$ is superpolynomial in n.

Let us instantiate the above using Pedersen's commitment scheme: suppose we want to commit to a value x of size b bits. Using Pedersen's commitment the decommitment witness would be of size $\mathcal{O}(b)$. Instead, we commit to x using the PR-based commitment over a finite field $\mathbb{F}$ with $\log|\mathbb{F}| > b$ by sending the value $F^I_{n,k,t}(x, r_1, \ldots, r_{k-1})$ (where I is the index-solution-set of the output of the PR-based one-way function); additionally we commit to v_I (which stands for a value that describes the set I) by sending $g^r h^{v_I}$. The decommitment information is $\langle r, v_I\rangle$ and is of size $\mathcal{O}(n)$. To achieve sublinear decommitment witness size we select the parameter n to be sublinear in the parameter b.

Proposition 38 *The combined commitment scheme described above is hiding, binding and non-interactive under the* $\mathrm{DPR}[n, k-1, t]$ *over a finite field* $\mathbb{F}$*, and the Discrete-Logarithm Assumption over a multiplicative group of element size* n*, and can be used to commit to values of size* $\log|\mathbb{F}| >> n$ *with decommitment witness information of size* $\mathcal{O}(n)$.

Proof. The proof is straightforward from the properties of the Pedersen's commitment scheme and corollary 36. ■

Corollary 39 *The combined commitment scheme supports sublinear decommitment witness size since* n *can be selected sublinear to* $\log|\mathbb{F}|$ *without affecting the security of the scheme (which depends solely on the security parameter* n*).*

4.4 A Secure Stateful-Cipher based on PR

A cipher design involves two parties, who share some common random input (the key). The goal of a cipher design is the secure transmission of a sequence of messages. Suppose that I denotes the shared randomness between the sender and the receiver. A cipher is defined by two probabilistic functions $f_I : \mathcal{K} \times \mathbb{P} \to \mathcal{K} \times \mathbb{C}$ and $g_I : \mathcal{K} \times \mathbb{C} \to \mathcal{K} \times \mathbb{P}$. The spaces $\mathcal{K}, \mathbb{P}, \mathbb{C}$ denote the state-space, plaintext-space and ciphertext-space respectively. The functions f, g have the property that if $f_I(s, m) = (s', c)$ (encryption) it holds that $g_I(s, c) = (s', m)$ (decryption): note that s' (given by both f, g) is the state that succeeds the state s.

Stream-ciphers use public state sequences of the form $\langle 0, 1, 2, 3, \ldots \rangle$. The reader is referred to [Lub96] for more details on stream ciphers and how they can be built based on pseudorandom number generators. Block-ciphers encrypt messages of size equal to some fixed security parameter which are called blocks. Such ciphers are typically at the same state throughout and this state is considered to be secret (it coincides with the secret shared random key). The reader is referred to [Gol98] for further details on block-ciphers and generic constructions.

If a cipher, which operates on blocks, employs a "secret state-sequence update" and uses the shared randomness (the key) only as the initial state of the state-sequence, it is called a *stateful* cipher, see figure 4.1: (note that in a stateful cipher we suppress the subscript I from the functions f, g).

In the remaining of this section we introduce a stateful cipher that is based on PR and possesses unique properties. We will revisit ciphers in more detail in section 4.5.2.

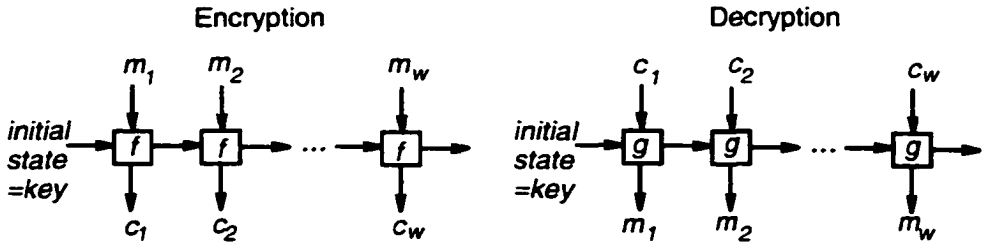

Figure 4.1: A Stateful Cipher

4.4.1 Description of the PR-Cipher

Let $[n, \frac{k-1}{2}, t]$ with $k \leq t$ be sound parameters for the PR problem. We work in a finite field $\mathbb{F}$ with $\log |\mathbb{F}| \geq 3n$. The state-space $\mathcal{K}$ is defined to be the set of n-bitstrings with Hamming weight t. For some $s \in \mathcal{K}$ we define I_s to be the corresponding subset of $\{1, \ldots, n\}$, and v_s be the corresponding integer that has s as its binary representation. We denote by $V_{\mathcal{K}}$ the set of all numbers that their binary representation belongs in $\mathcal{K}$. Let $\mathbb{P} := \mathbb{F}^{\frac{k-1}{2}}$ and $\mathbb{C} := (\mathbb{F})_n \times \mathbb{F}^n$. The shared randomness between the two parties is a random $s_0 \in \mathcal{K}$, that is the initial state of the cipher. The encryption function of the cipher is defined as follows

$$f(s, \mathbf{m}) := F^{I_s}_{n,k,t}(\langle s', (\mathbf{m})_1, \ldots, (\mathbf{m})_{\frac{k-1}{2}}, r_1, \ldots, r_{\frac{k-1}{2}} \rangle)$$

where $F^{I_s}_{n,k,t}$ is the PR-based one-way function of section 4.2 so that index-solution-set of the output of $F_{n,k,t}$ is set to I_s; $r_1, \ldots, r_{\frac{k-1}{2}}$ are random elements of $\mathbb{F}$, and s' is a random element of $V_{\mathcal{K}}$. The decryption function g is defined as follows: given $\langle s, C \rangle \in \mathcal{K} \times \mathbb{C}$, the polynomial p that corresponds to the pairs of C whose index is in I_s is interpolated. The decrypted message is set to be $\langle p(1), \ldots, p(\frac{k-1}{2}) \rangle$ and the next state is set to the binary representation of $p(0)$.

4.4.2 Semantic-Security

A semantic-security breaking adversary $\mathcal{A}$ for a stateful cipher is a PPT that takes the following steps: (i) queries a polynomial number of times the encryption-mechanism, (ii) generates two messages M_1, M_2 and obtains the ciphertext that corresponds to the encryption of M_b where b is selected at random from $\{1,2\}$, (iii) queries the encryption-mechanism a polynomial number of times. Finally the adversary predicts the value of b with probability substantially better than $1/2$. This is illustrated in figure 4.2. A cipher is said to be semantically secure if any semantic-security breaking adversary predicts b with negligible advantage in the security parameter n. For more details regarding semantically secure symmetric encryption, see [Lub96, KY00].

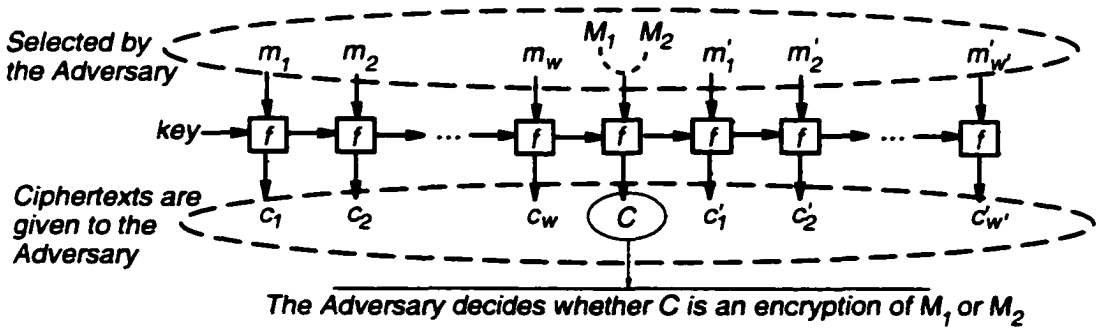

Figure 4.2: Semantic Security Adversary for the PR-cipher

More formally semantic security in the context of stateful ciphers is defined as follows:

Definition 40 *Let $\mathcal{O}^b$, with $b \in \{1,2\}$ be an encryption oracle for a stateful cipher initialized to a random initial state that accepts two kinds of input: (i) a plaintext, where $\mathcal{O}^b$ returns its encryption under the current state, (ii) a pair of plaintexts M_1, M_2, where $\mathcal{O}^b$ returns the encryption of M_b (such input is allowed only once). A*

semantic security breaking adversary is a PPT $\mathcal{A}$ *that given oracle access to* $\mathcal{O}^b$ *it predicts* b *with probability substantially better than* $1/2$*, i.e. the distance*

$$| \mathbf{Prob}_{b\in_U\{1,2\}}[\mathcal{A}^{\mathcal{O}^b}(1^n) = b] - \frac{1}{2} |$$

is non-negligible in n*, where the probability is taken over all internal coin-tosses of* $\mathcal{O}^b$ *and* $\mathcal{A}$ *and all possible initial states for the cipher. If, for a certain cipher, there do not exist semantic security breaking adversaries then we say that the cipher is semantically secure.*

We remark that the two kinds of input to the encryption oracle define three stages of adversarial action, namely (i) querying the encryption oracle a number of times, (ii) submitting the "challenge" (the pair of plaintexts of which the adversary receives the encryption of one of the two at random), and (iii) querying the encryption oracle a number of times before guessing which of the two plaintexts of the challenge was encrypted. We proceed to show that the PR-Cipher is semantically secure under the Decisional PR-Assumption, specifically:

Theorem 41 *The PR-Cipher is semantically secure under* DPR$[n, \frac{k-1}{2}, t]$.

Proof. We start with a definition: we denote by $\mathcal{L}^{(u)}_{n,k,t}[\mathbf{m}_1, \ldots, \mathbf{m}_u]$ the output of an encryption oracle of the PR-cipher when accessed by an semantic security adversary. In other words it is the space of sequences of $\mathcal{S}_{n,k,t}$ instances $X_1, \ldots, X_u$ so that $\mathbf{m}_j := \langle s_{X_j}(1), \ldots, s_{X_j}(\frac{k-1}{2})\rangle$ and so that the binary representation of $s_{X_j}(0)$ corresponds to the characteristic string of $I(X_{j+1})$, for $j = 1, \ldots, u-1$. For two families of sets A_n and B_n we write $A_n \approx B_n$ if they are polynomial-time indistinguishable (see definition 32).

Claim 1. For any $u \geq 1$, $L^{(u)}_{n,k,t}[\mathbf{m}_1, \ldots, \mathbf{m}_u] \approx (S_n) \times L^{(u-1)}_{n,k,t}[\mathbf{m}_2, \ldots, \mathbf{m}_u]$ unless the DPR with parameters $[n, \frac{k-1}{2}, t]$ fails.

Proof. Suppose the two families are distinguishable by some adversary $\mathcal{A}$ with non-negligible advantage. We will show how to use the adversary to violate the DPR with parameters $[n, \frac{k-1}{2}, t]$.

Adaptive Encryption Sampler. The input is $X \in \mathcal{S}_{n,\frac{k-1}{2},t}(I)$ so that $X := \{\langle z_i, y_i\rangle\}_{i=1}^{n}$ and $z_i \notin \{0,\ldots,k-1\}$, and a *sequence* of messages $\mathbf{m}_1,\ldots,\mathbf{m}_u$ (submitted one by one). Let $p'(x)$ be a random polynomial of degree less than k so that (i) $p'(0)$ is a random element so that $p'(0) \leq 2^n$ and the hamming weight of $p'(0)$ is t, and (ii) $p'(i) = (\mathbf{m}_1)_i$ for $i = 1,\ldots,\frac{k-1}{2}$. Consider the instance $X_{m_1} := \{\langle z_i, z_i(z_i-1)\ldots(z_i-\frac{k-1}{2})y_i + p'(z_i)\rangle\}$. Define I_2 to be the subset of $\{1,\ldots,n\}$ so that its characteristic string is identical to the binary representation of $p'(0)$. Next we sample X_{m_2} so that (i) $\langle s_{X_{m_2}}(1),\ldots,s_{X_{m_2}}(\frac{k-1}{2})\rangle = \mathbf{m}_2$, and (ii) the characteristic string of $I(X_{m_2})$ is identical to the binary representation of $s_{X_{m_1}}(0)$. Continuing in a similar manner we construct adaptively the instances $X_{m_3},\ldots,X_{m_u}$. It is clear that this series of samples is uniformly distributed over $L^{(u)}_{n,k,t}[\mathbf{m}_1,\ldots,\mathbf{m}_u]$.

Now suppose that the above sampling method is also given a parameter $i \in \{0,\ldots,n-\frac{k-1}{2}\}$ and the sampler randomizes the first i positions of X_{m_1}.

Now consider the predicates: $\mathcal{A}_1$ that simulates $\mathcal{A}$ using the adaptive encryption sampler to simulate the encryption oracle with parameter i, and $\mathcal{A}_2$ that simulates $\mathcal{A}$ using the adaptive encryption sampler to simulate the encryption oracle with parameter $i-1$. Following similar arguments as in the proof of lemma 24 one can see that $\mathcal{A}_1, \mathcal{A}_2$ constitute a gap-predicate-pair with parameters $[n, \frac{k-1}{2}, t]$. ■

Claim 2. $L^{(u)}_{n,k,t}[\mathbf{m}_1,\ldots,\mathbf{m}_u] \approx (\mathcal{S}_n)^u$ unless the DPR with parameters $[n, \frac{k-1}{2}, t]$ fails.

Proof. Suppose that there is a distinguisher $\mathcal{A}$ between the two families (the "extreme hybrids"). Then by the triangular inequality $\mathcal{A}$ can distinguish between two "neighboring hybrids" i.e.

$$(\mathcal{S}_n)^v \times L^{(u-v)}_{n,k,t}[\mathbf{m}_{u-v},\ldots,\mathbf{m}_u] \not\approx (\mathcal{S}_n)^{v+1} \times L^{(u-v-1)}_{n,k,t}[\mathbf{m}_{u-v-1},\ldots,\mathbf{m}_u]$$

for some $v \in \{0, \ldots, u-1\}$. Based on claim 1 this contradicts the DPR with parameters $[n, \frac{k-1}{2}, t]$. ■

(*Proof* of theorem 41) Suppose now that $\mathcal{A}$ is a semantic security breaking adversary for the PR-cipher. Consider a predicate $\mathcal{B}$ that operates as follows:

$\mathcal{B}$ receives as input $i \in \{0, \ldots, n - \frac{k-1}{2}\}$ and $X \in \mathcal{S}_{n,\frac{k-1}{2},t}$ and communicates with the adversary $\mathcal{A}$ (refer to figure 2 that presents the operation of the adversary). In the first w queries to the adversary, $\mathcal{B}$ replies by random samples of $\mathcal{S}_n$. The adversary cannot detect the difference as the results of claim 2 reveal. When the adversary submits M_1, M_2, $\mathcal{B}$ selects $b \in \{1, 2\}$ at random and using X, samples an encryption of M_b denoted by X_{M_b} using the technique described in the adaptive encryption sampler of the proof of claim 1. Subsequently $\mathcal{B}$ randomizes the first i positions of X_{M_b}. The remaining w' queries of $\mathcal{A}$ are answered by proper encryptions of the messages it submits (something that is possible for $\mathcal{B}$ since it resets the key of the cipher in the construction of X_{M_b}). Finally $\mathcal{B}$ returns 1 if the adversary guesses b correctly or 0 otherwise.

Define the predicate $\mathcal{A}_1 := \mathcal{B}$, and let $\mathcal{A}_2$ be the predicate that simulates $\mathcal{B}$ on input $i-1$ and X. Following similar arguments as in the proof of lemma 24 one can see that $\mathcal{A}_1, \mathcal{A}_2$ constitute a gap-predicate-pair for the parameters $[n, \frac{k-1}{2}, t]$ and as a result the DPR is violated. ■

4.4.3 Forward Secrecy

A cipher is said to satisfy forward secrecy if in the case of a total security breach at some point of its operation (i.e. the internal state is revealed) the adversary is unable to extract any information about the previously communicated messages.

This is formalized by two chosen plaintext security adversaries who are submitting adaptively messages to the encryption oracle. The encryption oracle flips a coin

and answers by encrypting the plaintexts submitted by one of the two adversaries (the same adversary throughout). At some point the internal state of the system is revealed to the adversaries. Forward secrecy is violated if the adversaries can tell with probability significantly better than one half whose messages the encryption oracle was returning. More formally.

Definition 42 *Let $\mathcal{O}^b_{\mathrm{fs}}$, with $b \in \{1,2\}$ be an encryption oracle for a stateful cipher initialized to a random initial state that accepts two kinds of input: (i) a pair of plaintexts $\mathbf{m}_1, \mathbf{m}_2$, where $\mathcal{O}^b_{\mathrm{fs}}$ returns the encryption of $\mathbf{m}_b$ under the current state. (ii) a termination message, where $\mathcal{O}^b_{\mathrm{fs}}$ returns the current internal state: no more queries are accepted by $\mathcal{O}^b_{\mathrm{fs}}$ after the termination message is submitted. A forward secrecy breaking adversary is a* PPT $\mathcal{A}$ *that given oracle access to $\mathcal{O}^b_{\mathrm{fs}}$ it predicts b with probability substantially better than $1/2$, i.e. the distance*

$$\left| \mathbf{Prob}_{b \in_U \{1,2\}}[\mathcal{A}^{\mathcal{O}^b_{\mathrm{fs}}}(1^n) = b] - \frac{1}{2} \right|$$

is non-negligible in n, where the probability is taken over all internal coin-tosses of $\mathcal{O}^b_{\mathrm{fs}}$ and $\mathcal{A}$ and all possible initial states for the cipher. If, for a certain cipher, there do not exist forward secrecy breaking adversaries then we say that the cipher satisfies forward secrecy.

The following theorem summarizes the properties of the PR-Cipher:

Theorem 43 *The PR-Cipher satisfies forward secrecy under* $\mathrm{DPR}[n, \frac{k-1}{2}, t]$.

Proof. We denote by $\mathcal{L}^{(u)}_{n,k,t}[\begin{smallmatrix}\mathbf{m}^0_1 \\ \mathbf{m}^1_1\end{smallmatrix}, \ldots, \begin{smallmatrix}\mathbf{m}^0_u \\ \mathbf{m}^1_u\end{smallmatrix}]$ the output of an encryption oracle of the stateful cipher when accessed by the two chosen plaintext adversaries that are part of the forward security attack. In other words it is the space of sequences of $S_{n,k,t}$ instances $X_1, \ldots, X_u$ so that $\langle s_{X_j}(1), \ldots, s_{X_j}(\frac{k-1}{2}) \rangle = \mathbf{m}^b_j$ for all $j = 1, \ldots, u$ where b

is a random coin toss: the binary representation of $s_{X_j}(0)$ corresponds to the characteristic string of $I(X_{j+1})$, for $j = 1, \ldots, u-1$.

Claim 3. For any $u \geq 1$, $\mathcal{L}^{(u)}_{n,k,t}[\begin{smallmatrix}\mathbf{m}_1^0 \\ \mathbf{m}_1^1\end{smallmatrix}, \ldots, \begin{smallmatrix}\mathbf{m}_u^0 \\ \mathbf{m}_u^1\end{smallmatrix}] \approx (S_n) \times \mathcal{L}^{(u-1)}_{n,k,t}[\begin{smallmatrix}\mathbf{m}_2^0 \\ \mathbf{m}_2^1\end{smallmatrix}, \ldots, \begin{smallmatrix}\mathbf{m}_u^0 \\ \mathbf{m}_u^1\end{smallmatrix}]$ unless the DPR with parameters $[n, \frac{k-1}{2}, t]$ fails.

The arguments of the proof of claim 3 are similar to those of the proof of claim 1 of theorem 41.

Claim 4. $\mathcal{L}^{(u)}_{n,k,t}[\begin{smallmatrix}\mathbf{m}_1^0 \\ \mathbf{m}_1^1\end{smallmatrix}, \ldots, \begin{smallmatrix}\mathbf{m}_u^0 \\ \mathbf{m}_u^1\end{smallmatrix}] \approx (S_n)^u$ unless the DPR with parameters $[n, \frac{k-1}{2}, t]$ fails.

Again, this is shown using the same argument as in the proof of claim 2 of theorem 41.

Now the result follows easily since: the output of the encryption oracle is indistinguishable for the choice of $b \in \{1,2\}$ provided that the DPR with parameters $[n, \frac{k-1}{2}, t]$ holds. This implies in a straightforward manner that the adversary cannot predict b with probability significantly better than $1/2$. ■

4.4.4 Computational Perfect Secrecy

A generic chosen plaintext adversary for a stateful cipher is defined as follows:

Definition 44 *Let $\mathcal{O}$ be an encryption oracle for a stateful cipher that is initialized to a random initial state; given a plaintext, $\mathcal{O}$ returns its encryption under the current state. A generic chosen plaintext adversary is a* PPT *$\mathcal{A}$ that is given oracle access to $\mathcal{O}$.*

For some stateful-cipher we consider the following two attacks that can be launched by a generic chosen plaintext adversary: (i) "existential" where the generic chosen plaintext adversary is allowed to query the encryption oracle a number of times and then is asked to decrypt the next ciphertext (which encrypts a random secret message) (ii) "universal" where a generic chosen plaintext adversary is allowed to query

the encryption oracle a number of times and then is asked to recover the state of the cipher (something that allows the recovery of all future messages from that point on).

It is clear that for any cipher an existential attack reduces to a universal attack. Nevertheless it is not at all apparent if the opposite direction in the reduction holds.

Definition 45 *A stateful-cipher for which it holds that a generic chosen plaintext adversary launching an existential attack implies the existence of a generic chosen plaintext adversary launching a universal attack is said to satisfy* computational perfect secrecy.

The equivalence of attacks that recover the message to attacks that recover the key has been postulated by Shannon as "perfect secrecy." Blum and Goldwasser [BG85] designed a factoring based public-key system where they reduced semantic security of a message to breaking the key (i.e. factoring the composite). They coined the notion of "computational perfect secrecy," a variant of which we define above.

Theorem 46 *The PR-Cipher satisfies computational perfect secrecy.*

Proof. Suppose that it is possible to launch an existential attack with u queries to the encryption mechanism. We show how to launch a universal attack: first we make $(u+1)$-queries to the encryption mechanism so we have the plaintext-ciphertext pairs $\langle M_1, C_1\rangle, \ldots, \langle M_{u+1}, C_{u+1}\rangle$ where $M_1, \ldots, M_u$ are chosen following the query algorithm of the existential attack algorithm and M_{u+1} is chosen at random. Suppose that $C_{u+1} := \{\langle z_i, y_i\rangle\}_{i=1}^{n}$. We compute $X' := \{\langle z_i + 1, y_i\rangle\}_{i=1}^{n}$, and we feed X' to the existential attack algorithm to obtain the "message" $\langle a_1, \ldots, a_{\frac{k-1}{2}}\rangle$ with probability of success α. Observe that $s_{X'}(x) = s_X(x-1)$ and as a result $a_1 = s_{X'}(1) = s_X(0)$. It follows that the binary representation of a_1 is the characteristic string of the next key (for the $(u+2)$-th encryption of the cipher). As a result the universal attack reduces to an existential attack with the same probability of success. ■

4.4.5 Superpolynomial Message-Size

A cryptosystem that has this property allows the plaintext size to be superpolynomial in the key-size, or in other words, it allows the key-size to be substantially shorter (inverse-super-polynomial) in the size of messages.

This property allows much saving in the storage of the shared key which can be an expensive resource in many settings. Additionally, it can be particularly useful in settings where we want to extract a key from a small amount of information (such as key-extraction from biometric data, see e.g. [MRLW02]).

In the PR-Cipher the plaintext size is $\frac{k-1}{2}\lfloor \log |\mathbb{F}| \rfloor$ and can be superpolynomial in the security parameter since $\log |\mathbb{F}|$ can be chosen to be superpolynomial in the security parameter n without affecting the security of the cryptosystem. This is because a brute-force attack against PR requires $\min\{\binom{n}{t}, \binom{n}{k}\}$ steps worst-case and this quantity can be selected to be superpolynomial in $\log |\mathbb{F}|$ even if $\log |\mathbb{F}|$ is superpolynomial in n.

4.4.6 Error-Correcting Decryption

A cryptosystem is said to allow error-correcting decryption if the decryption procedure is able to correct errors that are introduced during the transmission (possibly by an adversary). This combines the decryption operation with the error-correction operation (that is important to apply independently in any setting where two parties communicate).

A cryptosystem that transmits plaintext blocks of size d is called d'-error-correcting if up to d' corrupted blocks can be corrected for each transmitted ciphertext. The PR-cipher (which transmits plaintext blocks of size $\frac{k-1}{2}$ over the underling finite field $\mathbb{F}$ in each ciphertext) is $\frac{t-k}{2}$-error-correcting since the interpolation step during decryption can be substituted by the [BW86] polynomial-reconstruction algorithm that

can withstand up to $\frac{t-k}{2}$ errors (in the worst-case).

4.4.7 Key-Equivalence

A symmetric cryptosystem is said to satisfy the key-equivalence property if there are no families of keys of measurable size that are susceptible to attacks that do not apply to the key-space in general. By "measurable-size" we mean that the ratio of the size of the family of keys over the key-space size is a non-negligible function. More formally,

Definition 47 *Let $\mathcal{K}_n$ denote the key-space of a cipher, where n denotes the security parameter. Let $\mathcal{A}$ be a* PPT *(thought of as a generic adversary) that takes as input a sequence of ciphertexts s_κ as transmitted over the public channel by the sender to the receiver who share a secret-key κ. The cipher satisfies the key-equivalence property if there exists a* PPT *$\mathcal{A}'$ s.t. for any family of keys $\mathcal{K}'_n \subseteq \mathcal{K}_n$ of measurable size: $(\#\mathcal{K}'_n/\#\mathcal{K}_n)$ is non-negligible in n, it holds that for all v in the range of $\mathcal{A}$, the distance*

$$|\ \mathbf{Prob}_{\kappa \in_U \mathcal{K}'_n}[\mathcal{A}(s_\kappa) = v] - \mathbf{Prob}_{\kappa \in_U \mathcal{K}_n}[\mathcal{A}'(s_\kappa) = v]\ |$$

is negligible in n, where the probability is taken over the coin-tosses of $\mathcal{A}, \mathcal{A}'$ and the coin tosses of the sender who generates the sequence of ciphertexts. Intuitively this suggests that an attack of any kind against the cipher over a certain family of keys, can be generalized to an attack against the cipher over the whole key-space. Note that v is possibly a function of s_κ.

The key-equivalence property is an important security aspect for a symmetric cryptosystem as it suggests that there are no "weak" keys.

Proposition 48 *The PR-Based Stateful Cipher satisfies the key-equivalence property.*

Proof. This can be seen easily as a corollary of lemma 18 and the fact that the key-space for the PR-based stateful cipher is defined to be the set of all subsets of $\{1.....n\}$ of size t. ■

4.5 Generic Stateful Ciphers: Emulation of the One-time Pad with Private Randomness

Provable security analysis of symmetric key encryption formally shows security properties of cryptographic designs, capturing desired generic characteristics relevant to practice. In this section we will concentrate on the issue of automatic refreshing of a cipher with a truly random new key, a design we call "stochastic refresh-key cipher." Key-refreshing was a fundamental step in the realization of the PR-cipher of the previous section. Here we investigate this notion separately over a generic design.

From an engineering point of view, refresh-key is an operation by which a cipher is switched to an initial state with a new key. This aspect is quite useful and it may be looked upon as a built-in key management function integrated into the cipher operation. While, in practice, such refresh-key may be done every so often (to save resources such as bandwidth and available true randomness in the system), in the design (mode) which we study refresh-key is performed at each operation: an approach we take in order to study the power and properties of this idea in isolation. Our results provide an affirmative answer to the following question:

The fundamental security properties of the One-time Pad can be preserved, in a two party secure communication setting, when the two parties only share a very limited amount of initial randomness but (i) they do posses a hardware implementation of a pseudorandom number generator that has a small stretching factor (up to a single bit); (ii) large amounts of non-shared local randomness is available to each party.

We analyze various "provable security properties" of our cipher design. Such

analyses have been carried recently regarding various aspects. One such study investigates a "variety of possible challenges" posed to the attacker of symmetric ciphers [BDJR97]. In another study, various notions of types of "adversaries' power and goals" are investigated [KY00]. In the latter, an adversary is allowed various degrees of oracle access to the cipher in the attack (no access, encryption oracle access and decryption oracle access) and it then (or meanwhile) tries to either violate (semantic) security or try to generate related ciphertexts (violate non-malleability). Various separations and equivalences of the 18 various attack combinations have been pointed out for generic probabilistic symmetric encryption.

The provable perfect (information theoretic) security properties of the one-time pad has inspired a lot of security definitions and notions. Here, our attempt is to have a randomized cipher based on two basic hardware assumptions (i) pseudorandom number generator implementation with a limited stretching factor and (ii) unlimited local, non-shared, private randomness, that (computationally) captures, as a design, the most desirable properties of the idealized one-time pad. Our stochastic refresh-key cipher's goal is to emulate the one-time pad with only limited initial shared randomness and a pseudorandom generator with minimal stretching, in contrast with the traditional such emulation via a pseudorandom number generator that requires vast stretching. The shared randomness is replaced by local randomness at the device (which, even though it is a costly resource in practice, is extensively assumed by many key generation and cryptographic-operation designs, especially when achieving provable security).

What are the major provably secure properties of the one-time pad?

- Semantic security against chosen ciphertext and chosen plaintext attack: The attacker is allowed to query the encryption oracle by choosing messages of its own and, in addition, it is allowed to query the decryption oracle by ciphertexts

of its choice. At some point a challenge for security is set: the adversary may challenge with two plaintexts and receive the encryption of one of them at random. Subsequently, further encryption and decryption queries may be allowed. The adversary, in turn, cannot decide which of the two plaintexts was actually encrypted in the challenge phase. This security in the case of the one-time pad is achieved due to the total randomness of the pad portion used in the challenge phase.

- Forward Secrecy: In this attack scenario a total security breach occurs at some point of the cipher operation (i.e. the attacker obtains the contents of the encryption or the decryption device). This should *only* compromise the secure transmissions from this point on and not the previously communicated messages. The concept of forward secrecy can also be useful in the following context: suppose that the sender (alternatively the receiver) wishes to delegate the encryption mechanism to someone else who takes over from some point on. The initial sender wishes that all its previous communication with the receiver is not compromised. In the one-time pad case, as long as used portions of the pad are erased, the compromise of the pad at a point gives forward secrecy since the remaining pad portion gives no advantage in guessing the previous messages.

Our stochastic refresh-key cipher captures these properties. Chosen ciphertext semantic security is a very strong security notion that increases confidence in the usage of an encryption mechanism. Note that in the case of one-time pad, chosen ciphertext semantic security is not different than chosen plaintext semantic security (both only reveal the portion of the pad used for this message/ciphertext). Typically added randomness is required in block-ciphers (random IV, etc.) in order to achieve this property, while stream-ciphers which are pseudorandom (and are required to be

pseudorandom for long stretch of their seed) possess this property. Note that we do not deal with malleability, since one-time pad is not enough for achieving it. (In fact, an "unforgeable cipher design" as in [KY01a] is needed where a MAC is added to the cipher). Regarding forward secrecy, note that it is not possible to achieve it in the typical block-cipher stateless scenario as it requires a stateful design. Even in the stateful scenario it cannot be achieved if the state-sequence generated by the cipher is public, as is the case when a pseudorandom-function is used for encryption with a public counter being the state. In this case the initially shared randomness (function key) will be revealed to a forward secrecy adversary and thus the history will be also revealed. We can therefore see that forward secrecy is a property associated only with stateful ciphers possessing a secret state sequence.

Our stochastic refresh-key cipher will update the state of the cipher at each message transmission with a newly random state. It requires a basic pseudorandom generator only to have a *minimal* stretch factor. Indeed in theory [Lub96], such stretching (of one bit) is enough to have many pseudorandom bits and build stream-ciphers as well as pseudorandom functions and permutations, but the price is typically a substantial reduction of security level as the stretch grows. Such stretching often introduces statistical attacks. If the generator takes a seed (key) of size n and stretches it to $n + k$ bits (for some $k > 0$), then our cipher allows the secure transmission of sequences of k-bit messages. In fact, the design herein can use any k and our working example from now on is $k = n$ (i.e., doubling the seed).

We note that our refresh-key mode which restarts the cipher with a totally random key is different from the re-key mode studied in [AB00], where a key pseudorandomly updates itself (as a typical key-stream in a stream-cipher). The goal in that paper is extending the life of block cipher's key under chosen plaintext attack (only), and they need a large stretch factor of the generator to rekey the block cipher every so often. (In fact, they balance the stretching and the usage of each key in the block

cipher operation.) Their construction cannot take worthwhile advantage of constant stretching in contrast with our construction.

Our analysis can be viewed as analyzing a cipher with a built-in key management operation. In practice, we view that our mode of operation is not going to be performed on every message transmission (in order to save bandwidth). This will result in using a pseudorandom operation (or a stream cipher operation) for a while (a session) between the usage of the mode we suggest to refresh the key. As long as the stretch is within the allowed level of security we can employ this mixed mode. In turn, our construction yields forward secrecy between sessions in this mixed mode case.

We also add to our design a "resynchronization mechanism." Stateful ciphers with secure state (like stream-ciphers) are always vulnerable to loss of synchronization. We show that if one starts using our cipher in a synchronized fashion, then the cipher mechanism can take care of resynchronization while retaining all its security properties.

4.5.1 Preliminaries

Notations. The length of a string $x \in \{0,1\}^*$ is denoted by $|x|$. Given two strings $x, y \in \{0,1\}^*$, $x||y \in \{0,1\}^*$ denotes the concatenation of x, y; if $|x| = |y|$ then $x \oplus y$ denotes the bitwise binary addition (xor) between x, y. Given a $y \in \{0,1\}^k$, $[y]_s$ denotes the s-th bit of y. Given a $y \in \{0,1\}^{2k}$, $\overline{y}$ denotes the leftmost k bits of y whereas $\overline{\overline{y}}$ denotes the rightmost k bits of y; as a result $y = \overline{y}\,||\overline{\overline{y}}$. For some tuple $x := \langle x_1, \ldots, x_n \rangle$, we denote x_i by $\mathcal{P}_i^n(x)$ (the i-th projection). A function $\sigma : \mathbb{N} \to \mathbb{R}$ is called *negligible* if for all $c \in \mathbb{N}$ it holds that $\sigma(n) < n^{-c}$ for sufficiently large n.

All procedures that we consider are poly-time bounded probabilistic algorithms; we use the notation PPT to express "probabilistic polynomial-time" algorithm. Our results can be readily extended to the case of non-uniform adversaries (i.e. when the

adversary is a family of circuits of polynomial depth). All the reductions that we present are uniform.

Definition 49 *A Pseudorandom Number Generator (*PRNG*) is a deterministic function* $G : \{0,1\}^n \to \{0,1\}^{l(n)}$ *with* $l(n) > n$ *for all* n*, that for any* PPT C *with* $C : \{0,1\}^{l(n)} \to \{0,1\}$

$$\mathfrak{adv}_C^G(n) := |\, \mathbf{Prob}_{x \in_U \{0,1\}^n}[C(G(x)) = 1] - \mathbf{Prob}_{y \in_U \{0,1\}^{l(n)}}[C(y) = 1] \,|$$

is negligible in n*.*

The function $\mathfrak{adv}_C^G(n)$ is the advantage that the PPT C has in distinguishing the output of G w.r.t. the uniform distribution of strings over $\{0,1\}^{l(n)}$. A function $G : \{0,1\}^n \to \{0,1\}^{l(n)}$ is *not* a pseudorandom generator if there is a PPT C so that the function $\mathfrak{adv}_C^G(n)$ is not negligible. Such a PPT C will be called a *distinguisher* for G.

4.5.2 Ciphers

For sake of unified notations and treatment we next define ciphers (in more formal terms than usually done). We assume that a cipher involves two parties, the sender and the receiver who share some common random input (the key). The goal of a cipher is the secure transmission of a sequence of messages, i.e. creating a secure channel between the sender and the receiver. The number of total bits to be transmitted is substantially greater than the number of shared random bits n.

Suppose that $x \in \{0,1\}^n$ denotes the shared randomness between the sender and the receiver. A cipher is defined by two probabilistic functions $f_x : \mathcal{K} \times \mathbb{P} \to \mathcal{K} \times \mathbb{C}$ and $g_x : \mathcal{K} \times \mathbb{C} \to \mathcal{K} \times \mathbb{P}$ (typically only the encryption function f_x is probabilistic). The spaces $\mathcal{K}, \mathbb{P}, \mathbb{C}$ denote the state-space, plaintext-space and ciphertext-space respectively. The functions f, g should satisfy the next two conditions: for all $m \in \mathbb{P}$ and $s \in \mathcal{K}$, $\mathcal{P}_2^2(g(s, \mathcal{P}_2^2(f(s,m)))) = m$ and $\mathcal{P}_1^2(f(s,m)) = \mathcal{P}_1^2(g(s, \mathcal{P}_2^2(f(s,m))))$.

The initial state of the cipher is denoted by s_1 and is known by both parties. A "state-sequence" generated for the transmission of $m_1, \ldots, m_k \in \mathbb{P}$ is a tuple $\langle s_1, \ldots, s_k, s_{k+1} \rangle$ so that $\mathcal{P}_1^2(f(s_i, m_i)) = s_{i+1}$ for $i = 1, \ldots, k$. Note that the initial state s_1 can be either public or secret (in which case it depends in the secret shared randomness x). A representation of a cipher is in figure 4.3.

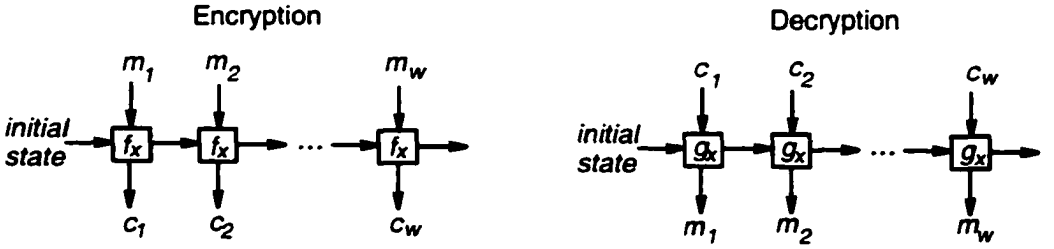

Figure 4.3: A Cipher

Examples of Ciphers.

- Stream-Ciphers. In a typical stream cipher a Linear-Feedback-Shift-Register (LFSR) is used to stretch the shared random key to a key-stream: usually the plaintext space and ciphertext space are defined as $\mathbb{P} = \mathbb{C} = \{0,1\}$ and the state space is $\mathcal{K} := \{0,1\}^n$ where n is the number of the LFSR's "stages." The shared random key constitutes the initialization of the LFSR and the state of the cipher at a certain point is identified to be the contents of the stages of the LFSR (the reader is referred to [MVV97] chapter 6 for a thorough introduction to the subject). The security and efficiency of a stream-cipher depends on the method used to stretch the shared-randomness.

- Block-Ciphers. In a block-cipher messages of size equal to some function $b(n)$ where n is the security parameter are transmitted. In this case $\mathbb{P} := \{0,1\}^{b(n)}$, $\mathbb{C} := \{0,1\}^{c(n)}$ where $c(n)$ is the ciphertext size; note that $c(n) \geq b(n)$ for all n.

The state space is defined as $\mathcal{K} := \{0,1\}^n$. The state-sequence employed by a block cipher is secret and is always constant $\langle x, x, x, \ldots \rangle$ (where x is the shared randomness between the two parties). A detailed introduction to block ciphers is given in [MVV97] chapter 7: the prime examples of block ciphers are DES [Nat99] and the more recent AES [Nat01]. In encrypting messages larger than a block size, modes of operations are defined for block ciphers. Our formalization allows for stateful block-ciphers as well, where the state changes as messages are sent.

4.5.3 Attacks and Security Definitions

Generic Chosen Plaintext Attack. A generic chosen plaintext attack involves an adversary $\mathcal{A}$ (query generator) that yields some output in $\{0,1\}^{q(n)}$. $\mathcal{A}$ is allowed access to an encryption oracle and deduces some information about the cipher's state and/or shared randomness used. This is illustrated in figure 4.4.

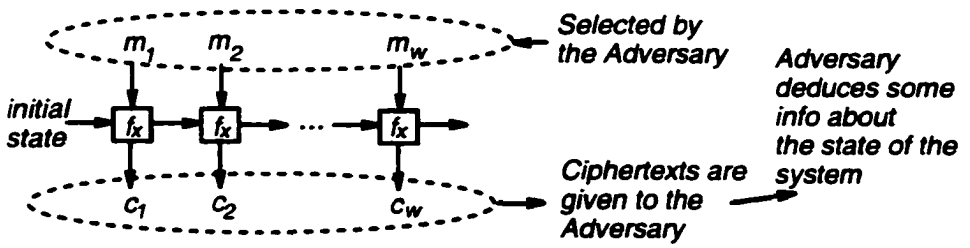

Figure 4.4: A Generic Chosen Plaintext Attack

A cipher is said to be *resilient* against a generic chosen plaintext attack if the encryption oracle is substituted by a probabilistic sampler of $\mathbb{C}$ (that operates independently from the queries of the adversary), this has no effect in the results reported by $\mathcal{A}$. Formally, we define a generic chosen plaintext attack adversary to be a pair of PPT $\mathcal{A}, C$: $\mathcal{A}$ is the query generator and C is a PPT that distinguishes the encryption

oracle's answers from truly random answers given the output of the query generator. The advantage of the adversary is defined as follows:

$$\mathfrak{adv}^{gcpa}_{\mathcal{A},C}(n) := |\ \mathbf{Prob}[C(\mathcal{A}^{\mathcal{E}[x]}(1^n)) = 1] - \mathbf{Prob}[C(\mathcal{A}^{\mathcal{R}}(1^n)) = 1]\ |$$

Definition 50 *A cipher is* resilient *against a generic chosen plaintext attack if for any generic chosen plaintext attack adversary $\mathcal{A}, C$ it holds that $\mathfrak{adv}^{gcpa}_{\mathcal{A},C}(n)$ is negligible in n.*

Another way of formulating the above definition is the following: a cipher is *resilient* against a generic chosen plaintext attack if the functions $\mathcal{A}^{\mathcal{E}[\cdot]}$ and $\mathcal{A}^{\mathcal{R}}$ are indistinguishable where $\mathcal{R}$ is a random sampler of $\mathbb{C}$. Intuitively this means that whatever conclusion the adversary can make about the state of the system and/or the shared randomness used it can also be drawn *without* access to the encryption oracle; as a result any adversary that is allowed a polynomial number of queries to the encryption device cannot deduce something non-trivial about the system.

Chosen Plaintext Semantic Security Attack. A generic chosen plaintext attack attempts to extract information about the state of the system. Although keeping the state of the system as well as the shared randomness secret is important, this is of no use if an adversary can deduce something about the encrypted messages.

A chosen plaintext semantic security adversary $\mathcal{A}$ is a chosen plaintext adversary of special form: $\mathcal{A}$ is allowed to query an encryption oracle like a generic chosen plaintext adversary; then $\mathcal{A}$ submits to the encryption oracle *two* messages M_1, M_2; the oracle selects $b \in \{1,2\}$ and returns to $\mathcal{A}$ the encryption of M_b; finally $\mathcal{A}$ proceeds as a generic chosen plaintext adversary (making further queries to the encryption oracle) and attempts to predict b. We refer to M_1, M_2 as the challenge plaintexts.

The advantage of a chosen plaintext semantic security adversary is defined as follows:

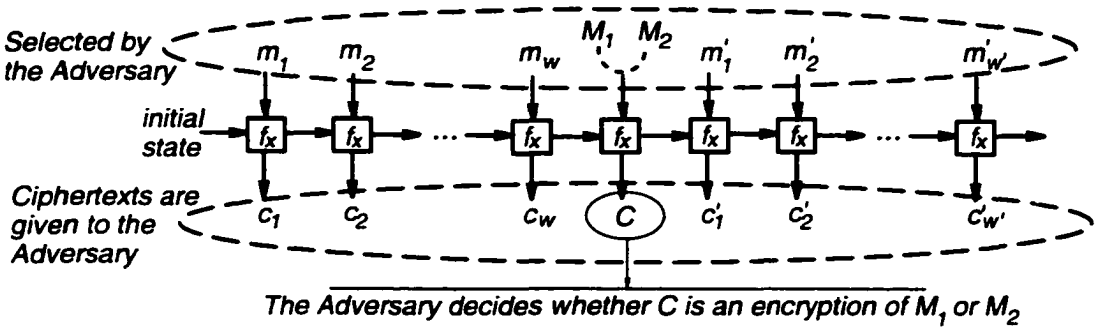

Figure 4.5: A Chosen Plaintext Semantic Security Adversary

$$\mathfrak{adv}_{\mathcal{A}}^{cpa}(n) := \mid \mathbf{Prob}_{x \in_U \{0,1\}^n; b \in_U \{1,2\}}[\mathcal{A}^{\mathcal{E}[x,b]}(1^n) = b] - \frac{1}{2} \mid$$

where the probability is also taken over all internal coin tosses of the encryption oracle, and the coin tosses of $\mathcal{A}$. Note that a cipher might be resilient against a generic chosen-plaintext attack but not semantically secure under chosen plaintext attack.

Definition 51 *A cipher is said to be semantically secure under a chosen plaintext attack if for any chosen plaintext semantic security adversary $\mathcal{A}$ it holds that $\mathfrak{adv}_{\mathcal{A}}^{cpa}(n)$ is negligible in n.*

Chosen Ciphertext Semantic Security Attack. A chosen ciphertext semantic security attack gives to the adversary greater intrusive power compared to a chosen ciphertext semantic security attack. In this case the adversary is capable of querying a *decryption* oracle a number of times before selecting the challenge plaintexts M_1, M_2 and obtain the encryption of M_b, $b \in_U \{1,2\}$. Subsequently after querying a number of times the decryption oracle again the adversary guesses b. Such an adversary is illustrated in figure 4.6. Note that since we are interested in stateful ciphers it is not

necessary to restrict the adversary's queries to the decryption oracle after receiving the encryption of one of the challenge plaintexts.

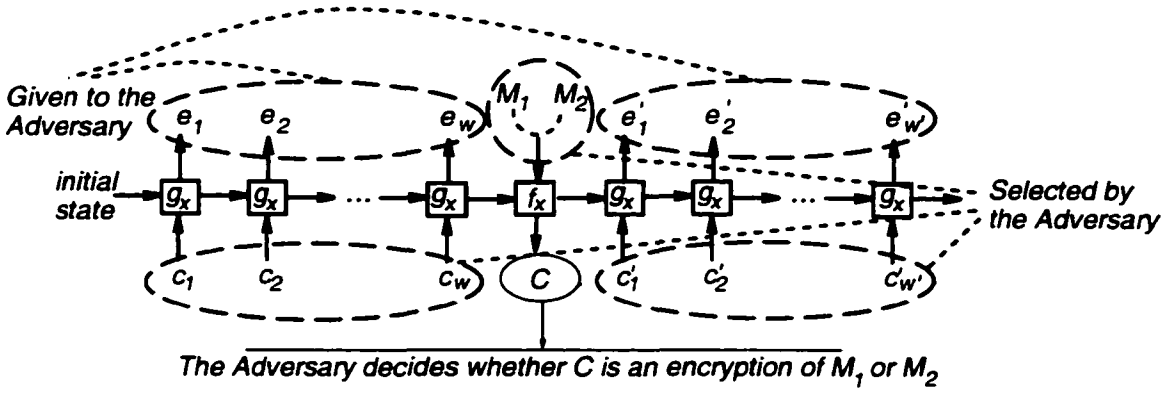

Figure 4.6: A Chosen Ciphertext Semantic Security Adversary

The advantage of a chosen ciphertext semantic security adversary is defined as follows:

$$\mathfrak{adv}^{cca}_{\mathcal{A}}(n) := \mid \mathbf{Prob}_{x\in_U\{0,1\}^n; b\in_U\{1,2\}}[\mathcal{A}^{\mathcal{DE}[x,b]}(1^n) = b] - \frac{1}{2} \mid$$

where the probability is also taken over all internal coin tosses of the encryption oracle (the decryption oracle is thought to be deterministic), and the coin tosses of $\mathcal{A}$.

Definition 52 *A cipher is said to be semantically secure under a chosen ciphertext attack if for any chosen ciphertext semantic security adversary $\mathcal{A}$ it holds that $\mathfrak{adv}^{cca}_{\mathcal{A}}(n)$ is negligible in n.*

Forward Secrecy. The notion of forward secrecy suggests that in the case of a total security breach in some point of a cipher's operation only the security of the future messages is compromised and not the security of the messages exchanged before the breach. This is a very important security notion that significantly increases

the usefulness of a cryptosystem. It additionally allows the delegation of the encryption/decryption device to different parties without compromising the security of the previous messages. Note that forward secrecy does not apply to ciphers that use the history of previously transmitted plaintexts to determine the upcoming encryptions/decryptions. Such ciphers cannot support forward secrecy by definition.

A forward secrecy adversary $\mathcal{A}, C$ operates as follows: The adversary $\mathcal{A}$ (query generator) first selects some k, then adaptively selects k pairs of messages $\langle m_i, m_i' \rangle$ and receives the sequence of encryptions of either $m_1, \ldots, m_k$ or $m_1', \ldots, m_k'$. Subsequently the PPT C is given the output of $\mathcal{A}$ and all information resulting from a total security breach before the encryption/decryption of the $(k+1)$-th message; this includes the current next state of the cipher s_{k+1} and perhaps additional information depending on the definition of f, g such as the shared random key; we denote this information by config_{k+1}. C given config_{k+1} and the output of $\mathcal{A}$ attempts to distinguish whether the k given ciphertexts are encryptions of $\langle m_1, \ldots, m_k \rangle$ or encryptions of $\langle m_1', \ldots, m_k' \rangle$. The advantage of the forward secrecy adversary is denoted by $\mathfrak{adv}^{fs}_{\mathcal{A}}$:

$$\mathfrak{adv}^{fs}_{\mathcal{A},C}(n) := \left| \mathbf{Prob}_{x \in_U \{0,1\}^n : b \in_U \{1,2\}}[C(\mathcal{A}^{\mathcal{E}_b[x]}(1^n), \text{config}) = b] - \frac{1}{2} \right|$$

note that $\mathcal{E}_1[x]$ (resp. $\mathcal{E}_2[x]$) denotes the encryption oracle with initial shared randomness x that given $\langle m_i, m_i' \rangle$ returns the encryption of m_i (resp. the encryption of m_i'). C attempts to predict which one of the two encryption oracles is $\mathcal{A}$ communicating with.

Definition 53 *A cipher is said to have forward secrecy if for any forward secrecy adversary $\mathcal{A}, C$ it holds that $\mathfrak{adv}^{fs}_{\mathcal{A},C}(n)$ is negligible in n.*

4.5.4 The Stochastic Refresh-key (SR) Cipher

Let n be a security parameter and $\mathbb{P} := \{0,1\}^n$, $\mathbb{C} := \{0,1\}^{2n}$, $\mathcal{K} := \{0,1\}^n$. Also Let $G : \{0,1\}^n \to \{0,1\}^{2n}$ be a PRNG.

Encryption. The encryption function is probabilistic and defined as follows:

$$f(s, m) = \langle s', G(s) \dot{+} (s' || m) \rangle$$

where $s' \in \{0,1\}^n$ and is selected at random.

Decryption. The decryption function works as follows:

$$g(s, c) = \langle \overline{G(s) \dot{+} c}\ , \overline{\overline{G(s) \dot{+} c}}\ \rangle$$

It is easy to verify the encryption/decryption functions satisfy the two cipher conditions:

$$\mathcal{P}_2^2(g(s, \mathcal{P}_2^2(f(s, m)))) = m$$

$$\mathcal{P}_1^2(f(s, m)) = \mathcal{P}_1^2(g(s, \mathcal{P}_2^2(f(s, m))))$$

for all $s \in \mathcal{K}, m \in \{0,1\}^n$.

The operation of the Stochastic Refresh-key Cipher (SR-Cipher) defined above is illustrated in figure 4.7

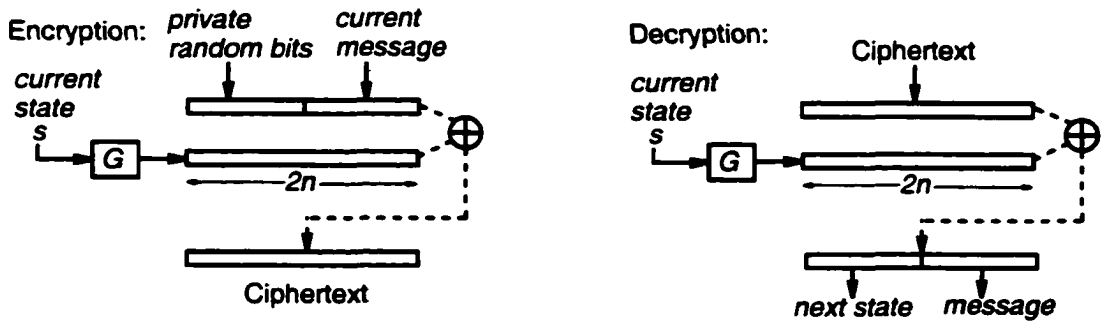

Figure 4.7: The SR-Cipher

4.5.5 Security of the SR-Cipher

We summarize the results of this section in figure 4.8.

Type of attack	Dependency on Underlying PRNG
Generic Chosen Plaintext Attack	$\mathbf{adv}_D^G(n) = k^{-1}\mathbf{adv}_{\mathcal{A},C}^{gcpa}(n)$
Chosen Plaintext Semantic Security	$\mathbf{adv}_D^G(n) = (k+1)^{-1}\mathbf{adv}_{\mathcal{A}}^{cpa}(n)$
Chosen Ciphertext Semantic Security	$\mathbf{adv}_D^G(n) = (k+1)^{-1}\mathbf{adv}_{\mathcal{A}}^{cca}(n)$
Forward Secrecy Attack	$\mathbf{adv}_D^G(n) = k^{-1}\mathbf{adv}_{\mathcal{A},C}^{fs}(n)$

Figure 4.8: Presentation of the security properties of the SR-Cipher: the value k is the number of queries posed by the adversary to the encryption/decryption oracle prior to receiving/submitting the challenge.

An interesting fact revealed by our results is that the number of queries posed to the oracle after the submission of the challenge plaintexts in a chosen plaintext/ciphertext semantic security adversary is not significant in the determination of the distinguishing advantage for the underlying PRNG.

The following theorem suggests that the SR-Cipher is resilient against generic chosen plaintext attack, provided that G is a PRNG.

Theorem 54 *Let $\mathcal{A}, C$ be a generic chosen plaintext attack adversary for the cipher. Then there exists a distinguisher D for the* PRNG *G, with advantage $\mathbf{adv}_D^G(n) = k^{-1}\mathbf{adv}_{\mathcal{A},C}^{gcpa}(n)$, where k is the number of queries used by $\mathcal{A}$.*

Proof. Assume $\mathcal{A} : \{0,1\}^{p(n)} \to \{0,1\}^{q(n)}$ and $C : \{0,1\}^{q(n)} \to \{0,1\}$ is a generic chosen plaintext adversary for the cipher.

Let k be the number of queries performed by $\mathcal{A}$. Given a $i \in \{1,\ldots,k\}$ we define the following hybrid encryption oracle $\mathcal{RE}_i$ that operates as follows: the first i queries of the adversary are answered by random elements of $\{0,1\}^{2n}$ (ignoring the queries); from the $(i+1)$-th query m_{i+1} and on the queries are answered as regular encryptions using a random initial state $s_i \in \{0,1\}^n$. It is easy to verify that the $\mathcal{RE}_0$ oracle behaves exactly as the encryption oracle $\mathcal{E}$, whereas the $\mathcal{RE}_n$ oracle behaves like the random sampler.

Now consider the following PPT D: given $T \in \{0,1\}^{2n}$ the circuit chooses $i \in \{1,\ldots,k\}$ uniformly at random and simulates the adversary $\mathcal{A}$ with the following oracle: the first $i-1$ queries are answered by random elements of $\{0,1\}^{2n}$. The i-th query of the adversary is answered by $T \dotplus (s||m_i)$ where s is randomly chosen over $\{0,1\}^n$. The remaining queries of the adversary are answered as standard encryptions with initial state s. Finally D outputs the result of C over the output of the adversary $\mathcal{A}$.

Let $\mathbf{Prob}_T[D(T)=1|j]$ where $j \in \{1,\ldots,k\}$ denote the probability D returns 1, given the fact that the random choice of i made by D equals j. Using standard probability rules it holds that $\mathbf{Prob}_T[D(T)=1] = \frac{1}{k}\sum_{j=1}^{k} \mathbf{Prob}_T[D(T)=1|j]$. Also depending on how the choice of T is made D behaves differently in this manner: $\mathbf{Prob}_{T\in_U\{0,1\}^{2n}}[D(T)=1|j] = \mathbf{Prob}[C(\mathcal{A}^{\mathcal{RE}_j}(1^n))=1]$ and $\mathbf{Prob}_{s\in_U\{0,1\}^n}[D(G(s))=1|j] = \mathbf{Prob}[C(\mathcal{A}^{\mathcal{RE}_{j-1}}(1^n))=1]$. By combining the above facts we easily deduce that:

$$\mathbf{Prob}_{T\in_U\{0,1\}^{2n}}[D(T)=1] = \frac{1}{k}\sum_{j=1}^{k} \mathbf{Prob}[C(\mathcal{A}^{\mathcal{RE}_j}(1^n))=1]]$$

$$\mathbf{Prob}_{s\in_U\{0,1\}^{n}}[D(G(s))=1] = \frac{1}{k}\sum_{j=0}^{k-1} \mathbf{Prob}[C(\mathcal{A}^{\mathcal{RE}_j}(1^n))=1]]$$

As a result

$$|\ \mathbf{Prob}_{T\in_U\{0,1\}^{2n}}[D(T)=1] - \mathbf{Prob}_{s\in_U\{0,1\}^{n}}[D(G(s))=1]\ | = k^{-1}\mathfrak{adv}^{gcpa}_{\mathcal{A},C}(n)$$

this completes the proof. ■

The following theorem suggests that the SR-Cipher is semantically secure under chosen plaintext attack, provided that G is a PRNG.

Theorem 55 *Let $\mathcal{A}$ be a chosen plaintext semantic security adversary for the cipher. Then there exists a probabilistic distinguisher algorithm D for the pseudorandom num-*

ber generator G, with advantage $\mathfrak{adv}_D^G(n) = (k+1)^{-1}\mathfrak{adv}_{\mathcal{A}}^{\mathcal{CPA}}(n)$, where k is the number of queries used by $\mathcal{A}$ prior to submitting the challenge plaintexts.

Proof. Consider the following "partially randomized" encryption oracle $\mathcal{RE}_j$ for $j \in \{1,\ldots,k+1\}$: queries $m_i \in \{0,1\}^n$, with $i \leq j$ are answered by R where $R \in_U \{0,1\}^{2n}$: The remaining queries m_i, $i > j$ are answered as regular encryptions: $G(s_i) \oplus (s_{i+1}||m_i)$ where $s_{i+1} \in_U \{0,1\}^n$ (note that s_j is selected at random from $\{0,1\}^n$). The challenge query M_1, M_2 is treated as above with $\mathcal{RE}_j$ selecting $b \in_U \{1,2\}$ at random and returning the encryption of M_b in the current state, unless $j = k+1$ where in this case the encryption of M_b is merely $R \oplus M_b$ with $R \in_U \{0,1\}^{2n}$. The queries m_i with $i > k+1$ are answered as proper encryption in the current state, i.e. $G(s_i) \oplus (s_{i+1}||m_i)$ with $s_{i+1} \in \{0,1\}^n$ (the new random state).

Consider now the following procedure D operating on input $T \in \{0,1\}^{2n}$: first a random $j \in \{1,\ldots,k+1\}$ is selected. Subsequently the adversary $\mathcal{A}$ is simulated using the following partially randomized encryption oracle: the $\mathcal{RE}_j$ is used with the change that the j-th query m_j is answered by $T \oplus (s_{j+1}||m_j)$, where $s_{j+1} \in_U \{0,1\}^n$ — note that if $j = k+1$ the answer is set to $T \oplus (s_{j+1}||M_b)$ where $b \in_U \{1,2\}$. Finally D outputs 1 if the adversary is successful in guessing b, otherwise it returns 0. Using similar arguments as in the proof of theorem 54, we deduce that: $\mathbf{Prob}_{T \in_U \{0,1\}^{2n}}[D(T) = 1] = \frac{1}{k+1}\sum_{j=1}^{k+1} \mathbf{Prob}_{b \in_U \{1,2\}}[\mathcal{A}^{\mathcal{RE}_j[b]}(1^n)) = b]$ and $\mathbf{Prob}_{s \in_U \{0,1\}^n}[D(G(s)) = 1] = \frac{1}{k+1}\sum_{j=0}^{k} \mathbf{Prob}_{b \in_U \{1,2\}}[\mathcal{A}^{\mathcal{RE}_j[b]}(1^n)) = b]$.

Moreover it is easy to see that when $j = 0$ the behavior or the $\mathcal{RE}_0$ oracle is identical to normal operation of the encryption oracle in the chosen plaintext semantic security attack:

$$\mathbf{Prob}_{b \in_U \{1,2\}}[\mathcal{A}^{\mathcal{RE}_0[b]}(1^n)) = b] = \mathbf{Prob}_{x \in_U \{0,1\}^n; b \in_U \{1,2\}}[\mathcal{A}^{\mathcal{E}[x,b]}(1^n) = b]$$

For the case $j = k+1$ we argue that:

$$\mathbf{Prob}_{b \in_U \{1,2\}}[\mathcal{A}^{\mathcal{RE}_{k+1}[b]}(1^n)) = b] = \frac{1}{2}$$

this is so, because for a fixed $b = 1$, for every sequence of coin-tosses of $\mathcal{A}$ and of the oracle $\mathcal{RE}_{k+1}$ that the adversary is successful (i.e. outputs 1) the same sequence of coin tosses will be unsuccessful for the choice of $b = 2$.

It follows that

$$|\ \mathbf{Prob}_{T\in_U\{0,1\}^{2n}}[D(T)=1] - \mathbf{Prob}_{s\in_U\{0,1\}^n}[D(G(s))=1]\ | = (k+1)^{-1}\mathfrak{adv}^{cpa}_{\mathcal{A}}(n)$$

this completes the proof. ■

The following theorem suggests that the SR-Cipher is semantically secure under a chosen ciphertext attack, provided that G is a PRNG.

Theorem 56 *Let $\mathcal{A}$ be a chosen ciphertext semantic security adversary for the cipher. Then there exists a probabilistic distinguisher algorithm D for the pseudorandom number generator G, with advantage $\mathfrak{adv}^G_D(n) = (k+1)^{-1}\mathfrak{adv}^{cca}_{\mathcal{A}}(n)$, where k is the number of decryption queries used by $\mathcal{A}$ prior to submitting the challenge plaintexts.*

Proof. Consider the following "partially randomized" decryption/encryption oracle $\mathcal{RDE}_j$ for $j \in \{1,\ldots,k+1\}$: queries $c_i \in \{0,1\}^n$, with $i < j$ are answered by R where $R \in_U \{0,1\}^n$. Suppose that $j < k+1$; the j-th query c_j is answered by selecting a random $s_j \in_U \{0,1\}^n$ and replying by $\overline{G(s_j)\oplus c_j}$. The next state $s_{j+1} := \overline{\overline{G(s_j)\oplus c_j}}$. Subsequent decryption queries c_i with $j < i < k+1$ are answered as regular decryptions: $\overline{G(s_i)\oplus c_i}$ and setting the next state to $s_{i+1} := \overline{\overline{G(s_i)\oplus c_i}}$. The $(k+1)$-th query is an *encryption* query in the current state that involves the challenge plaintexts. The oracle returns $G(s_{k+1})\oplus(s_{k+2}||M_b)$ with $s_{k+2} \in_U \{0,1\}^n$. Subsequent decryption queries c_i are answered as regular decryptions in the current state: the oracle's answer is $\overline{G(s_i)\oplus c_i}$ and setting the next state to $s_{i+1} := \overline{\overline{G(s_i)\oplus c_i}}$. In the case $j = k+1$ the j-th query is the challenge plaintexts encryption query. A random $s_{k+1} \in_U \{0,1\}^n$ is selected and the oracle returns $G(s_{k+1})\oplus(s_{k+2}||M_b)$ for $b \in_U \{1,2\}$. Subsequent queries are answered as decryption queries in the current state.

Consider now the following probabilistic algorithm D operating on input $T \in \{0,1\}^{2n}$: first a random $j \in \{1,\ldots,k+1\}$ is selected. Subsequently the adversary $\mathcal{A}$ is simulated using the "partially randomized" encryption oracle $\mathcal{RDE}_j$ with the following modification:

- If $j < k+1$, the j-th decryption query is answered by $\overline{T \dotplus c_j}$ and the next state s_{j+1} is set to $s_{j+1} := \overline{\overline{T \dotplus c_j}}$.
- If $j = k+1$, the challenge plaintexts encryption query is answered by $T \dotplus (s_{k+2}||M_b)$ with $b \in_U \{1,2\}$.

Finally D outputs 1 if the adversary is successful in guessing b, otherwise it returns 0. Using a similar argument as in the proof of theorem 54 we deduce that:

$$\mathbf{Prob}_{T\in_U\{0,1\}^{2n}}[D(T)=1] = \frac{1}{k+1}\sum_{j=1}^{k+1}\mathbf{Prob}_{b\in_U\{1,2\}}[\mathcal{A}^{\mathcal{RDE}_j[b]}(1^n))=b]$$

$$\mathbf{Prob}_{s\in_U\{0,1\}^{n}}[D(G(s))=1] = \frac{1}{k+1}\sum_{j=0}^{k}\mathbf{Prob}_{b\in_U\{1,2\}}[\mathcal{A}^{\mathcal{RDE}_j[b]}(1^n))=b]$$

Moreover it is easy to see that when $j = 0$ the behavior or the $\mathcal{RDE}_0$ oracle is identical to normal operation of the decryption/encryption oracle in the chosen ciphertext semantic security attack:

$$\mathbf{Prob}_{b\in_U\{1,2\}}[\mathcal{A}^{\mathcal{RDE}_0[b]}(1^n))=b] = \mathbf{Prob}_{x\in_U\{0,1\}^n;b\in_U\{1,2\}}[\mathcal{A}^{\mathcal{DE}[x,b]}(1^n)=b]$$

For the case $j = k+1$ we argue that:

$$\mathbf{Prob}_{b\in_U\{1,2\}}[\mathcal{A}^{\mathcal{RDE}_{k+1}[b]}(1^n))=b] = \frac{1}{2}$$

this is so, because for a fixed $b = 1$, for every sequence of coin-tosses of $\mathcal{A}$ and of the oracle $\mathcal{RDE}_{k+1}$ that the adversary is successful (i.e. outputs 1) the same sequence of coin tosses will be unsuccessful for the choice of $b = 2$.

It follows that

$$|\ \mathbf{Prob}_{T\in_U\{0,1\}^{2n}}[D(T)=1]-\mathbf{Prob}_{s\in_U\{0,1\}^n}[D(G(s))=1]\ |=(k+1)^{-1}\mathfrak{adv}^{cca}_{\mathcal{A}}(n)$$

this completes the proof. ■

Remark. It is possible to show semantic security in a hybrid adversarial model that employs an arbitrary interleaving of chosen plaintext and chosen ciphertext queries: in this setting the adversary is capable of posing queries to both an encryption and a decryption oracle prior and after selecting the challenge plaintexts. The proof combines the proofs of theorems 55 and 56.

Next we deal with forward secrecy. Note that for our SR-Cipher a configuration consists only of the current state of the system. The theorem below suggests that the SR-Cipher is forward secure provided that G is a PRNG.

Theorem 57 *Let $\mathcal{A}, C$ be a forward secrecy adversary for the SR-Cipher. Then there exists a distinguisher D for the* PRNG G. *with advantage $\mathfrak{adv}^G_D(n)=k^{-1}\mathfrak{adv}^{fs}_{\mathcal{A},C}(n)$. where k is the number of queries used by $\mathcal{A}$.*

Proof. Let $\mathcal{A}:\{0,1\}^n\to\{0,1\}^{q(n)}$ and $C:\{0,1\}^{q(n)}\times\{0,1\}^n\to\{0,1\}$

Let k be the queries performed by $\mathcal{A}$. We define the following hybrid encryption oracle $\mathcal{RE}_j$ that operates as follows: first a random $b\in_U\{1,2\}$ is selected: the first j queries of the adversary are answered by random elements of $\{0,1\}^{2n}$ (ignoring the queries); the remaining queries $\langle m_i,m'_i\rangle$, $i\geq j$, are answered as standard encryptions of m_i, if $b=1$, or encryptions of m'_i if $b=2$: i.e. in the case $b=1$, $G(s_i)\oplus(s_{i+1}||m_i)$ with $s_{i+1}\in_U\{0,1\}^n$ (note that s_j is selected at random from $\{0,1\}^n$ in the j-th query) — similarly using m'_i, if $b=2$.

Consider the following PPT D: given $T\in\{0,1\}^{2n}$ the circuit chooses $j\in\{1,\ldots,k\}$ uniformly at random and simulates the adversary $\mathcal{A}$ using $\mathcal{RE}_j$ as the oracle with the modification that the j-th query is answered $T\oplus(s_{j+1}||m_j)$ where

$s_{j+1} \in_U \{0,1\}^n$. Finally D outputs 1 if C given the output of the adversary $\mathcal{A}$ and s_{k+1} correctly predicts b. It follows that:

$$\mathbf{Prob}_{T\in_U\{0,1\}^{2n}}[D(T)=1] = \frac{1}{k}\sum_{j=1}^{k} \mathbf{Prob}_{b\in_U\{1,2\}}[C(\mathcal{A}^{\mathcal{RE}_j[b]}(1^n), s_{k+1}) = b]$$

and

$$\mathbf{Prob}_{s\in_U\{0,1\}^n}[D(G(s))=1] = \frac{1}{k}\sum_{j=0}^{k-1} \mathbf{Prob}[C(\mathcal{A}^{\mathcal{RE}_j[b]}(1^n), s_{k+1}) = b]$$

Now observe that $\mathcal{RE}_0[b]$ behaves exactly as the encryption oracle $\mathcal{E}[b]$ used in the definition of the forward security adversary. This suggests that

$$\mathbf{Prob}_{b\in_U\{1,2\}}[C(\mathcal{A}^{\mathcal{RE}_0[b]}(1^n), s_{k+1}) = b] = \mathbf{Prob}_{x\in\{0,1\}^n; b\in_U\{1,2\}}[C(\mathcal{A}^{\mathcal{E}_b[x]}(1^n), s_{k+1}) = b]$$

On the other hand it is easy to see that $\mathbf{Prob}_{b\in_U\{1,2\}}[C(\mathcal{A}^{\mathcal{RE}_k[b]}(1^n), s_{k+1}) = b] = \frac{1}{2}$ (as in this case all the oracle answers are random therefore guessing b can only be done with probability 1/2). It follows that

$$|\ \mathbf{Prob}_{T\in_U\{0,1\}^{2n}}[D(T)=1] - \mathbf{Prob}_{s\in_U\{0,1\}^n}[D(G(s))=1]\ | = k^{-1}\mathfrak{adv}^{fs}_{\mathcal{A},C}(n)$$

this completes the proof. ■

The Mixed Mode. The key refresh operation of the SR-Cipher is expensive in terms of bandwidth overhead if the stretch of the PRNG G is small. In our exposition we considered a stretch factor of 2, i.e. $G : \{0,1\}^n \to \{0,1\}^{2n}$. This is not a restriction though, as our construction can be readily employed over a PRNG with greater stretch factor $G : \{0,1\}^n \to \{0,1\}^{n+p(n)}$ where p is some polynomial in n. In this case the plaintext space of the SR-Cipher is $\mathbb{P} := \{0,1\}^{p(n)}$ and the cipher behaves more like a mode of operation: each plaintext transmission becomes a "session" that allows the secure transmission of $p(n)$ bits and a new random key of n bits to be used for the next session. Our security analysis can be immediately applied to this extended setting retaining semantic security while at the same time yielding forward secrecy between sessions.

4.5.6 Dealing with Resynchronization

All stateful-ciphers (e.g., stream ciphers) have to deal with desynchronization problems since the two parties, sender and receiver, need to be in a lock step in order to enable communication (in other words the sequence of blocks should be decrypted in exactly the same order as encrypted). Any change in block-sequence (error or swap of ciphertexts) will introduce errors in upcoming decryptions (error propagation). This is a well known problem that any stream (stateful) cipher has to deal with. Typically the problem is dealt by employing error-correcting codes that can withstand a number of errors that occur in the communication channel either at random or introduced by an adversary.

Since we are interested in forward secrecy it is not possible for the state-sequence to be entirely public. As a result the typical resynchronizing technique of sending the current state in the clear and thus resynchronizing the sender and receiver does not apply in the forward secrecy setting.

An easy to implement and effective solution we propose here is the exchange of a random key in the beginning of a communication "thread" between the sender and the receiver. This random key will be used to introduce a new communication thread in case the current one gets irreparably desynchronized. Of course this will sacrifice n bits of transmitted data but it will allow the restart of a new thread in case of a catastrophic error (when the employed error-correcting code is unable to correct the introduced errors).

Resynchronizing. When the receiver detects an unrecoverable error in a certain ciphertext it sends a "restart" message to the sender starting from the block that was transmitted with errors (we assume all cipher blocks are serial-numbered). Both sender and receiver use the previously stored n bits as the new shared random key (we will refer to them as the "back-up" state) to start a new thread of cipher operation.

This operation is illustrated in figure 4.9. We note here that it is not secure to resend the corrupted message encrypted again without starting a new thread since this will result in a total security breach if the adversary is capable of corrupting messages transmitted in the channel.

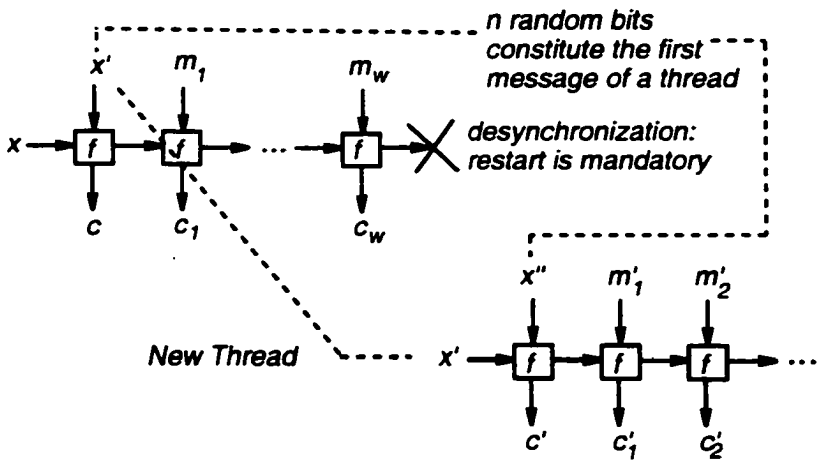

Figure 4.9: Resynchronizing a Stateful Cipher

The security definitions given in section 4.5.3 can be extended to allow the query generator $\mathcal{A}$ apart from querying the encryption oracle to also force an arbitrary number of "restarts" of the cipher.

We now consider the case of a generic chosen plaintext adversary capable of forcing restarts of the cipher. This is a pair of PPTs $\mathcal{A}, C$ so that $\mathcal{A}$ submits adaptively two types of queries to the oracle: (i) plaintexts, where it receives the encryption of the plaintext under the current state of the system, (ii) restart-requests in which case the state of the system is set to a previously stored back-up state and the adversary receives the encryption of a random string that is going to be used as the initial state of the next communication thread (the new back-up state). We assume that the first query of the adversary is always a restart-request (to start a new communication

thread); note that the initial "back-up" state is selected at random (coincides with the shared randomness between sender and receiver during normal protocol operation). Finally C given the output of $\mathcal{A}$, it attempts to distinguish whether the output of the encryption oracle to the adversary agrees with the legitimate operation of the system or is just random samples of the ciphertext space $\{0,1\}^{2n}$. The advantage of the adversary $\mathfrak{adv}^{r-gcpa}_{\mathcal{A},C}$ is defined as in section 4.5.3.

Theorem 58 *Let $\mathcal{A}, C$ be a generic chosen plaintext adversary with restart capability. Then there exists a distinguisher D for the* PRNG *G, with advantage $\mathfrak{adv}^{G}_{D}(n) = k^{-1}\mathfrak{adv}^{r-gcpa}_{\mathcal{A},C}(n)$, where k is the number of queries used by $\mathcal{A}$ (including the restart requests).*

Proof. Let k be the number of queries made by $\mathcal{A}$ (including the restart requests). Given a $j \in \{1,\ldots,k\}$, we define the following partially randomized encryption oracle $\mathcal{RE}_j$: Given the i-th query with $i \leq j$, $\mathcal{RE}_j$ returns a random element $R \in_U \{0,1\}^{2n}$. For the i-th query with $i > j$ there are two possibilities: (i) given a message m_i the oracle responds with the encryption of m_i in the current state: $G(s_i) \oplus (s_{i+1}||m_i)$ (and sets the next state to $s_{i+1} \in_U \{0,1\}^n$) — note that if $j = i+1$ the current state s_i is selected at random from $\{0,1\}^n$. (ii) given a restart-request the oracle responds with the encryption of the next back-up state x' encrypted under the current back-up state $G(x) \oplus (s_{i+1}||x')$ and sets the next state to $s_{i+1} \in_U \{0,1\}^n$. Note that if this is the first restart-request the oracle chooses the current back-up state at random from $\{0,1\}^n$.

Now consider the following distinguisher PPT D: given $T \in \{0,1\}^{2n}$, D selects a $j \in_U \{1,\ldots,k\}$ and then simulates $\mathcal{A}$ using the $\mathcal{RE}_j$ oracle with the following modification: the j-the query is answered using T in the following sense: (i) if the j-th query is a message to be encrypted the oracle's reply is $T \oplus (s_{j+1}||m_j)$ with $s_{j+1} \in_U \{0,1\}^n$ (the next state of the system); (ii) if the j-th query is a restart request the

oracle returns $T \oplus (s_{j+1}||x')$ where $x' \in_U \{0,1\}^n$ and x' is the new back-up state, and $s_{j+1} \in_U \{0,1\}^n$ is the new state. Using a similar argument as in the proof of theorem 54 we deduce that: $\mathbf{Prob}_{T\in_U\{0,1\}^{2n}}[D(T) = 1] = \frac{1}{k}\sum_{j=1}^{k} \mathbf{Prob}[C(\mathcal{A}^{\mathcal{RE}_j}(1^n)) = 1]]$ $\mathbf{Prob}_{s\in_U\{0,1\}^n}[D(G(s)) = 1] = \frac{1}{k}\sum_{j=0}^{k-1} \mathbf{Prob}[C(\mathcal{A}^{\mathcal{RE}_j}(1^n)) = 1]]$. As a result

$$|\ \mathbf{Prob}_{T\in_U\{0,1\}^{2n}}[D(T) = 1] - \mathbf{Prob}_{s\in_U\{0,1\}^n}[D(G(s)) = 1]\ | = k^{-1}\mathfrak{adv}^{r-gcpa}_{\mathcal{A},C}(n)$$

this completes the proof. ■

The other attacks described in section 4.5.3 can be extended to the restart scenario similarly as above. We summarize these results in the theorem below:

Theorem 59 *The results below refer to a (generic chosen plaintext attack, chosen plaintext semantic, chosen ciphertext semantic, forward secrecy) adversary that is capable of forcing a number of restarts in the cipher operation.*

1. *Generic Chosen Plaintext Attack* $\mathfrak{adv}^{G}_{D}(n) = k^{-1}\mathfrak{adv}^{r-gcpa}_{\mathcal{A},C}(n)$
2. *Chosen Plaintext Semantic Security* $\mathfrak{adv}^{G}_{D}(n) = (k+1)^{-1}\mathfrak{adv}^{r-cpa}_{\mathcal{A}}(n)$
3. *Chosen Ciphertext Semantic Security* $\mathfrak{adv}^{G}_{D}(n) = (k+1)^{-1}\mathfrak{adv}^{r-cca}_{\mathcal{A}}(n)$
4. *Forward Secrecy Attack* $\mathfrak{adv}^{G}_{D}(n) = k^{-1}\mathfrak{adv}^{r-fs}_{\mathcal{A},C}(n)$

note that k is the number of encryption/decryption queries plus the number of restart-requests posed by the adversary.

Chapter 5

Secure Games with Polynomial Expressions

5.1 Chapter Preface

One of the most important results on the foundations of Cryptography (suggested by Yao [Yao86], generalized to multi-party by Goldreich, Micali and Wigderson [GMW87], and characterized based on the Oblivious Transfer primitive by Kilian [Kil90]) is that given any polynomially computable function $f(x, y)$, it is possible for two parties, Alice (A for short) and Bob (B for short), to jointly compute $f(\alpha, \beta)$, with A contributing α and B contributing β, in such a way so that no party learns anything more than what can be deduced by the final output. The resulting protocols are relative to the size of the circuit that computes f that, even for simple functions, are considerably expensive to implement. Consequently, nowadays where distributed applications over the Internet are about to become a reality, it is worthwhile to seek special cases of useful function families that can accept more efficient protocol techniques (as advocated in [Gol97]).

In that spirit, Naor and Pinkas [NP99] introduced an efficient protocol for obliviously computing the value of a polynomial (Oblivious Polynomial Evaluation, OPE). In their setting, B possesses a polynomial P, A has a value α and wishes to obliv-

iously compute $P(\alpha)$. The security of their protocol was based on a variant of the Polynomial Reconstruction Problem.

In this chapter, we further investigate possibilities for efficient solutions of new useful problems in the general area of secure function evaluation by introducing a family of protocols called *Secure Games with Polynomial Expressions* (SGPEs). The general idea of our approach is to consider the joint computation of a polynomial expression that is made up of secret polynomials owned by the two players (as well as non-secret components). Player A selects an input for the expression, and wishes to obtain the value of the expression on this input. Depending on the contribution of A to the expression we can categorize SGPEs to those that A contributes only field elements to the expression (type 1), and to those that A contributes also polynomials (type 2). An example of a type-1 SGPE is the Secure Multivariate Polynomial Evaluation (SMPE): B holds a secret multivariate polynomial P, and A wishes to obtain a point in the graph of P of her choice. A Secure Nested Polynomial Game (SNPG for short) is an example of SGPE of type-2: A holds a constant number of c secret polynomials $Q_2, \ldots, Q_c$ and wants to compute $P_c(Q_c(\ldots(P_2(Q_2(P_1(\alpha))))\ldots))$ for an α of her choice, where the polynomials $P_1, P_2, \ldots, P_c$ are contributed by B.

The security conditions that we consider, are the following: A does not want to reveal anything about the data she contributes to the game, and B does not want to disclose his data beyond what is trivially inferred from A's output. In addition to the above (traditional) conditions, both players wish that if the secrecy of some of their private data is compromised, the secrecy of the remaining secrets will remain unaffected (we call this property *secret independence*), or more generally that their data are secure even if they are not uniformly distributed over all possible inputs.

We present an efficient construction for SGPEs of type-1 and an efficient transformation of a type-2 game to a type-1 game. We get a protocol of two flows of communication, one of which is employing an implementation of a single t-out-of-n

Oblivious Transfer over values of the proper field, where t and n are small polynomial functions (in the size of the polynomial expression used in the game). The security assumption we employ is an extension of the Decision Polynomial Reconstruction assumption (from section 3.1.5).

We present two cryptographic applications of Secure Games with polynomial Expressions, namely *Oblivious Negotiations*, and *Oblivious Affine Evaluations* which can be utilized to allow the construction of an efficient *Oblivious Scoring* protocol.

Given our setting and its applications, we note that the OPE problem, which was our initial inspiration, can be solved as an instance of our setting. Reducing our setting to OPE encounters a number of problems: (1) functionality: A's contribution to the expression should be randomized, and (2) security: secret-independence is not enforced. These requirements, that fail in the reduction, are necessary for the new applications. Finally, (3) complexity: naive reductions may blow up the computation, exponentially. We note that starting from the OPE protocol two-flow structure for two layer computation (polynomial over data), it is not at all obvious how to retain this protocol structure for our multi-layer setting, but, interestingly, we show it to be possible.

5.2 Preliminaries and Definitions

Let $\mathcal{P} := \{P_1, P_2, \ldots\}$ be a set of predicates, and $\mathcal{X} := \{x_1, x_2, \ldots\}$ a set of variables. An expression $\mathcal{E}$ is a rooted-DAG (directed acyclic graph) with all arcs directed towards the root specified as follows: each node is one of the following: P_i, $+$, $\cdot$, or a natural number. If a node is $+$ or $\cdot$ then it has two children, if a node is a number it has a single child; if a node is P_i then it has any non-zero number of children: each arc entering P_i is labeled by a non-zero natural number; The leaves of the DAG are selected from $\mathcal{X}$. The *value* of a path from a leaf to the root, is the product of all

labels and number nodes that are in its course (and is set to 1, if there are no labels or number nodes). The *degree* of a variable is defined as the maximum path value taken over all paths from the variable node to the root. Let $\mathcal{E}$ be an expression, and let $P_1, \ldots, P_v$ denote its predicate nodes; if we map each predicate P_i to a polynomial with the same number of variables as the children of P_i and of the same degrees as the labels of its incoming arcs, an in-order traversal of $\mathcal{E}$ can be seen as a polynomial (interpreting each number node as exponentiation); we denote this polynomial by $\mathcal{E}(P_1, \ldots, P_v)$, and say that the polynomials $P_1, \ldots, P_v$ "fit into" $\mathcal{E}$.

If P is a predicate node we denote by $\text{label}(P, j)$ the label of the j-th incoming arc. Let $|\mathcal{E}|$ denote the size of the DAG (number of arcs). We define $\text{size}(\mathcal{E}) := |\mathcal{E}| + \sum_P \prod_j (\text{label}(P, j) + 1)$ where the sum is over all predicate nodes of $\mathcal{E}$. In order to store $\mathcal{E}(P_1, \ldots, P_v)$ we need $\text{size}(\mathcal{E})$ space. One of the reasons for introducing expressions instead of talking simply about polynomials is space: if $\text{coef}(P)$ denotes the number of coefficients of a polynomial P, then it holds that $\text{coef}(\mathcal{E}(P_1, \ldots, P_v))$ can be exponentially large compared to $\text{size}(\mathcal{E})$. In order to compute a value of $\mathcal{E}(P_1, \ldots, P_v)$ using the expression representation we need $\mathcal{O}(\text{size}(\mathcal{E}))$ field operations. If $\mathcal{E}$ is an expression, denote by $d_1, \ldots, d_r$ the degrees of its variables. For a fixed constant c, we say that an expression is c-bound if $\text{lcm}(d_1, \ldots, d_r) = \mathcal{O}([\text{size}(\mathcal{E})]^c)$.

For the following, fix a c-bound expression $\mathcal{E}$ with v predicates and r variables. A type-1 SGPE is as follows: player B has v secret polynomials $P_1, \ldots, P_v$, player A has r secret values $\alpha_1, \ldots, \alpha_r \in \mathbb{F}$ and wants to obtain $\mathcal{E}(P_1, \ldots, P_v)(\alpha_1, \ldots, \alpha_r)$. Some of the polynomials of player B may be publicly known. If $v = 1$ and $\mathcal{E}(P) := P$, then the game is called "Secure Multivariate Polynomial Evaluation" (SMPE). A type-2 SGPE is defined similarly with the only difference that some of the $P_1, \ldots, P_v$ polynomials are contributed by A. When $\mathcal{E}$ has the form $P_c(Q_c(\ldots(P_2(Q_2(P_1(x))))\ldots))$ with the P_i contributed by B, and the Q_i contributed by A then we will call this game a "Secure Nested Polynomial Game" (SNPG). Our game schema involves two flows of

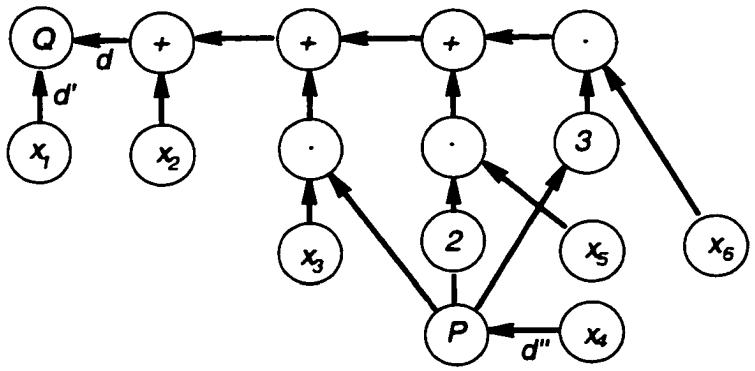

Figure 5.1: Example of an expression that defines the polynomial $Q(x_1, x_2 + x_3P(x_4) + x5(P(x_4))^2 + x_6(P(x_4))^3)$, with degree$(x_2)$ = degree(x_3) = degree(x_5) = degree(x_6) = d, degree(x_1) = d', degree(x_4) = $3dd''$.

information, from A to B and from B to A (this latter flow employs a t-out-of-n OT). Correctness and security requirements for both types of games are as follows:

Definition 60 *Let $\mathcal{E}$ be a c-bound expression with v predicates $P_1, \ldots, P_v$, and r variables. Let $\mathcal{H}_A, \mathcal{H}_B$ denote the sequence of secrets contributed by the two players to the expression. There are two PPT algorithms $\mathcal{A}, \mathcal{B}$ and a deterministic algorithm $\mathcal{C}$ (parts of our protocol) so that: $\mathcal{C}(\mathcal{B}(\mathcal{A}(\mathcal{H}_A), \mathcal{H}_B)) = \mathcal{E}(P_1, \ldots, P_v)(\alpha_1, \ldots, \alpha_r)$ (independently of the coin tosses of $\mathcal{A}, \mathcal{B}$). Informally, $\mathcal{A}$ is used by A to hide her secrets and give them to B; B uses $\mathcal{B}$ to hide his secrets and apply them over the secrets of A; $\mathcal{C}$ is used by A to reconstruct the output of the protocol from the reply of B (which is obtained through a t-out-of-n OT). The computation cost is polynomial in* size$(\mathcal{E})$.

Security of A. *Informally, the security of A is established by showing that B cannot deduce anything meaningful out of the protocol transcript he receives. More formally, for all PPT $\mathcal{B}'$ playing B's part and all distributions $\mathcal{D}_A$ under which $\mathcal{H}_A$ is distributed there is a PPT $\mathcal{B}''$ such that the following is negligible:*

$$| \mathbf{Prob}[Z = \mathcal{H}_A : Z \leftarrow \mathcal{B}'(\mathcal{A}(\mathcal{H}_A))] - \mathbf{Prob}[Z = \mathcal{H}_A : Z \leftarrow \mathcal{B}''] |$$

Security of B. *Informally, security of B can be claimed by comparing with the ideal implementation. Let $\mathcal{A}_0$ be the PPT used by player A to generate the query to player B in the first communication flow. Let $\mathcal{I}(\mathcal{H}_A, \mathcal{H}_B)$ is what player A obtains in the ideal implementation of the protocol Also. let $\mathcal{T}(\mathcal{H}_A, \mathcal{H}_B)$ be the protocol transcript obtained by player A at the end of the protocol. We show that for any PPT $\mathcal{A}'$ and any $\mathcal{H}_A$ there is a PPT $\mathcal{A}''$ s.t.*

$$| \mathbf{Prob}[\mathcal{A}'(\mathcal{T}(\mathcal{H}_A, \mathcal{H}_B)) = 1] - \mathbf{Prob}[\mathcal{A}''(\mathcal{I}(\mathcal{H}_A, \mathcal{H}_B)) = 1] |$$

is negligible (the probability is taken over the internal coin tosses of $\mathcal{A}', \mathcal{A}''$ and $\mathcal{H}_B$ is distributed according to $\mathcal{D}_B$).

Note that we have restricted the applicability of our protocol to c-bound expressions. Although we do not rule out the existence of a construction for unbounded expressions, c-bound expressions are sufficient for all applications discussed here.

The security of the party A for the OPE protocol of [NP99] was based on the polynomial reconstruction problem (see definition 10), which is also related to the security of A in our construction. We recall the definition below:

Definition 61 Polynomial Reconstruction (PR). *Given n, k, t and the distinct points $\{\langle z_i, y_i \rangle\}_{i=1}^{n}$ of $\mathbb{F}^2$, output all $\langle p, I \rangle$ such that $p \in \mathbb{F}[x]$, $\text{degree}(p) < k$, $I \subseteq \{1, \ldots, n\}$, $|I| = t$ and $\forall i \in I(p(z_i) = y_i)$.*

Let us comment briefly on the relation of SGPEs and OPE. In particular if it is possible to simulate a SGPE using OPE; there are two possibilities: (1) if only univariate polynomials appear in the expression the two players can use many individual OPEs to obtain intermediate results and finally player A will compose the final output. Nevertheless to conform to the security requirements randomization of the partial results is necessary something that appears to be hard unless the expression

degenerates to an affine transformation. (2) in the case of multivariate polynomial evaluation e.g. $P(\alpha, \beta)$, it can be performed by OPE as follows: A sends to B, random $s(x), s'(x) \in \mathbb{F}[x]$ s.t. $s(x_0) = \alpha$ and $s'(x_0) = \beta$ (x_0 is kept secret by A): A and B engage in OPE so that A obtains $P(s(x_0), s'(x_0))$. This approach has the deficiency that the values contributed by A are not "independently secure", i.e. partial knowledge of some of the values (or a small search-space for one of the values) can lead to the recovery of all secret input of A with non-negligible probability.

5.3 SPGEs of type 1

In the following construction, a t-out-of n OT protocol is used as a primitive.

- *Protocol parameter*: a c-bound expression $\mathcal{E}$ of v predicates.
- *Input of B*: Polynomials $P_1, \ldots, P_v$ that fit into $\mathcal{E}$.
- *Input of A*: r elements of $\mathbb{F}$, $\alpha_1, \ldots, \alpha_r$.
- *Output of A*: $\mathcal{E}(P_1, \ldots, P_v)(\alpha_1, \ldots, \alpha_r)$.
- *Security parameters*: n, l.
- Let $P(x_1, \ldots, x_r) := \mathcal{E}(P_1, \ldots, P_v)(x_1, \ldots, x_r)$, and denote by d_ℓ the degree of x_ℓ in P. Set $d := l r \mathrm{lcm}(d_1, \ldots, d_r)$, and $k := \min_\ell \frac{d}{r d_\ell} + 1$.

Step 1. A generates r instances of the noisy PR, $\{\langle z_i, y_{i,\ell} \rangle\}_{i=1}^n$ with solution $\langle p_\ell, l \rangle$, such that $p_\ell(0) := \alpha_\ell$, $\mathrm{degree}(p_\ell) = k - 1$, $z_i \neq 0$ and $z_i \neq z_j$ for all i, j, $j \neq i$. Then, A, forms the $(r+1)$-tuples $\{\langle z_i, y_{i,1}, \ldots, y_{i,r} \rangle\}_{i=1}^n$, and she sends them to B.

Step 2. B hides P in a random polynomial Q: Let $C, C' \in \mathbb{F}[x]$ be random polynomials of degree d such that $C(x) = C'(x) = 0$. Define a polynomial $Q \in \mathbb{F}[x_0, x_1, \ldots, x_r]$ as follows: $Q(x_0, \ldots, x_r) = P(x_1, \ldots, x_r) + C(x_0) + x_1^{d_1} \ldots x_r^{d_r} C'(x_0)$. The storage

space needed for Q is size($\mathcal{E}$) $+2d$. Computing a value of Q requires $\mathcal{O}$(size($\mathcal{E}$) $+d$) field operations. For each tuple $(z_i, y_{i,1}, \ldots, y_{i,r})$ B computes the value $Q(z_i, y_{i,1}, \ldots, y_{i,r})$. Note that the polynomial $R(x) := Q(x, p_1(x), \ldots, p_r(x))$ on 0 gives $R(0) = P(\alpha_1, \ldots, \alpha_r)$. The degree of R is $d_R = d + d_1 d_{p_1} + \ldots + d_r d_{p_r} \leq 2d$. Therefore, if A learns $t := 2d + 1$ values of R, she can interpolate it and compute $R(0)$.

Step 3. A and B engage in a t-out-of n OT in which A chooses to learn the values $Q(z_i, p_1(z_i), \ldots, p_r(z_i))$. Now A knows $2d + 1$ values of the polynomial R and can interpolate it to compute $R(0) = P(\alpha_1, \ldots, \alpha_r)$.

Implementation and Complexity. Clearly, A can compute $P(\alpha_1, \ldots, \alpha_r)$ for any $\alpha_1, \ldots, \alpha_r$ of her choice. The time-complexity of the protocol is $\mathcal{O}(rn + d \log d \log\log d + f_A(t, n))$ for player A and $\mathcal{O}(nd + n\text{size}(\mathcal{E}) + f_B(t, n))$ for player B, where $f_A(t, n)$, $f_B(t, n)$ denotes the running time of the t-out-of n OT protocol for each player respectively. The communication complexity is $\mathcal{O}(rn + c(t, n))$ where $c(t, n)$ is the communication complexity of the t-out-of n OT. We point here that if the expression $\mathcal{E}$ is 0-bound, then the complexity of player A does not depend on the size of the expression. For a t-out-of-n OT protocol the reader is referred to e.g. [NP99] where t-out-of n OT is efficiently and unconditionally reduced to 1-out-of 2 OT.

5.3.1 Comments on the Security of A

Player B, out of his participation in our protocol, receives an instance of the following problem, which we call the Multisample Polynomial Reconstruction (or MPR).

Definition 62 MPR. *Given n, k, t, r, and the distinct tuples $\{\langle z_i, y_{i,1}, \ldots, y_{i,r}\rangle\}_{i=1}^n$ so that each $\{\langle z_i, y_{i,\ell}\rangle\}_{i=1}^n$ is a noisy PR instance with parameters n, k, t and solution $\langle p_\ell, I\rangle$, find $\langle p_1, \ldots, p_r, I\rangle$.*

MPR appears to be hard on the average a fact that is justified as in the case of PR vs. noisy PR: given an instance of MPR it is possible to randomize the polynomials,

but (as in the case of PR) it is not apparent how to randomize the noise. We will formulate this as a complexity assumption:

Complexity Assumption. For any r there are n, k, t polynomially related parameters so that any probabilistic algorithm solving the MPR has negligible success probability in n.

Solving MPR either involves using techniques against a specific noisy PR instance that is included in the MPR instance (since the recovery of some $\langle p_\ell, I \rangle$ immediately implies the recovery of $\langle p_1, \ldots, p_r, I \rangle$) or in a more direct fashion trying to take advantage of the relation between the noisy PR instances included in the MPR instance. The best algorithm for solving PR is [GS98], which succeeds when $t \geq \sqrt{kn}$.

Alternatively, it is possible to solve MPR in a more direct manner. This can also be seen by our observation that the MPR problem corresponds to the decoding problem of interleaved Reed-Solomon Codes (under a certain noise-assumption). This relation is explained in depth in section 6.1. There, we detail an algorithm that solves the MPR for choices of the parameters that satisfy $r \geq \frac{n}{t}$. Some recent observations indicate that this might be improved to choices $r \geq \log_{\frac{t}{k}} \frac{n}{k}$, [Cop02]. For other choices of the parameters it is not apparent that MPR is solvable.

The relation of MPR to the security of A is detailed in section 5.5.

5.3.2 Comments on the Security of B

The security of player B is established by showing that the output of player A out of a protocol execution (the protocol transcript obtained by A) is essentially identical to what she gets in an ideal implementation. This holds true independently of A's behavior. In an ideal implementation, A gives to a trusted third party C all information send to B in step 1 of the protocol together with the randomness she used — note that this reveals her secret values $\alpha_1, \ldots, \alpha_r$. Player B gives to C its secret

input $P_1, \ldots, P_v$. In turn, C returns to A, either a value of $\mathcal{E}(P_1, \ldots, P_v)(x_1, \ldots, x_r)$ or a linear combination of some values of $\mathcal{E}(P_1, \ldots, P_v)(x_1, \ldots, x_r)$. In section 5.6, we will show the following regarding the security of player B:

Lemma 63 *There is a PPT $\mathcal{G}$ that given the output of the ideal implementation of the protocol for player A, and all information available to player A, generates a protocol transcript that is statistically indistinguishable from legitimate protocol transcripts generated during normal operation, under the assumption that t-out-of-n OT can be implemented ideally.*

Theorem 64 *Our construction satisfies definition 60 for the security of B, under the assumption that the underlying t-out-of-n OT is secure.*

5.4 SGPEs of type 2

In this section we present a transformation of type-2 games to type-1 games. First we deal with SNPGs: we will consider only the two round case and it will become clear how to generalize to any constant number of rounds. Suppose B possesses the secret polynomials $P_2, P_1 \in \mathbb{F}[x]$ and A the secret polynomial $Q_2 \in \mathbb{F}[x]$ of degree δ (known to B). A wants to compute $P_2(Q_2(P_1(\alpha)))$ for an α of her choice. B defines the expression $\mathcal{E}(P_1, P_2)(x_0, \ldots, x_\delta, x) = P_2(x_0 + x_1 P_1(x) + \ldots x_\delta (P_1(x))^\delta)$.

If $Q_2(x) = a_0 + a_1 x + \ldots a_\delta x^\delta$ then A, using the type-1 protocol, can compute the value $\mathcal{E}(P_2, P_1)(a_0, \ldots, a_\delta, \alpha)$ for an α of her choice. Now by the definition: $\mathcal{E}(P_2, P_1)(a_0, \ldots, a_\delta, \alpha) = P_2(a_0 + a_1 P_1(\alpha) + \ldots a_\delta (P_1(\alpha))^\delta) = P_2(Q_2(P_1(\alpha)))$.

The case for any type-2 game can be sketched as follows: player A should obtain $\mathcal{E}(P_1, \ldots, P_v, Q_1, \ldots, Q_{v'})(\alpha_1, \ldots, \alpha_r)$ where the polynomials $P_1, \ldots, P_v$ are contributed by B, and the values $\alpha_1, \ldots, \alpha_r$, and polynomials $Q_1, \ldots, Q_{v'}$ are contributed by A. For simplicity we assume that the polynomials Q_i are univariate. Let the degree of Q_i be δ_i. B substitutes in the expression $\mathcal{E}$ each occurrence of $Q_i(V)$ with

$x_0 + x_1 V + \ldots x_{\delta_i} V^{\delta_i}$ for all $i = 1, \ldots, v'$; the resulting expression is $\mathcal{E}'$. Note that $\mathcal{E}'$ is independent of the sequence of the substitutions (each substitution works on a disjoint portion of the DAG). It is not hard to show that $|\mathcal{E}'| = \mathcal{O}(\text{size}(\mathcal{E}))$, and consequently $\text{size}(\mathcal{E}') = \mathcal{O}(\text{size}(\mathcal{E}))$: note also that if $\mathcal{E}$ is c-bound then $\mathcal{E}'$ is also c-bound. By engaging in type-1 game with $\mathcal{E}'$, player A can "plug-in" all coefficients of her polynomials (along with $\alpha_1, \ldots, \alpha_r$) and therefore the type-2 game transforms to a type-1 game.

Theorem 65 *The correctness and security of our construction for type-1 games, implies the correctness and security of the type-2 protocol described above according to definition 60.*

We note that in general SNPGs are not produced by c-bound expressions: an expression for an SNPG is c-bound only if the number of polynomials contributed by both players is constant (constant nesting).

5.5 Security of Player A

In this section we will consider the security of player A in more detail. Our working assumption is that the Oblivious Transfer that is implemented in the second communication flow of the protocol is implemented in an ideal manner.

We denote by $\mathcal{S}^r_{n,k,t}$ the set of MPR instances with parameters r, n, k, t. For some $I \subseteq \{1, \ldots, n\}$ with $|I| = t$ we denote by $\mathcal{S}^r_{n,k,t}(I)$ the subset of $\mathcal{S}^r_{n,k,t}$ such that any $\mathcal{X} \in \mathcal{S}^r_{n,k,t}(I)$ has a solution of the form $\langle p_1, \ldots, p_r, I \rangle$, for some polynomials $p_1, \ldots, p_r$. Obviously, $\mathcal{S}^r_{n,k,t} = \cup_{|I|=t} \mathcal{S}^r_{n,k,t}(I)$. With similar arguments as in section 3.1.1 we can show that any instance of $\mathcal{S}^r_{n,k,t}$ has a unique solution with very high probability.

Let $\mathcal{D}$ be a probability distribution over $\mathbb{F}^r$. We expand the scope of $\mathcal{D}$ over $\mathcal{S}^r_{n,k,t}$ so that the values $\langle p_1(0), \ldots, p_r(0) \rangle$ are chosen according to $\mathcal{D}$ — and the remaining

elements of the instance at random (in a similar manner we can expand the scope of $\mathcal{D}$ over $\mathcal{S}^r_{n,k,t}(I)$ for some I). Given an instance $\mathcal{X} \in \mathcal{S}^r_{n,k,t}$ denote by $s(\mathcal{X})$ the tuple $\langle p_1(0), \ldots, p_r(0) \rangle$ where $\langle p_1, \ldots, p_r, I \rangle$ is the solution of $\mathcal{X}$. Using a similar argument as in the proof of lemma 18, we can show the following:

Lemma 66 *Let $\mathcal{D}$ be a probability distribution over $\mathbb{F}^r$. Let $\mathcal{A} : \mathcal{S}^r_{n,k,t} \to V$ be some PPT. Then it holds that there exists a PPT $\mathcal{A}'$ s.t. for all $v \in V$ and $I \subseteq \{1, \ldots, n\}$ with $|I| = t$.*

$$| \mathbf{Prob}_{\mathcal{X} \in_{\mathcal{D}} \mathcal{S}^r_{n,k,t}}[\mathcal{A}(\mathcal{X}) = v] - \mathbf{Prob}_{\mathcal{X} \in_{\mathcal{D}} \mathcal{S}^r_{n,k,t}(I)}[\mathcal{A}'(\mathcal{X}) = v] |$$

is negligible in n.

Proof. Similar to the proof of lemma 18. ■

Following the arguments for the PR-problem, the lemma above shows that the particular choice of the index-solution-set for a MPR instance does not affect the solvability of MPR.

The intractability assumption that we use is a direct generalization of the DPR (see also definition 19). First we define the notion of gap-predicates for the case of MPR:

Definition 67 *A pair of* PPT *predicates $\mathcal{A}_1, \mathcal{A}_2$ is called a gap-predicate-pair for the parameters n, k, t, r if for all $I \subseteq \{1, \ldots, n\}$ with $|I| = t$ it holds that:*

$$| \mathbf{Prob}[\mathcal{A}_1(i, \mathcal{X}) = 1] - \mathbf{Prob}[\mathcal{A}_2(i, \mathcal{X}) = 1] | = \begin{cases} \text{negligible} & \forall i \notin I \\ \text{non-negligible} & \text{for some } i \in I, \\ & i \leq n - k \end{cases}$$

where the probabilities are taken over all choices of $\mathcal{X} \in \mathcal{S}^r_{n,k,t}(I)$ and internal coin-tosses of the predicates $\mathcal{A}_1, \mathcal{A}_2$.

Decisional-MPR-Assumption. (DMPR$[n,k,t,r]$)

For any sound parameters $[n,k,t,r]$ there does not exist a gap-predicate-pair.

Soundness of parameters depends on the current state of the art algorithms against MPR as discussed in section 5.3.1.

The security of player A, depends on the hardness of partial information extraction of a MPR-instance (see definition 23). Following similar arguments as in section 3.2, we can show that MPR leaks no partial information based on the DMPR. The heart of the proof is the following lemma:

Lemma 68 *Suppose that there is a poly-time computable* $g : \mathbb{F} \to R$ *and a probability distribution* $\mathcal{D}$ *for which MPR with parameters* n,k,t,r *leaks partial information. Then there exists a* PPT $\mathcal{B}$ *such that for all* $I \subseteq \{1,\ldots,n\}$ *with* $|I| = t$, *if* $\beta_i(n) := \mathbf{Prob}_{\rho \in_U R, \mathcal{X} \in_U S^r_{n,k-1,t}(I)}[\mathcal{B}(i,\rho,\mathcal{X}) = 1]$ *with* $i \in \{0,\ldots,n\}$ *it holds that*

1. For all $i \notin I$ $|\beta_{i-1}(n) - \beta_i(n)|$ *is negligible.*

2. There exists an $i_0 \in I$ *such that* $|\beta_{i_0-1}(n) - \beta_{i_0}(n)|$ *is non-negligible and* $i_0 \leq n-k+1$.

Proof. Similar to the proof of lemma 24. ■

As a result, it is possible to transform any adversary playing the role of player B, that extracts some partial information of A's secrets from the protocol transcript, into a gap-predicate-pair for the MPR problem, something that violates the DMPR assumption with parameters $[n,k-1,t,r]$.

Corollary 69 *If* B *breaks the security of* A *in our protocol then, assuming that the underlying t-out-of-n OT is secure,* DMPR *is violated for parameters* $n,k := \min_\ell \frac{d}{rd_\ell}, t := 2d+1, r$.

5.6 Security of Player B

Ideal Implementation. Description of the operation of the trusted third party C in the ideal implementation: C receives from player A $\{\langle z_i, y_{i,1}, \ldots, y_{i,r}\rangle\}_{i=1}^{n}$ and $I \subseteq \{1, \ldots, n\}$ that corresponds to the choice of A in the oblivious transfer operation — note that for simplicity throughout the proof we assume that $I = \{1, \ldots, t\}$. C receives also $P_1, \ldots, P_v$ from player B. For simplicity denote by $P(x_1, \ldots, x_r)$ the polynomial $\mathcal{E}(P_1, \ldots, P_v)(x_1, \ldots, x_r)$.

C interpolates the polynomials $b_1, \ldots, b_r$ that correspond to the positions of I in the given input by player A, i.e. $b_\ell(z_i) = y_{i,\ell}$ for $i = 1, \ldots, t, \ell = 1, \ldots, r$; let $\beta_\ell = \text{degree}(b_\ell)$. Define $S(x) := P(b_1(x), \ldots, b_r(x))$ and $B(x) = b_1^{d_1}(x) \ldots b_r^{d_r}(x)$. It holds that $m := \text{degree}(B) = \beta_1 d_1 + \ldots + \beta_r d_r \geq \text{degree}(S)$. In the case $m \leq d$, C returns to A the value $R(0)$. If $m > d$, consider the following array:

$$M := \left(\begin{array}{c|c|c|c|c|c|c|c} z_1 & z_1^2 & \ldots & z_1^d & B(z_1)z_1 & B(z_1)z_1^2 & \ldots & B(z_1)z_1^d \\ z_2 & z_2^2 & \ldots & z_2^d & B(z_2)z_2 & B(z_2)z_2^2 & \ldots & B(z_2)z_2^d \\ \vdots & \vdots & \ldots & \vdots & \vdots & \vdots & \ldots & \vdots \\ z_t & z_t^2 & \ldots & z_t^d & B(z_t)z_t & B(z_t)z_t^2 & \ldots & B(z_t)z_t^d \end{array}\right)$$

The dimensions of M are $(2d+1) \times (2d)$ (note that $t = 2d + 1$). Because of the fact that $m = \beta_1 d_1 + \ldots + \beta_r d_r > d$ it follows easily that M is of full rank: $2d$. As a result it is possible to eliminate the first row in the matrix by multiplying by some $\lambda_2, \ldots, \lambda_t$ the remaining rows, and adding all of them to the first row. C returns to A the value $g := S(z_1) + \lambda_2 S(z_2) + \ldots + \lambda_t S(z_t)$.

Algorithm $\mathcal{G}$. Goal: given the output of the ideal implementation and the values selected by player A (together with A's randomness) generate a protocol transcript. First we compute the polynomials $b_1, \ldots, b_r$ and the value m as before.

Case 1. $m = d$. We are given a value $V(:= S(0) = P(b_1(0), \ldots, b_r(0))$ and the input $\{\langle z_i, y_{i,1}, \ldots, y_{i,r}\rangle\}_{i=1}^{n}$ with $I \subseteq \{1, \ldots, n\}$, $|I| = t$.

We select $2d$ random values $q_2, \ldots, q_t$ and we solve the following system for the unknowns $a_1, \ldots, a_{t-1}, x$:

$$\begin{pmatrix} 1 & z_1 & z_1^2 & \ldots & z_1^{t-1} \\ 1 & z_2 & z_2^2 & \ldots & z_2^{t-1} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 1 & z_t & z_t^2 & \ldots & z_t^{t-1} \end{pmatrix} \cdot \begin{pmatrix} V \\ a_1 \\ \vdots \\ a_{t-1} \end{pmatrix} = \begin{pmatrix} x \\ q_2 \\ \vdots \\ q_t \end{pmatrix}$$

or equivalently:

$$\begin{pmatrix} -1 & z_1 & z_1^2 & \ldots & z_1^{t-1} \\ 0 & z_2 & z_2^2 & \ldots & z_2^{t-1} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 0 & z_t & z_t^2 & \ldots & z_t^{t-1} \end{pmatrix} \cdot \begin{pmatrix} x \\ a_1 \\ \vdots \\ a_{t-1} \end{pmatrix} = \begin{pmatrix} -V \\ q_2 - V \\ \vdots \\ q_t - V \end{pmatrix}$$

The transcript in this case will be the $(2d+1)$-vector containing $q_1 = x, q_2, \ldots, q_t$.

Case 2. $m < d$. We are given a value $V(:= S(0) = P(b_1(0), \ldots, b_r(0))$ and the input $\{\langle z_i, y_{i,1}, \ldots, y_{i,r}\rangle\}_{i=1}^n$ with $I \subseteq \{1, \ldots, n\}$, $|I| = t$. We select $d+m$ random values $q_{d-m+2}, \ldots, q_t$ and we solve the following systems of equations for the unknowns $a_1^j, \ldots, a_{d+m}^j, x_j$, $j = 1, \ldots, d-m+1$:

$$\begin{pmatrix} 1 & z_j & z_j^2 & \ldots & z_j^{t-1} \\ 1 & z_{d-m+2} & z_{d-m+2}^2 & \ldots & z_{d-m+2}^{d+m} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 1 & z_t & z_t^2 & \ldots & z_t^{d+m} \end{pmatrix} \cdot \begin{pmatrix} V \\ a_1^j \\ \vdots \\ a_{d+m}^j \end{pmatrix} = \begin{pmatrix} x_j \\ q_{d-m+2} \\ \vdots \\ q_t \end{pmatrix}$$

or equivalently:

$$\begin{pmatrix} -1 & z_j & z_j^2 & \ldots & z_j^{d+m} \\ 0 & z_{d-m+2} & z_{d-m+2}^2 & \ldots & z_{d-m+2}^{d+m} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 0 & z_t & z_t^2 & \ldots & z_t^{d+m} \end{pmatrix} \cdot \begin{pmatrix} x_j \\ a_1^j \\ \vdots \\ a_{d+m}^j \end{pmatrix} = \begin{pmatrix} -V \\ q_{d-m+2} - V \\ \vdots \\ q_t - V \end{pmatrix}$$

The transcript in this case will be the $(2d+1)$-vector containing

$$q_1 = x_1, \ldots, q_{d-m+1} = x_{d-m+1}, q_{d-m+2}, \ldots, q_t$$

Case 3. $m > d$. In this case we are given the value $g(:= S(z_1) + \lambda_2 S(z_2) + \ldots + \lambda_t S(z_t)$. We select random $q_2, \ldots, q_t$. We compute the $\lambda_2, \ldots, \lambda_t$ values from the matrix M (as in the ideal implementation). We compute an x such that $x + \lambda_2 q_2 + \ldots + \lambda_t q_t = g$.

The transcript will be the $(2d+1)$-vector containing $q_1 = x, q_2, \ldots, q_t$.

Claim. The transcripts generated as above are indistinguishable from legitimate protocol transcripts.

Regular protocol transcripts are generated as follows: choose a random $(2d)$-vector $v := \langle a_1, \ldots, a_d, b_1, \ldots, b_d \rangle$. Define M as in the ideal implementation. Let f be the vector $\langle S(z_1), \ldots, S(z_t) \rangle$. Then the transcript generated by the protocol is the $(2d+1)$-vector $q^T := f^T + M \cdot v^T$. Alternatively we may view the transcript as the vector $\langle R(z_1), \ldots, R(z_t) \rangle$ where $R(x) := Q(x, b_1(x), \ldots, b_r(x)) = S(x) + \sum_{i=1}^{d} a_i x^i + B(x) \sum_{i=1}^{d} b_i x^i$.

Claim A. Any protocol transcript can be generated by $\mathcal{G}$.

Proof. Let $q' := \langle q'_1, \ldots, q'_t \rangle$ be a protocol transcript for the polynomials $P, b_1, \ldots, b_r$ and values $z_1, \ldots, z_t$. We will prove that $\mathcal{G}$ can generate this transcript.

First assume that $m = d$. We have to show that there is a choice of randomness for $\mathcal{G}$ so that the generated transcript is equal to q'. We select randomness such that $q_2 = q'_2, \ldots, q_t = q'_t$. Subsequently for this randomness we have to show that the value x computed by $\mathcal{G}$ is exactly q'_1. The value x is computed by the following system:

$$\begin{pmatrix} 1 & z_1 & z_1^2 & \ldots & z_1^{t-1} \\ 1 & z_2 & z_2^2 & \ldots & z_2^{t-1} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 1 & z_t & z_t^2 & \ldots & z_t^{t-1} \end{pmatrix} \cdot \begin{pmatrix} V \\ a_1 \\ \vdots \\ a_{t-1} \end{pmatrix} = \begin{pmatrix} x \\ q_2 \\ \vdots \\ q_t \end{pmatrix}$$

so we want to show that if x is substituted by q'_1 the equality remains solvable.

$$\begin{pmatrix} 1 & z_1 & z_1^2 & \dots & z_1^{t-1} \\ 1 & z_2 & z_2^2 & \dots & z_2^{t-1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & z_t & z_t^2 & \dots & z_t^{t-1} \end{pmatrix} \cdot \begin{pmatrix} V \\ a_1 \\ \vdots \\ a_{t-1} \end{pmatrix} = \begin{pmatrix} q'_1 \\ q'_2 \\ \vdots \\ q'_t \end{pmatrix}$$

or equivalently that

$$\begin{pmatrix} 1 & z_1 & z_1^2 & \dots & z_1^{t-1} \\ 1 & z_2 & z_2^2 & \dots & z_2^{t-1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & z_t & z_t^2 & \dots & z_t^{t-1} \end{pmatrix} \cdot \begin{pmatrix} R(0) \\ a_1 \\ \vdots \\ a_{t-1} \end{pmatrix} = \begin{pmatrix} R(z_1) \\ R(z_2) \\ \vdots \\ R(z_t) \end{pmatrix}$$

which is of course solvable. since $R(x)$ is a polynomial of degree $2d$.

Now consider the case $m < d$. The randomness used by $\mathcal{G}$ now is only $d + m$ elements. We select the randomness $q_{d-m+2} = q'_{d-m+2}, \dots, q_t = q'_t$. We want to verify that the values $x_1, \dots, x_{d-m+1}$ defined by $\mathcal{G}$ are exactly equal to the values $q'_1, \dots, q'_{d-m+1}$. In other words we want to show that for any $j = 1, \dots, d - m + 1$ if we substitute the value x_j by q'_j in the following system. it remains solvable.

$$\begin{pmatrix} 1 & z_j & z_j^2 & \dots & z_j^{t-1} \\ 1 & z_{d-m+2} & z_{d-m+2}^2 & \dots & z_{d-m+2}^{d+m} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & z_t & z_t^2 & \dots & z_t^{d+m} \end{pmatrix} \cdot \begin{pmatrix} V \\ a_1^j \\ \vdots \\ a_{d+m}^j \end{pmatrix} = \begin{pmatrix} x_j \\ q'_{d-m+2} \\ \vdots \\ q'_t \end{pmatrix}$$

if we substitute x_j by q'_j and V by $Q(0, b_1(0), \dots, b_r(0))$ then we have:

$$\begin{pmatrix} 1 & z_j & z_j^2 & \dots & z_j^{t-1} \\ 1 & z_{d-m+2} & z_{d-m+2}^2 & \dots & z_{d-m+2}^{d+m} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & z_t & z_t^2 & \dots & z_t^{d+m} \end{pmatrix} \cdot \begin{pmatrix} R(0) \\ a_1^j \\ \vdots \\ a_{d+m}^j \end{pmatrix} = \begin{pmatrix} R(z_j) \\ R(z_{d-m+2}) \\ \vdots \\ R(z_t) \end{pmatrix}$$

which is of course solvable for all $j = 1, \dots, d - m + 1$ since $R(x)$ is a polynomial of degree $d + m$.

Finally consider the case $m > d$. The randomness used by $\mathcal{G}$ is $2d$ elements: we select them $q_2 = q'_2, \dots, q_t = q'_t$. We want to show now that the value x as defined by

$\mathcal{G}$ is equal to q'_1. $\mathcal{G}$ defines x as follows: $x := g - \lambda_2 q_2 - \ldots - \lambda_t q_t$, where $\lambda_2, \ldots, \lambda_t$ are define using matrix M. So we want to show that $x = q'_1$ or equivalently that $q'_1 = g - \lambda_2 q'_2 - \ldots - \lambda_t q'_t$. i.e. $g = q'_1 + \lambda_2 q'_2 + \ldots \lambda_t q'_t$ (since $q'_2 = q_2, \ldots, q'_t = q_t$). Recall that q'_j is defined as $(f^T)_j + (M \cdot v^T)_j$ i.e.

$$q'_j = S(z_j) + \sum_{i=1}^{d} a_i z_j^i + B(z_j) \sum_{i=1}^{d} b_i z_j^i$$

where $a_1, \ldots, a_d, b_1, \ldots, b_d$ is the randomness selected for the generation of the protocol transcript (by player B).

By the definition of $\lambda_2, \ldots, \lambda_t$ we have that

$$\lambda_2 z_2^i + \ldots \lambda_t z_t^i = -z_1^i. \quad i = 1, \ldots, d$$

and

$$\lambda_2 B(z_2) z_2^i + \ldots + \lambda_t B(z_t) z_t^i = -B(z_1) z_1^i$$

As a result

$$\begin{gathered} q'_1 + \lambda_2 q'_2 + \ldots \lambda_t q'_t = \\ S(z_1) + \sum_{i=1}^{d} a_i z_1^i + B(z_1) \sum_{i=1}^{d} b_i z_1^i \\ +\lambda_2 \Big(S(z_2) + \sum_{i=1}^{d} a_i z_2^i + B(z_2) \sum_{i=1}^{d} b_i z_2^i \Big) \\ + \ldots \\ +\lambda_t \Big(S(z_t) + \sum_{i=1}^{d} a_i z_t^i + B(z_t) \sum_{i=1}^{d} b_i z_t^i \Big) \\ = S(z_1) + \lambda_2 S(z_2) + \ldots + S(z_t) = g \end{gathered}$$

Claim B. Any "transcript" generated by $\mathcal{G}$ can be generated as a protocol transcript. Let q be the output of $\mathcal{G}$ for some polynomials $P, b_1, \ldots, b_r$ and values $z_1, \ldots, z_t$. We will specify the randomness $\langle a_1, \ldots, a_d, b_1, \ldots, b_d \rangle$ that has to be selected in the

transcript generation so that the generated transcript $q' = \langle q'_1, \ldots, q'_t \rangle$ is exactly equal to $q = \langle q_1, \ldots, q_r \rangle$. This means that the randomness $v = \langle a_1, \ldots, a_d, b_1, \ldots, b_d \rangle$ used by player B in the generation of the transcript should be such that $f^T + M \cdot v^T = q^T$, or equivalently that $M \cdot v^T = q^T - f^T$ (a system of $2d + 1$ equations).

Consider the case $m = d$. In this case M is a matrix of full rank $2d$, and in particular if we denote by M_t^2 the square matrix that results from the exclusion of the first row of M: it is easy to see that M_t^2 is of full rank. As a result we can define $v = \langle a_1, \ldots, a_d, b_1, \ldots, b_d \rangle$ to be the solution of the system: $M_t^2 \cdot v^T = [q_t^2]^T - [f_t^2]^T$, where $[q_t^2]$ and $[f_t^2]$ refer to the vectors v and f with their first value excluded.

To complete this step we have to show that with the given randomness as defined by the system the value q'_1 defined as $S(z_1) + \sum_{i=1}^{d} a_i z_1^i + B(z_1) \sum_{i=1}^{d} b_i z_1^i$ will be equal to q_1 as defined by the algorithm $\mathcal{G}$. $\mathcal{G}$ defines $q_1 = x$ as follows:

$$x := \begin{vmatrix} -V & z_1 & \ldots & z_1^{t-1} \\ q_2 - V & z_2 & \ldots & z_2^{t-1} \\ \vdots & \vdots & \ldots & \vdots \\ q_t - V & z_t & \ldots & z_t^{t-1} \end{vmatrix} \Big/ \begin{vmatrix} -1 & z_1 & \ldots & z_1^{t-1} \\ 0 & z_2 & \ldots & z_2^{t-1} \\ \vdots & \vdots & \ldots & \vdots \\ 0 & z_t & \ldots & z_t^{t-1} \end{vmatrix}$$

where $V := S(0) = P(b_1(0), \ldots, b_r(0))$. This is equivalent to

$$\begin{vmatrix} x - V & z_1 & \ldots & z_1^{t-1} \\ q_2 - V & z_2 & \ldots & z_2^{t-1} \\ \vdots & \vdots & \ldots & \vdots \\ q_t - V & z_t & \ldots & z_t^{t-1} \end{vmatrix} = 0$$

Consequently we want to verify that for $q'_1 = S(z_1) + \sum_{i=1}^{d} a_i z_1^i + B(z_1) \sum_{i=1}^{d} b_i z_1^i$, it holds that:

$$\begin{vmatrix} q'_1 - V & z_1 & \ldots & z_1^{t-1} \\ q_2 - V & z_2 & \ldots & z_2^{t-1} \\ \vdots & \vdots & \ldots & \vdots \\ q_t - V & z_t & \ldots & z_t^{t-1} \end{vmatrix} = 0$$

The choice of $a_1, \ldots, a_d, b_1, \ldots, b_d$ was made so that $q_2 = q'_2, \ldots, q_t = q'_t$; as a result the equation above can be written as follows

$$\begin{vmatrix} R(z_1)-R(0) & z_1 & \dots & z_1^{t-1} \\ R(z_2)-R(0) & z_2 & \dots & z_2^{t-1} \\ \vdots & \vdots & \dots & \vdots \\ R(z_t)-R(0) & z_t & \dots & z_t^{t-1} \end{vmatrix} = 0$$

where $R(x) = Q(x, b_1(x), \dots, b_r(x))$, recall that $Q(x_0, x_1, \dots, x_r) := P(x_1, \dots, x_r) + \sum_{i=1}^{d} a_i x_0^i + x_1^{d_1} \dots x_r^{d_r} \sum_{i=1}^{d} b_i x_0^i$ (note that $R(0) = S(0) = V$). It is easy to see that the degree of R is at most $2d = t - 1$. As a result the determinant above is equal to 0.

Consider the case $m < d$. In this case M is not of full rank, and in particular it is of rank $d + m$. Denote by M_t^u where $u := d - m + 2$ a square matrix belonging in the lower $d + m$ rows of M of rank $d + m$ (it is easy to see that such a matrix exists). Let v' be a $d + m$ subvector of v that corresponds to the columns of M_t^u and denote by v'' the remaining elements of v. Also denote by $\overline{M}_t^u$ the remaining $d - m$ columns of length $d + m$ that correspond to whatever is left in the lower $d + m$ rows of M after the removal of M_t^u. We select v'' at random; given v'' we define v' from the system: $M_t^u \cdot [v']^T = [q_t^u]^T - [f_t^u]^T - \overline{M}_t^u [v'']^T$, the notation q_t^u (resp. f_t^u) means the rightmost $d + m$ values of the vector v (resp. f).

We want to show that for the computation of $v = \langle a_1, \dots, a_d, b_1, \dots, b_d \rangle$ as described in the previous paragraph it holds that the values $q_j' = R(z_j) = S(z_j) + \sum_{i=1}^{d} a_i z_j^i + B(z_j) \sum_{i=1}^{d} b_i z_j^i$, for $j = 1, \dots, d - m + 1$ are equal to the values $q_j = x_j$ defined by $\mathcal{G}$ as follows:

$$q_j := \begin{vmatrix} -V & z_j & \dots & z_j^{d+m} \\ q_2 - V & z_u & \dots & z_u^{d+m} \\ \vdots & \vdots & \dots & \vdots \\ q_t - V & z_t & \dots & z_t^{d+m} \end{vmatrix} \Big/ \begin{vmatrix} -1 & z_j & \dots & z_j^{d+m} \\ 0 & z_u & \dots & z_u^{d+m} \\ \vdots & \vdots & \dots & \vdots \\ 0 & z_t & \dots & z_t^{d+m} \end{vmatrix}$$

where $V (:= S(0) = P(b_1(0), \dots, b_r(0)))$. This is equivalent to

$$\begin{vmatrix} q_j - V & z_j & \dots & z_j^{d+m} \\ q_2 - V & z_u & \dots & z_u^{d+m} \\ \vdots & \vdots & \dots & \vdots \\ q_t - V & z_t & \dots & z_t^{d+m} \end{vmatrix} = 0$$

If $R(x) := Q(x, b_1(x), \dots, b_r(x)) = S(x) + \sum_{i=1}^{d} a_i x^i + B(x) \sum_{i=1}^{d} b_i x^i$, R is of degree at most $d + m$. By the choice of $v = \langle a_1, \dots, a_d, b_1, \dots, b_d \rangle$, it holds that $R(z_i) = q_i$ for $i = u, \dots, t$.

As a result we want to show that, for all $j = 1, \dots, d - m + 1$,

$$\begin{vmatrix} R(z_j) - R(0) & z_j & \dots & z_j^{d+m} \\ R(z_u) - R(0) & z_u & \dots & z_u^{d+m} \\ \vdots & \vdots & \dots & \vdots \\ R(z_t) - R(0) & z_t & \dots & z_t^{d+m} \end{vmatrix} = 0$$

which is true since R is of degree at most $d + m$.

Finally consider the case $m > d$. In this case M is a matrix of full rank $2d$, and in particular if we denote by M_t^2 the square matrix that results from the exclusion of the first row of M it is easy to see that M_t^2 is of full rank. As a result we can define $v = \langle a_1, \dots, a_d, b_1, \dots, b_d \rangle$ to be the solutions of the system: $M_t^2 \cdot v^T = [q_t^2]^T - [f_t^2]^T$, where $[q_t^2]$ and $[f_t^2]$ refer to the vectors v and f with their first value excluded.

To complete this step we have to show that with the given randomness as defined by the system the value q_1' defined as $S(z_1)) + \sum_{i=1}^{d} a_i z_1^i + B(z_1) \sum_{i=1}^{d} b_i z_1^i$ will be equal to q_1 as defined by the algorithm $\mathcal{G}$. $\mathcal{G}$ defines $q_1 = x$ as follows:

$$x = S(z_1) + \lambda_2(S(z_2) - q_2) + \dots + \lambda_t(S(z_t) - q_t)$$

Since we want to show that $q_1' = q_1$, we have to show that for the choice of $a_1, \dots, a_d, b_1, \dots, b_d$ we made it holds that

$$q_1' = S(z_1) + \sum_{i=1}^{d} a_i z_1^i + B(z_1) \sum_{i=1}^{d} b_i z_1^i$$

$$= S(z_1) + \lambda_2(S(z_2) - q_2) + \ldots + \lambda_t(S(z_t) - q_t)$$

but the choice of $a_1, \ldots, a_d, b_1, \ldots, b_d$ we have made implies that $q_l = S(z_l) + \sum_{i=1}^{d} a_i z_l^i + B(z_l) \sum_{i=1}^{d} b_i z_l^i$ for $l = 2, \ldots, t$. As a result we have to show that:

$$-\sum_{i=1}^{d} a_i z_1^i + B(z_1) \sum_{i=1}^{d} b_i z_1^i$$

$$= \lambda_2(\sum_{i=1}^{d} a_i z_2^i + B(z_2) \sum_{i=1}^{d} b_i z_2^i) + \ldots + \lambda_t(\sum_{i=1}^{d} a_i z_t^i + B(z_t) \sum_{i=1}^{d} b_i z_t^i)$$

something that follows immediately by the definition of $\lambda_2, \ldots, \lambda_t$.

To complete the section we overview our proof technique in figure 5.2.

5.7 Oblivious Bargaining and Oblivious-Strategy Negotiations

In this section, we present two examples of an interaction that can be modeled by type-2 SPGEs. In *Oblivious Bargaining*, A is the prospective buyer of a product or services and B is the seller. A wants to make a monetary offer to B without revealing the exact amount initially. B has a polynomial f that given an amount, returns the acceptance rate for the transaction (in some predetermined range). A has also a polynomial g that given an acceptance rate makes a counter offer. Players do not want to reveal much information about their bargaining strategies, therefore a round by round polynomial evaluation leaks too much information. A and B can decide on a number of bargaining rounds, and using the SNPG protocol A obtains $f(g(f(\ldots(f(\alpha))\ldots)))$, where α is the initial offer by A. Based on the result, A can either proceed to buying the product (the polynomials will be revealed) or abort the transaction. The security of our protocol implies that in the case the transaction is aborted, the bargaining strategies and the initial offer will remain secret. It should

be noted that players may choose different bargaining strategies for each round (e.g. compromise in the last rounds) or even bargain simultaneously for many different things (multivariate SNPGs).

In an *Oblivious-Strategy Negotiation* player A has a proposal α and wants to obtain the evaluation of her proposal without revealing much information about the proposal or her tactics. Consider the following setting: A is a representative of a company that makes a proposal α on a specific project. B as the representative of the funding company has two polynomial functions: f_0 that gives the approximate level of funding for a given proposal, and f_1 that gives the approval percentage based on how much of the project will be completed in a given time period. A, has also a polynomial function g_1 that given the funds, approximates the percentage of the project that will be completed in a given time period. Engaging in a SNPG, A obliviously obtains the value $f_1(g_1(f_0(\alpha)))$ that is the approval percentage for the proposal α. Note that our security specifications ensure that B is not revealing the funding level assigned to A's proposal, but merely the final acceptance rate based on the percentage of the project completed in the given time. The same offer can be used by A to obtain approval rates by other prospective funders in the place of B. After the end of all games, A will proceed to make an open offer to the funder that gave the highest approval rate.

5.8 Oblivious Affine Evaluations

Consider an expression $\mathcal{E}(f, p_1, \ldots, p_r)$ where f is restricted to be a publicly known affine transformation, $f(x_1, \ldots, x_r) = a_1x_1 + \ldots a_rx_r + a$, with $a_1, \ldots, a_r, a \in \mathbb{F}$, and $\mathcal{E}(f, p_1, \ldots, p_r) := f(p_1(x_1), \ldots, p_r(x_r))$. This family of SGPEs, which we call Oblivious Affine Evaluations, has a variety of applications of which we will discuss two. We note that it is possible to efficiently reduce an Oblivious Affine Evaluation to the OPE problem achieving secret-independence at the same time. Consequently

Oblivious Affine Evaluations is a family of SGPEs that can be solved successfully by both our construction and the OPE construction in [NP99]. In the applications that we will present below $a_1 = \ldots = a_r = 1$ and $a = 0$.

In "Oblivious Scoring", Alice has answered in r "sensitive" questions. Each question accepts as an answer a numerical value α_ℓ in $\mathbb{F}$. Bob is the evaluator that assigns a score to each of the answers of Alice. Bob evaluates the answer to the ℓ-th question with a polynomial p_ℓ. So $p_\ell(\alpha_\ell)$ is the score assigned by Bob to the answer α_ℓ for question ℓ. In order to avoid wrap-around the boundaries of the field when adding scores, we require that $\forall a \in A_\ell(-\lfloor|\mathbb{F}|/r\rfloor \leq p_\ell(a) \leq \lfloor|\mathbb{F}|/r\rfloor)$ where A_ℓ is the set of possible answers for question ℓ — alternatively we can make the addition of the scores in a sufficiently large extension of the field $\mathbb{F}$. Alice wants to obtain Bob's overall evaluation of her answers (probably pays for that) but she does not want to reveal them as she considers them private. On the other hand, Bob agrees to evaluate Alice's answers, but he does not want to reveal the evaluation functions. By engaging in a Oblivious Affine Evaluation, Alice can obliviously obtain her overall score: $\sum_{\ell=1}^{r} p_\ell(\alpha_\ell)$.

5.9 Relations Between Basic Primitives

In this section, for completeness, we compare 1-out-of $|S|$ OT, OPE and SMPE as abstract problems and we investigate relations among them. Note that in our implementation – as well as in the implementation of OPE in [NP99], OT was actually used as a subroutine. This however does not rule out a construction for SMPE (or OPE) without using OT. The following reductions are also of practical interest: especially case (ii) of the theorem clarifies that there are applications for OPE and SMPE that are not worth actualizing using these protocols but rather rely to a standard OT taking advantage of our reduction.

Theorem 70 *(i)* OT *reduces to* OPE *and* OPE *reduces to* SMPE. *Consequently* OPE *and* SMPE *are complete for secure function evaluation. (ii) If the set* S *of the possible choices of* A *is "small" and is known to* B, *then 1-out-of* $|S|$ OT, OPE *and* SMPE *are equivalent.*

Proof. (i) The fact that OPE reduces to SMPE is immediate since for $r = 1$, SMPE is exactly OPE.

OT reduces to OPE: The elements of S accept some ordering known to both parties. B computes a "selector" polynomial p with the following property: $p(i) = \alpha$ iff i is the index of α in the (ordered) set S. The degree of p is at most $|S| - 1$. Suppose A wants to get the i-th element of S with a 1-out-of $|S|$ OT: she can use OPE instead and obliviously compute $p(i)$.

(ii). Reducing OPE to OT:

- Each element α in S is assigned a number in $\{1, \ldots, |S|\}$ denoted by $\overline{\alpha}$. This ordering is input to both A and B.
- B computes $p(\alpha)$ for all $\alpha \in S$, and orders these values with the following criterion: $p(\alpha_1) < p(\alpha_2)$ iff $\overline{\alpha_1} < \overline{\alpha_2}$. Denote this set by S_p.
- Suppose A's input for OPE is α: using 1-out-of $|S|$ OT, A obliviously obtains the element of S_p with index $\overline{\alpha}$; clearly A gets the value $p(\alpha)$.

Reducing SMPE to OT is done in a similar manner, and using part *(i)* we deduce that they are all equivalent. ■

Note that "small" means that the reduction involves a $\Theta(|S|)$ computation ($|S|$ is exponentially large in the general case).

For a given random choice of P (according to any distribution) and $b_1, \ldots, b_r$ as defined by the private input of player A the following are true:

- For any legitimate protocol transcript it is possible to define the randomness used by algorithm $\mathcal{G}$ so that $\mathcal{G}$ outputs exactly this legitimate protocol transcript.
- For any "transcript" generated by $\mathcal{G}$, it is possible to define the randomness used by player B so that the resulting protocol transcript is equal to the "transcript" generated by $\mathcal{G}$ (provided that the t-out-of-n OT is ideal).

Let any PPT $\mathcal{A}'$ that takes as input a protocol transcript and any other information available to player A (how the query to player B was generated etc.). Then there exists a PPT $\mathcal{A}''$ that takes as input only the value given to player A in the ideal implementation and the information available to player A so that the output of $\mathcal{A}''$ is indistinguishable from that of $\mathcal{A}'$. The description of $\mathcal{A}''$ is simple: first it uses $\mathcal{G}$ to generate a "transcript" with the information it has from the ideal implementation and then it simulates $\mathcal{A}'$. The output of $\mathcal{A}''$ is indistinguishable from that of $\mathcal{A}'$ since the "transcripts" generated by $\mathcal{G}$ are indistinguishable from protocol transcripts provided that the t-out-of-n OT is ideal.

Figure 5.2: Outline for the proof of the security of player B.

Chapter 6

Relations to Coding

6.1 Randomized Decoding of Interleaved Reed Solomon Codes

Random noise assumptions have been considered extensively in the coding theory literature with substantial results. One prominent example is Forney Codes [For66] that were designed over the binary symmetric channel (BSC). The BSC suggests that when transmitting binary digits, errors are independent and every bit transmitted has a fixed probability of error. The BSC provides a form of a random noise assumption, which allowed probabilistic decoding for message rates that approach the capacity of the channel.

Worst-case non-ambiguous decoding (i.e., when only a bound on the number of faults is assumed and a unique solution is required) has a natural limitation of correcting a number of errors that is up to half the distance of the code. Going beyond this natural bound, either requires re-stating the decoding problem (e.g. consider list-decoding: output all possible decodings for a corrupted codeword), or assuming some "noise assumption" that will restrict probabilistically the combinatorial possibilities for a multitude of possible solutions. Typically, such assumptions are associated with physical properties of given channels (e.g., bursty noise, etc.). Recent breakthrough

results by Guruswami and Sudan in list-decoding ([Sud97, GS98]) showed that decoding beyond the natural error-correction bound is possible in the worst-case, by outputting all possible decodings. Naturally, there are still limitations in the case of worst-case decoding that prohibit the decoding of very high error-rates.

In this section, motivated by the above, we consider a channel model that is native to the non-binary setting. In particular we employ a "Non-Binary Symmetric Channel" (NBSC), presented in figure 6.1.

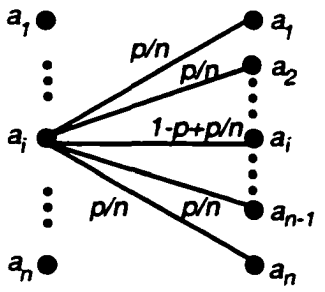

Figure 6.1: A non-binary symmetric channel over an alphabet of n symbols. The probability of successful transmission is $1 - p + p/n$. We will refer to p as the error-rate of the NBSC.

As a channel model for bit-level transmission the Non-Binary Symmetric Channel model applies to settings where aggregates of bits are sent and errors are assumed to be bursty. Thus, in contrast with the Binary Symmetric Channel, errors in consecutive bits are assumed to be correlated. There are additional settings where the NBSC describes the transmission model. For example consider the case of Information Dispersal Algorithms introduced by Rabin in [Rab89] for omission errors, and extended by Krawczyk [Kra93] to deal with general errors. In this setting, a word is encoded into a codeword and various portions of the codeword are sent over different channels, some of which may introduce errors. In the case where the channels are radio channels (i.e., operating in different frequencies), errors may be introduced by

jammed channels which emit white noise. Namely, they randomize the transmitted symbol. As a result the communication model in this case approximates the NBSC. Another setting which approximates the NBSC is the transmission of encrypted data where each sub codeword is sent encrypted with what is called "error propagation encryption mode." These popular modes (e.g. the CBC mode), over noisy channels, will produce a transmission that also approximates the NBSC model ([MVV97], page 230). Finally, the NBSC model is essential for the security of player A in our protocol for secure games with polynomial expressions described in chapter 5.

In this work, we concentrate on the "code interleaving" encoding schema, see e.g. section 7.5, [VV89], which is a technique to increase the robustness of a code in the setting of burst errors. Here we take advantage of code-interleaving in the NBSC model to allow decoding with increased error-correction potential. In particular, we consider the case of interleaved Reed-Solomon Codes and we propose a new probabilistic decoding algorithm that takes advantage of interleaving in the NBSC model. While typically error-correction in interleaved codes treats each interleaved codeword independently, our algorithm attempts to correct all codewords *simultaneously*. We show that our algorithm achieves a high error-correction rate (namely, any rate which is bounded away from the error-rate that totally randomizes the transmitted word) in the NBSC model. Our decoding algorithm can be seen as a probabilistic "multi-dimensional" extension of the Berlekamp-Welch RS-decoding algorithm [BW86]. The analysis of the success probability of our algorithm introduces a "minor matrix decomposition" technique that, together with Schwartz's Lemma [Sch80] for bounding the number of roots of multivariate polynomials, provides an effective way to show the uniqueness and efficiency of the reconstruction procedure.

For the generic coding theoretic problem, our decoding method for interleaved Reed-Solomon codes provides an improved coding solution for the case of Non-Binary Symmetric Channel model. In figure 6.2, we compare our result with known worst-

case decoding algorithms.

	Berlekamp-Welch [BW86]	Guruswami-Sudan [GS98]	Our Decoding Algorithm
Max. error-rate allowed	$\frac{1-\kappa}{2}$	$1-\sqrt{\kappa}$	$1-\kappa-\alpha$

Figure 6.2: Comparison of the maximum error-rate allowed in the case of interleaved RS-Codes: κ is the message-rate and $\alpha \in Q^+$ is an arbitrary constant. We emphasize that our results hold in the Non-Binary Symmetric Channel model (whereas the other two algorithms are in the worst-case model, thus more general) and that we assume "large enough" finite field size.

We note that employing our methodology, setting and analysis techniques in other cases (i.e. simultaneous decoding of all interleaved codewords for other families of interleaved codes in the NBSC model) is an interesting research direction.

6.1.1 The Basic Problem

Below we state informally the basic coding theory problem which we deal with in this work (with emphasis on the large alphabet):

Basic Problem. Transmit k-length words from large alphabet Σ, with message-rate κ and maximize the error-correction capability. We will work on this problem, not in the worst-case, but rather in the Non-Binary Symmetric Channel as it was described in the introduction. Specifically,

The NBSC Model. The probability of correct transmission of a symbol from an alphabet of n symbols is $1-p+p/n$, and the probability of receiving some of the remaining symbols from the alphabet is p/n. We will refer to p as the "error-rate" of the channel.

More formally, we state the general coding theoretic problem, for a certain alphabet Σ. For a given message rate $\kappa := k/n$ we need to provide efficient encoding and decoding algorithms enc, dec so that the following are satisfied:

- The encoding function $enc : \Sigma^k \rightarrow \Sigma^n$ is 1-1 and maps strings of length k to strings of length n.

- There is some $0 < \epsilon < 1$ so that the decoding success probability $\mathbf{Prob}[dec(x) = m]$ is "substantial", where $x \in \mathcal{B}(enc(m), e)$ with $e/n \leq \epsilon$. The notation $\mathcal{B}(y, e))$ denotes the Hamming ball of radius e around y, i.e. $\mathcal{B}(y, e)) := \{z \mid \mathrm{d}(z, y) \leq e\}$, where $\mathrm{d}(z, y)$ denotes the Hamming distance of the two strings z, y (i.e. the number of positions they differ).

The natural bound on the worst-case decoding capability of any code is $\frac{1-\kappa}{2}$. This is due to the fact that any code with message rate $\kappa := k/n$ has minimum distance at most $n - k + 1$ (i.e. two different codewords differ in at least $n - k + 1$ positions).

As we assume that the alphabet is large, Reed-Solomon (RS) Codes can be employed. RS-Codes are defined as follows: find an embedding of Σ into a finite field $\mathbb{F}$. Without loss of generality we will assume that the embedding is onto. Then a k-length word of Σ can be thought of as the coefficients of a polynomial $p \in \mathbb{F}[x]$ of degree less than k. The encoding of a k-length word equals to the set of pairs $\{(z_i, p(z_i))\}_{i=1}^n$ where $z_1, \ldots, z_n \in \mathbb{F}$ are some fixed distinct values.

Decoding can be achieved by the algorithm of Berlekamp and Welch [BW86], and it will allow error-correction provided that the number of intact symbols is $t \geq \frac{n+k}{2}$ (this corresponds to the maximum number of errors that can be corrected unambiguously in the worst-case). If $\kappa := k/n$ (the message rate) and $\epsilon := (n - t)/n$ (the error rate) it follows that we can correct error-rates of up to: $\epsilon \leq \frac{1-\kappa}{2}$. The above is optimal (at least in terms of unique decoding), in the sense that it allows the maximum number that can be corrected without any further assumptions on the noise (Reed-Solomon codes are Maximum Distance Separable (MDS) Codes).

In the NBSC model one can in fact see that unambiguous decoding can theoretically be extended much further than the worst-case error-rate $\frac{1-\kappa}{2}$. Intuitively this is

because of the size of the alphabet over which the NBSC is employed. The randomization of symbols that is performed by the NBSC suggests that the event that many different solutions fit the data is a rare probability event.

Proposition 71 *Let* $\{\langle x_i, y_i\rangle\}_{i=1}^{n}$ *be a RS-Code codeword of a random message* $p \in \mathbb{F}[x]$ *with* degree$(p) < k$ *that has* e *errors under the noise assumption suggested by the NBSC, s.t.* $e < n - k$. *Then the probability that it accepts another decoding* $p' \in \mathbb{F}[x]$ *with* $p \neq p'$ *is at most* $\binom{n}{t}/(|\mathbb{F}|^{t-k} - \binom{n}{t}^2)$ *(the probability is taken over all possible messages and noise corruptions)*

Proof. For some n, k and $t := n-e$ we denote by A_1 the number of strings from Σ^n that are partially corrupted RS-codewords with at most e errors. Furthermore we denote by A_2 the number of strings from Σ^n that are partially corrupted RS-codewords with at most e errors and accept more than one decoding.

First observe that $A_1 \leq \binom{n}{t}|\mathbb{F}|^{n-t+k}$. Equality does not hold since $\binom{n}{t}|\mathbb{F}|^{n-t+k}$ will count v times partially corrupted codewords that accept v solutions. Since the number of solutions of a partially corrupted codeword cannot exceed $\binom{n}{t}$ we conclude that $A_1 \geq \binom{n}{t}|\mathbb{F}|^{n-t+k} - \binom{n}{t}A_2$.

In order to approximate A_2 observe the following: let $p, p' \in \mathbb{F}[x]$ be the different ways to decode a partially corrupted RS-codeword with e errors. Suppose that they overlap in m points; clearly $m \in \{0, \ldots, k-1\}$. It follows that the total number of ways to select p, p' is $|\mathbb{F}|^{2k-m}$. For the remaining points the total number of ways to select them is $|\mathbb{F}|^{n-2t+m}$. It follows easily that $A_2 \leq \binom{n}{t}^2|\mathbb{F}|^{n-2t+m+2k-m} = \binom{n}{t}^2|\mathbb{F}|^{n-2t+2k}$

It is clear from the statement of the proposition that the probability that we would like to approximate equals A_2/A_1. Now observe that,

$$\frac{A_2}{A_1} \leq \frac{\binom{n}{t}^2|\mathbb{F}|^{n-2t+2k}}{\binom{n}{t}|\mathbb{F}|^{n-t+k} - \binom{n}{t}^3|\mathbb{F}|^{n-2t+2k}} = \frac{\binom{n}{t}}{|\mathbb{F}|^{t-k} - \binom{n}{t}^2}$$

this completes the proof. ■

Now observe that if the message rate is $\kappa := k/n$ and the error-rate is $\epsilon := e/n$, with $\kappa, \epsilon \in Q^+$ then it follows that the probability in proposition 71 is less than $\frac{2^n}{|\mathbb{F}|^{(1-\epsilon-\kappa)n}-4^n}$. As a result, provided that $\mathbb{F}$ satisfies $|\mathbb{F}|^{1-\epsilon-\kappa} > 4$ it follows that the probability of proposition 71 is "negligible."

So, the NBSC model will ensure unique solution with high probability, and therefore Reed-Solomon "List-Decoding" algorithms can also be employed for unambiguous decoding. By using the Guruswami-Sudan list-decoding algorithm of [GS98], we can obtain error-correction for error-rates:

$$\epsilon \leq 1 - \sqrt{\kappa}$$

It is easy to verify that as long as ϵ does not exceed $1 - \kappa$ a solution is preserved within the corrupted codeword. Due to proposition 71 it will be with high probability unique (provided that the underlying finite field is large enough) so the crucial question for NBSC decoding is whether we can devise efficient decoding algorithms for RS-Codes that can actually correct up to any error-rate strictly less than $1 - \kappa$. We deal with this issue in the sequel.

6.1.2 Interleaved codes

Interleaved codes are not an explicit family of codes, but rather an encoding mode that can be instantiated over any concrete family of codes. In this work we deal with Reed-Solomon Codes. The mode can be applied to any family of codes; in this section we shall give a code independent description.

Let Σ' be an alphabet with $|\Sigma'| = \sqrt[r]{|\Sigma|}$. Let $\phi : \Sigma \to (\Sigma')^r$ be a 1-1 mapping. We denote $\phi(x) = x^\phi[1]x^\phi[2]\ldots x^\phi[r]$, with $x^\phi[\ell] \in \Sigma'$, for $\ell = 1, \ldots, r$, for any $x \in \Sigma$.

Now let $enc : (\Sigma')^k \to (\Sigma')^n$ be an encoding function. An interleaved code w.r.t. ϕ of enc is a function $enc_\phi : (\Sigma)^k \to (\Sigma)^n$ that is defined as follows: Let $m_0 m_1 \ldots m_{k-1} \in$

$(\Sigma)^k$. First the following strings of $(\Sigma')^n$ are computed:

$$c_{1,1}\ldots c_{n,1} = enc(m_0^{\phi}[1]\ldots m_{k-1}^{\phi}[1])$$

$$\vdots$$

$$c_{1,r}\ldots c_{n,r} = enc(m_0^{\phi}[r]\ldots m_{k-1}^{\phi}[r])$$

An interleaved code is defined as follows:

$$enc_{\phi}(m_0 m_1\ldots m_{k-1}) = \phi^{-1}(c_{1,1}\ldots c_{1,r})\ldots\phi^{-1}(c_{n,1}\ldots c_{n,r})$$

A graphical representation of code interleaving is presented in figure 6.3.

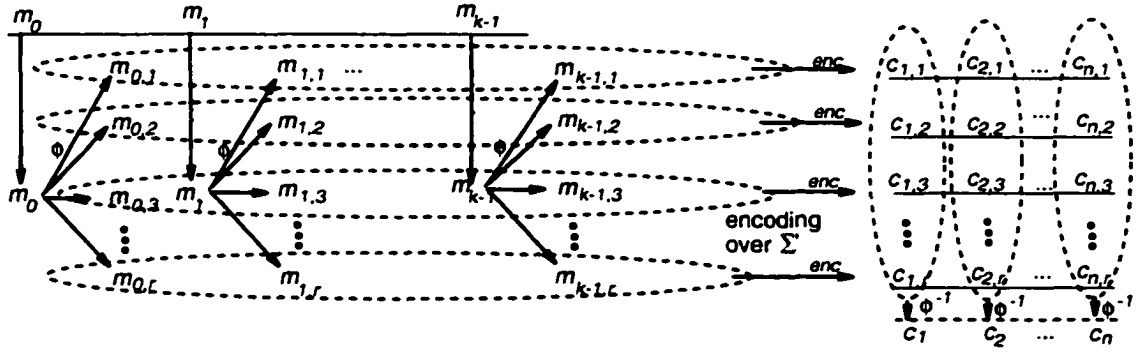

Figure 6.3: Encoding schema for an interleaved code. Single subscript symbols (m_i, c_i) belong to the "outer" alphabet Σ: double subscript symbols $(m_{i,j}, c_{i,j})$ belong to the "inner" alphabet Σ'.

The common way to use an interleaved code, is simply decode each of the code words $(c_{1,i}\ldots c_{n,i})$ separately. Such a decoding does not increase the error correction rate. Rather burst errors are distributed over several code words.

The approach we take uses code interleaving as the basic encoding mode. In contrast to the standard approach of decoding each one of the codewords individually, we will present a decoding technique that attempts to correct all codewords simultaneously so that in combination with the NBSC model our decoding algorithm can

withstand higher error-rates. We concentrate on Reed-Solomon Codes as described in section 6.1.3.

An extended version of the above schema, known as cross-interleaving is to replace the 1-1 mapping ϕ^{-1} with a second error correcting code. Cross-interleaving increases the code size, but allows to correct a larger class of errors. By testing the second (outer) code potential error locations can be found. Then decoding the inner code can be done by treating those potential error locations as erasures. This increases the error correcting capabilities of the scheme, because generally a code allows more erasures than errors. The present work focuses on simple interleaving (without an outer code).

6.1.3 Interleaved Reed-Solomon Codes

Let $\Sigma = GF(2^B)$ be the alphabet for the encoding function (without loss of generality we will focus only on binary extension fields — all our results hold also for general finite fields). The parameters are $n, k \in \mathbb{N}$ where $\kappa := k/n$ is the message rate. We assume additionally a parameter $r \in \mathbb{N}$ with the property $br = B$ (we remark here that a similar scheme is also possible when B is prime, however, for notational simplicity we do not deal with this case in this abstract). Let $z_1, \ldots, z_n \in GF(2^b)$ be fixed distinct constants.

We now describe our approach for the case of interleaved Reed-Solomon Codes. First, observe that there exists a straightforward bijection mapping $\phi : GF(2^B) \rightarrow (GF(2^b)^r$. Given $m_0 \ldots m_{k-1} \in GF(2^B)$ we define the following polynomials over $GF(2^b)$, for $\ell = 1, \ldots, r$:

$$p_\ell(x) := m_0^\phi[\ell] + m_1^\phi[\ell]x + \ldots + m^\phi[\ell]_{k-1}x^{k-1}$$

The encoding of $m_0 \ldots m_{k-1}$ is set to be the string

$$\phi^{-1}(p_1(z_1) \ldots p_r(z_1)) \ldots \phi^{-1}(p_1(z_n) \ldots p_r(z_n))$$

The common way to decode RS-interleaved-codes is to concentrate to each of the r coordinates individually and employ the decoding algorithm of the underlying RS-Code over Σ'. This can be done as follows: given a (partially corrupted) codeword $c_1 \dots c_n \in (\Sigma)^n$ we treat the string $c_1^o[1] \dots c_n^o[1] \in (\Sigma')^n$ as a partially corrupted RS-codeword over Σ' and we employ the RS-Decoding of Berlekamp-Welch to recover p_1. Observe that the recovery of p_1 will imply the recovery of $p_2, \dots, p_r$ immediately, provided that the error-rate is at most $\frac{1-\kappa}{2}$ (due to the employment of the NBSC model in the transmission of the $GF(2^B)$ strings it is easy to verify that all codewords $c_1^o[\ell] \dots c_n^o[\ell]$, $\ell = 1, \dots, r$ have identical error-pattern).

Moreover, due to the properties of the NBSC model one can further employ the Guruswami-Sudan list-decoding algorithm that will produce a unique solution with high probability for error-rates up to $1 - \sqrt{\kappa}$. The main focus of the next section is to go beyond this bound, in the NBSC model.

6.1.4 Our decoding algorithm

Suppose we want to correct an error-rate ϵ for a message-rate κ with $\epsilon < 1 - \kappa$ over the finite field $GF(2^B)$. We select $r = \lceil \frac{1-\tau}{\tau-\kappa} \rceil$, (where $\tau \in Q^+$ is a parameter that belongs in $(\kappa, 1-\epsilon)$) and we perform interleaved encoding over the finite field $GF(2^b)$ with $br = B$.

In this section we present a generalization of the Berlekamp-Welch algorithm for polynomial reconstruction [BW86] for interleaved Reed-Solomon codes. Let $c_1 \dots c_n \in (GF(2^B))^n$ be the received codeword. Let $y_{i,1} \dots y_{i,r} = \phi(c_i)$, with $y_{i,\ell} \in GF(2^b)$ for all $i = 1, \dots, n, \ell = 1, \dots, r$. With this interpretation, and in the NBSC model, we can reformulate the decoding problem as follows:

The Decoding Problem. The problem we are dealing with in this section is the following: given $\{\langle z_i, y_{i,1}, \dots, y_{i,r} \rangle\}_{i=1}^n$ s.t. there exists a set $I \subseteq \{1, \dots, n\}$ (the set of indices that were received intact) with $|I| := T := n - E$ where E is the random

variable that corresponds to the number of errors (E follows a binomial probability distribution with ϵ probability of success over n trials). Additionally for all $i \in I$, $y_{i,\ell} = p_\ell(z_i)$ for some polynomials $p_1, \ldots, p_r$ of degree less than k. According to the NBSC model, it holds that all $y_{i,\ell}$ with $i \notin I$ are chosen uniformly at random from $GF(2^b)$. Under these conditions the goal is to "reconstruct" the polynomials $p_1, \ldots, p_r$.

Our Decoding Algorithm. We set $t = \tau n$. Let $\tilde{E}(x) = \prod_{i \in I}(x - z_i)$: $\tilde{E}$ is monic with degree $n - T$. If $\tilde{M}_\ell(x) := p_\ell(x)\tilde{E}(x)$ it holds that $\tilde{M}_\ell(z_i) = p_\ell(z_i)\tilde{E}(z_i) = y_{i,\ell}\tilde{E}(z_i)$, for all $i = 1, \ldots, n$. The degree of $\tilde{M}_\ell$ is less than $n - T + k$.

Condition. The constant τ is selected so that $T \geq t = \tau n$ (this will be ensured using the Chernoff bound — see below). It follows that the degree of $\tilde{E}$ is at most $n - t$ and that the degree of each $\tilde{M}_\ell$ is less than $n - t + k$. Since T is a binomial random variable with success probability $1 - \epsilon$ over n experiments, it holds that (using the Chernoff bound):

$$\textbf{Prob}[T \leq \tau n] = \textbf{Prob}[T \leq (1 - \alpha)n(1 - \epsilon)] \leq e^{-\alpha^2(1-\epsilon)n/2} = e^{-\frac{(1-\tau-\epsilon)^2}{2(1-\epsilon)}n}$$

this assumes that $\tau < 1 - \epsilon$, and that $\alpha = 1 - \frac{\tau}{1-\epsilon}$. It follows that we can assume that $T > t$ is true with probability $1 - e^{-\frac{(1-\tau-\epsilon)^2}{2(1-\epsilon)}n}$.

Consider the following system of rn equations:

$$[M_1(z_i) = y_{i,1}E(z_i)]_{i=1}^n \ldots [M_r(z_i) = y_{i,r}E(z_i)]_{i=1}^n \qquad (*)$$

where the unknowns are the coefficients of the polynomials $M_1, \ldots, M_r, E$ (a total number of $r(n - t + k) + n - t$ unknowns).

First observe that due to the choice of $r = \lceil \frac{1-\tau}{\tau-\kappa} \rceil$ the system $(*)$ is not underspecified: the number of equations rn is larger than the number of unknowns: $r(n - t + k) + n - t$ (recall that $t = \tau n$).

The system $(*)$ has at least one solution since we know that $\tilde{M}_1, \ldots, \tilde{M}_r, \tilde{E}$ (constructed above) satisfy the equations. Next we will show that with high probability the system accepts only a unique solution, that can be recovered by solving an appropriate sub-system of $(*)$.

Theorem 72 *The system of equations $(*)$ defined above has a unique solution with probability at least $1 - n\frac{1-r}{2^b}$, and an algorithm for recovering this solution.*

Proof. Consider the following matrices, for $\ell = 1, \ldots, r$:

$$M = \begin{pmatrix} 1 & z_1 & z_1^2 & \ldots & z_1^{n-t+k-1} \\ 1 & z_2 & z_2^2 & \ldots & z_2^{n-t+k-1} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 1 & z_n & z_n^2 & \ldots & z_n^{n-t+k-1} \end{pmatrix}$$

$$M_\ell = \begin{pmatrix} y_{1,\ell} & y_{1,\ell} z_1 & y_{1,\ell} z_1^2 & \ldots & y_{1,\ell} z_1^{n-t-1} \\ y_{2,\ell} & y_{2,\ell} z_2 & y_{2,\ell} z_2^2 & \ldots & y_{2,\ell} z_2^{n-t-1} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ y_{n,\ell} & y_{n,\ell} z_n & y_{n,\ell} z_n^2 & \ldots & y_{n,\ell} z_n^{n-t-1} \end{pmatrix}$$

The matrix of the system $(*)$ is the following (where $\mathbf{0}$ stands for a $n \times (n-t+k)$-matrix with 0's everywhere):

$$A = \begin{pmatrix} M & \mathbf{0} & \ldots & \mathbf{0} & -M_1 \\ \mathbf{0} & M & \ldots & \mathbf{0} & -M_2 \\ \vdots & \vdots & \ldots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & M & -M_r \end{pmatrix}$$

We index each row of A by the pair $\langle i, \ell \rangle$ with $i \in \{1, \ldots, n\}$ and $\ell \in \{1, \ldots, r\}$. The ℓ-th block row of A contains the rows $\langle 1, \ell \rangle, \ldots, \langle n, \ell \rangle$.

We need to show how to select a minor $\hat{M}$ of size $n - t$ from the $n - t$ rightmost columns of A, so that the probability that it is singular is small, with the property that it contains up to $t - k$ rows from each block-row of A (this is possible since $r(t-k) \geq n - t$). Once we have such minor we will form a matrix $\hat{A}$ that contains

$n-t+k$ rows from each block-row of A and the rows contained in $\hat{M}$. Clearly $\hat{A}$ will be a square matrix with $r(n-t+k)+n-t$ rows and it will be non-singular. We will use $\hat{A}$ to solve the linear system $(*)$.

To complete the proof we show how to select the minor $\hat{M}$. We refer to this method of constructing $\hat{M}$ as "minor matrix decomposition." We form $\hat{M}$ by selecting the rows indexed from 1 through $t-k$ from the first block-row of A, the rows indexed from $t-k+1$ through $2(t-k)$ from the second block-row of A, and so on until we select $n-t$ rows. As a result $\hat{M}$ will be of the form:

$$\hat{M} = \begin{pmatrix} y_{1,1} & y_{1,1}z_1 & y_{1,1}z_1^2 & \cdots & y_{1,1}z_1^{n-t-1} \\ y_{2,1} & y_{2,1}z_2 & y_{2,1}z_2^2 & \cdots & y_{2,1}z_2^{n-t-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ y_{t-k,1} & y_{t-k,1}z_{t-k} & y_{t-k,1}z_{t-k}^2 & \cdots & y_{t-k,1}z_{t-k}^{n-t-1} \\ y_{t-k+1,2} & y_{t-k+1,2}z_{t-k+1} & y_{t-k+1,1}z_{t-k+1}^2 & \cdots & y_{t-k+1,2}z_{t-k+1}^{n-t-1} \\ y_{t-k+2,2} & y_{t-k+2,2}z_{t-k+2} & y_{t-k+2,1}z_{t-k+2}^2 & \cdots & y_{t-k+2,2}z_{t-k+2}^{n-t-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ y_{t-k+t-k,2} & y_{t-k+t-k,2}z_{t-k+t-k} & y_{t-k+t-k,2}z_{t-k+t-k}^2 & \cdots & y_{t-k+t-k,2}z_{t-k+t-k}^{n-t-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ y_{n-t,\ell} & y_{n-t,\ell}z_{n-t} & y_{n-t,\ell}z_{n-t}^2 & \cdots & y_{n-t,\ell}z_{n-t}^{n-t-1} \end{pmatrix}$$

Let $J := \{1,\ldots,n-t\} \cap I$. It follows that for all $i \notin J$, $y_{i,\ell}$ will be randomly distributed over $GF(2^B)$. Additionally if $|J| > k$ the variables $y_{i,\ell}$ for $i \in J$ are k-wise independent. For a fixed J, $\det(\hat{M})$ can be seen as a multi-variate polynomial with $y_{i,\ell}$ as variables. The total degree of $\det(\hat{M})$ is at most $n-t$. Additionally $\det(\hat{M})$ is not the 0-polynomial since the assignment of all $y_{i,\ell}$ to 1 (which is consistent with the k-wise independency), makes $\det(\hat{M})$ a Vandermonde determinant that is non-zero. It follows by Schwartz's lemma, [Sch80], that $\det(\hat{M})$ cannot have more than $\frac{n-t}{2^b}$ roots and as a result $\det(\hat{M})$ will be 0 with probability at most $\frac{n-t}{2^b}$. This fact is independent from the overlap of I to $\{1,\ldots,n-t\}$ (i.e. for all possible overlaps of the two sets, the probability will be bounded as above — the result follows). ■

The above theorem ensures the uniqueness of the linear system's $(*)$ solution

$\langle M_1^*(x), \ldots, M^*(x), E^*(x)\rangle$. Whereas in the case of the Berlekamp and Welch algorithm it can be shown using the condition $t \geq \frac{n+k}{2}$, in our case such relation does not hold necessarily (and in fact we are interested precisely in the cases where it does not hold, since otherwise one can apply the Berlekamp-Welch algorithm to perform interleaved RS decoding directly). In our case uniqueness is shown in the probabilistic sense.

As a result let $M_1^*, \ldots, M_r^*, E^*$ be a solution of the system $(*)$ computable by standard linear algebra using the sub-system of $(*)$ suggested in the proof of theorem 72. The output of the algorithm is the tuple of polynomials

$$\langle M_1^*(x)/E^*(x), \ldots, M^*(x)/E^*(x)\rangle$$

The following theorem is an immediate result of the above exposition:

Theorem 73 *Suppose that κ is the message rate and ϵ is the error-rate, with $\kappa, \epsilon \in Q^+$. Then, our decoding algorithm succeeds with probability at least $(1 - n\frac{1-\tau}{2^b})(1 - e^{-\frac{(1-\tau-\epsilon)^2}{2(1-\epsilon)}n})$. where τ is a constant of Q^+ that satisfies $\kappa < \tau < 1 - \epsilon$.*

Examples. As an example consider a message-rate of 1/4, and an error-rate of 5/8 (which is non-correctable by the known decoding algorithms, e.g. the [GS98]-method would work for error-rates of up to 1/2 — given that the message-rate is 1/4). Now if we set $\tau = 5/16$ and we use the interleaved schema for $r = 11$ with alphabets $\Sigma = GF(2^B) = GF(2^{440})$ and $\Sigma' = GF(2^b) = GF(2^{40})$, it follows that the probability of success is $(1 - \frac{11n}{2^{54}})(1 - e^{-\frac{n}{192}})$ (in the NBSC model). For block-length of $n = 4992$, it follows that the success probability equals approximately $(1 - 2^{-38})^2$.

Practical Example: We remark that when the NBSC channel model is extended to assure an upper bound on the number of errors, we can eliminate the need of the Chernoff bound analysis part. This implies a simplified probabilistic analysis and assures that the method applies to smaller codewords (note that such bounds on the

number of errors are typical in the analysis of many decoding algorithms). Then, using the same message-rate and error-rate as above, we can apply a block-length of $n = 64$ with a bound on number of errors which equals to 40 (that is non-correctable by the known decoding algorithms, e.g. the [GS98]-method works up to 32 errors). The decoding algorithm for interleaved RS-codes with alphabet $\Sigma = GF(2^{440})$ and $\Sigma' = GF(2^{40})$ will be successful with probability at least $1 - 2^{-44}$.

6.2 Hardness vs. Self-Reducibility in Decoding Problems

Random self-reducible problems have important applications in Computational Complexity, Cryptography and Program Testing and Checking. For example, an early application was by Blum and Micali [BM84] who employed it in their work on pseudorandom generation, while further applications include program self-correcting [BLR90]. Random self-reducibility of concrete problems, e.g. the Permanent, [Lip91], or of lattice related problems, [Ajt96], has led to significant advances in computational complexity. Many structural properties of randomly self-reducible problems have been investigated as well, for example lower bounds on such problems were investigated in [FKN90]. Continuing this line of research and considering extended formulations such as multiple query self-reducibility and adaptive and non-adaptive queries. Feigenbaum and Fortnow [FF93] showed that complete problems in the polynomial hierarchy are most likely non-randomly self-reducible. Later, [FFLS94], Feigenbaum et al. showed that adaptiveness of the queries yields, a strictly more powerful formulation of random self-reducibility.

Many of the natural computational problems (e.g., the discrete logarithm function and the modular square-root relation) exhibit a property called "partial random self-reducibility" a notion which was formally introduced and investigated in [FKN90].

Partial random self-reducibility (which is a generalization of random self-reducibility) is more often exhibited by natural computational problems. It is defined over what is called "orbits" which are sub-domains obtained by an "orbit-division" of the problem's instance space (e.g. when fixing the prime and the generator in the discrete logarithm problem or when fixing a modulus in the modular square-root problem).

In this section, we will investigate tradeoffs between partial random self-reducibility and computational hardness of a set of problems in the area of decoding linear error-correcting codes. Establishing mutual exclusive conditions within the realm of these problems is important. The fact that random-self reducibility and hardness can be mutually exclusive may be an interesting direction in understanding the power of randomization, the limitation on sampling or as a way to look for avenues by which to establish efficient (randomized) algorithms for these problems or problems of a similar structure.

We shall start with a general treatment of orbit divisions in order to obtain preliminary insight. We note, in fact, that (supposedly) intractable problems (e.g. problems related to satisfiability) possess orbit-divisions for which they *cannot be* randomly self-reducible unless some complexity assumptions is violated. On the other hand, it is easy to see that any polynomial-time computable function is trivially random self-reducible for any orbit-division. We note that orbit-divisions for a function f can either be "input-defined" (i.e. the orbit into which an element x belongs to depends on some easily computable property of x) or "output-defined" (i.e. the orbit into which an element x belongs to depends also on some easily computable property of $f(x)$).

Since our goal is to deal with problems that are assumed to be hard and trade hardness against random self-reduction we introduce the notion of a *hardcore* orbit-division (where given x, if the orbit that x belongs to can be identified, then $f(x)$ is easily computable). Hardcore orbit-divisions possess a natural associated measure

which we call the *saturation factor*: the number of random samples from an orbit that we need in order to determine the orbit index (name) efficiently, and thus solve the problem.

Functions that are partially random self-reducible over a hardcore orbit-division have been widely used in Cryptography where we observe that (under the underlying complexity assumption) it should hold that the saturation factor is in fact at least super-polynomial in the input size. (As an example consider trying to solve the quadratic residuosity problem: Indeed, given an input we can sample polynomially many random elements with the same residuosity in polynomial time, yet under the quadratic residuosity assumption, solving it should be hard). Complementing the above, we show that a *polynomial saturation factor* for a hardcore orbit-division of a presumably hard function *implies lack of partial random-self-reducibility.*

With the above relations established, we proceed with our central investigation, that of "Linear Codes Decoding" problem and specific instantiations thereof (namely, Reed-Solomon decoding). Problems of random linear code decoding and Reed-Solomon decoding have been suggested for usage as a hard cryptographic problem [McE78, NP99, KY01c, KY02]. The problem of General Linear Codes decoding has been studied extensively and has been shown to be NP-Hard for various challenges, see e.g. [Bar98]. The Decoding Problem of Reed-Solomon Codes (aka Polynomial Reconstruction — PR) has also been studied extensively, and the problem is not known to be solvable in better than exponential time if the error-rate ϵ becomes larger than $1 - \sqrt{\kappa}$ where κ is the message-rate, [Sud97, GS98]. Previous work considered the problem in the worst-case. Some challenges related to a Polynomial Reconstruction instance have been shown NP-Hard, [GRS95].

Our investigation shows that these problems possess hardcore orbit-divisions for which the following properties are mutually exclusive: Either the problems are NOT partially random-self reducible or they are polynomial-time solvable with very high

probability. (Note that our challenges are over a large finite field, and have instance parameters for which currently there do not exist efficient decoding algorithms). Our arguments follow a probabilistic algorithm over the sample, which our analysis shows to solve most instances in the space of challenges.

Regarding the saturation factor (number of needed samples), we show that in the case of RS-codes the saturation factor of the hardcore orbit-division we consider is constant, whereas in the case of general (e.g. Random) Linear Codes the upper bound on the saturation factor is linear.

Notation. Denote by $(n)_k := n(n-1)\ldots(n-k+1)$, and if A is a set denote by $(A)_k$ the set of all k-tuples over A without repetitions. Let $Pow_t(A)$ denote the set of all subsets of A of size t.

Partial Random Self-Reductions

The notion we use, partial random self-reducibility with orbits, was first introduced by Feigenbaum, Kannan, and Nisan [FKN90].

Let D, R be sets of strings. Denote by $[D]_n$ the subset of D that contains all strings of size n. Given two sets with $A \subseteq B$ we say that $[A]_n$ is negligible w.r.t. $[B]_n$, if $\#[A]_n/\#[B]_n$ is a negligible function in n. A function $f : D \to R$ is called length-preserving if $\exists p \in \mathbb{Z}[x]$ s.t. $x \in [D]_n$ implies that $f(x) \in [R]_{n'}$ where $n' \leq p(n)$.

Definition 74 *Let $f : D \to R$ be a length preserving function. An* orbit-division *of the instance space D is a collection of sets $\{[D^i]_n\}_{i \in A_n; n \in \mathbb{N}}$ so that: (i) $\cup_{i \in A_n}[D^i]_n = [D]_n$ for all n: (ii) the set $[D^i]_n \cap [D^j]_n$ for $i \neq j$ is negligible (w.r.t. $[D]_n$). (iii) given $i \in A$ and 1^n the set $[D^i]_n$ is polynomial-time samplable. (iv) given $x, f(x)$ then an index $i \in A_n$ s.t. $x \in [D^i]_n$ can be computed in polynomial-time.*

Given a $x \in [D]_n$ suppose that $i \in A_n$ is such that $x \in [D^i]_n$. We define $\mathcal{O}(x) = [D^i]_n$ to be the *orbit* of x. Note that for some elements no unique orbit is defined (but

due to property (ii) above the number of these elements of is very small).

Definition 75 *A length preserving function $f : D \to R$ is partial random self-reducible (p-rsr) w.r.t. the orbit-division $\{[D^i]_n\}_{i \in A_n; n \in \mathbb{N}}$ if there exist two poly-time computable functions ϕ, σ, so that*

i. $f(x) = \phi(x, r, f(\sigma(r, x)))$ *for all* $x \in D, r \in \{0,1\}^{q(|x|)}$.

ii. if r is uniformly distributed over $\{0,1\}^{q(|x|)}$ then $\sigma(r, x)$ is statistically indistinguishable from the uniform distribution over $\mathcal{O}(x)$.

Note that we make no assumptions on how σ operates if the given instance x belongs to the intersection of two or more orbits. Since the intersection of two orbits is a negligible set w.r.t. the instance space for a certain length n, such an event is rare.

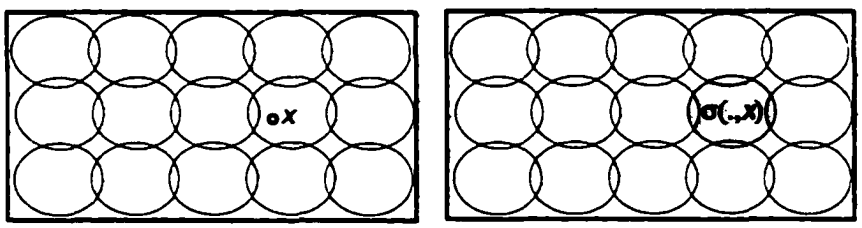

Figure 6.4: Orbit Division of the instance space, and the randomization of an instance to its orbit, in a partial random self-reduction.

Examples. (i) Consider the discrete-logarithm function $f(g^x \bmod p, g, p) = x$, where $p = 2q + 1$, g is an element of $\mathbb{Z}_p^*$ of degree q. It follows easily that f is p-rsr w.r.t. the orbit-division $\{\{\langle g^x \bmod p, g, p\rangle \mid g, x\}\}_{p \in A}$ where A is the set of all n-bit primes with the property $p = 2q + 1$ for some other prime q: indeed it is easy to verify that if $\sigma(r|r', \langle g^x \bmod p, g, p\rangle) := \langle g^{rx} \bmod p, g^{r'}, p\rangle$ and $\phi(\langle v, g, p\rangle, r|r', x') := (x'r')r^{-1} (\mathrm{mod} q)$, then ϕ, σ show that f is p-rsr.

(ii) Consider the square root function f over the set of residues modulo a composite N, with $f(a \bmod N, N)$ equal to the set of all square roots of $a \bmod N$ in $\mathbb{Z}_N^*$. The function is a p-rsr with each orbit having a fixed N: the randomization for $a \bmod N$ is by selecting a random $r \in \mathbb{Z}_N^*$ and computing $r^2 \cdot a (\bmod N)$. If t is a square root of $r^2 a \bmod N$ then it holds that $r^{-1}t$ is a square root of the given input $a \bmod N$.

The following proposition asserts that polynomial-time computable functions accept partial random self-reductions in a trivial manner (this is merely due to the power of the reduction).

Proposition 76 *Any polynomial time computable function f is p-rsr w.r.t. any orbit-division.*

P-RSRs and Satisfiability

To illustrate further the perspective of the present work we present some results on hardness — self-reducibility tradeoffs in satisfiability related problems. These results extend the work of [FKN90] who presented some similar results (but for the stronger notion of "1-sided" partial random self-reductions, where the randomizer function σ is assumed to be "membership preserving").

For a certain set S denote by χ_S its characteristic function. In this section we denote by $\chi_{\overline{\text{SAT}}}$ the function that given an unsatisfiable formula returns "1", otherwise (if the input formula is satisfiable) it returns "0". Similarly we denote χ_{QBF} the function that given a quantified boolean formula it returns "1" if and only if it is true.

Theorem 77 *There exist orbit-divisions for which if $\chi_{\overline{\text{SAT}}}$ is p-rsr then* NP = coNP.

Proof. Suppose that the orbit-division of the instance space is such that if $X \in [D^i]_n$ then X is of length n and i is the first half of X. Suppose that $\chi_{\overline{\text{SAT}}}$ is p-rsr w.r.t. this orbit division. Consider the following NP-machine:

Given X:

1. Construct $Y = X_{\frac{1}{2}} \wedge u \wedge \overline{u} \wedge (\text{padding})$, where u is a literal that does not appear in the first half of X and padding denotes a formula of appropriate length (so that Y is of length n).

2. Guess r.

3. Compute $\sigma(r, X)$.

4. If $\sigma(r, X) = Y$ then output $\phi(X, r, 1)$ else output 0.

Claim. The NP-machine above will accept (i.e. output 1) if and only if $X \in \overline{\text{SAT}}$.

First, suppose that $X \in \overline{\text{SAT}}$. It holds that Y as constructed in the NP-machine it belongs in the same orbit as X. By the definition of the p-rsr ϕ, σ there exists at least one r s.t. $\sigma(r, X) = Y$. Such an r will be guessed in step 2 of the NP-machine. We know by the construction of Y that $\chi_{\overline{\text{SAT}}}(Y) = 1$. Second, since by the definition of ϕ it holds that $\phi(X, r, 1) = 1$. As a result the NP-machine above will output 1, for at least one of its guesses for r.

Now suppose that $X \notin \overline{\text{SAT}}$. Then the NP-machine above should output "0" independently of guessed r. If r is such that $\sigma(r, X) \neq Y$ then this is immediate from step 4 of the NP-machine. On the other hand, if $\sigma(r, X) = Y$, we know by the construction of Y that $\chi_{\overline{\text{SAT}}}(Y) = 1$ and by the properties of ϕ it holds that $\phi(X, r, 1) = 0$. As a result the NP-machine will output 0, also for any guesses r for which it holds that $\sigma(r, X) = Y$. ■

It is possible to extend the techniques employed in the above theorem to quantified boolean formulas to obtain the following result:

Theorem 78 *There exists orbit-divisions for which if* χ_{QBF} *is p-rsr it holds that* NP = PSPACE.

6.2.1 Hardcore Orbit Divisions and Saturation

In this section we introduce a central concept in our methodology, the notion of a hardcore orbit-division: informally, in a hardcore orbit-division, if we know the orbit into which a certain instance x belongs to, then we can compute efficiently the evaluation of the function f over x.

Definition 79 *An orbit-division $\{[D^i]_n\}_{i\in A_n; n\in\mathbb{N}}$ of D is hardcore for the function $f : D \to R$ if there is a computable function ψ so that $f(x) = \psi(i, x)$ for all $i \in A, n \in \mathbb{N}$ and $x \in [D^i]_n$.*

Hardcore orbit-divisions arise naturally in many settings.

A "direct" hardcore orbit-division for a function $f : D \to R$ can be immediately obtained by grouping instances according to their evaluation $f(x)$: $\{x \in D \mid f(x) = y\}_{y\in R}$. As an example consider the quadratic residuosity predicate function $\chi_{QR} : \mathbb{J}_N \to \{-1, 1\}$ where $\mathbb{J}_N$ denotes the set of Jacobi symbol $+1$ elements of $\mathbb{Z}_N^*$: the function is defined as follows: $\chi_{QR}(x) = 1$ iff x is a quadratic-residue modulo N. Consider the orbit-division A_{-1}, A_1 of $\mathbb{J}_N$ where A_1 contains exactly those elements of $\mathbb{J}_N$ that are quadratic-residues modulo N. It is easy to see that $\{A_i\}_{i\in\{-1,1\}}$ constitutes an orbit-division. Additionally, it is immediate that the orbit-division $\{A_i\}_{i\in\{-1,1\}}$ is hardcore for χ_{QR}.

Presumably hard functions that are partially random-self reducible w.r.t. a hardcore orbit-division (such as the χ_{QR} function) have many applications in Cryptography.

An important measure for a (hardcore) orbit-division is the *saturation factor* that we introduce next:

Definition 80 *The saturation factor $\rho(n)$ of a (hardcore) orbit-division $\{[D^i]_n\}_{i\in A}$ is an upper bound (quantified over all orbits) of the number of random samples that*

we need from some orbit $[D^i]_n$ *in order to compute* i *efficiently. Formally,* $\rho(n)$ *has the property that there exists a* PPT $\mathcal{A}$ *that runs in time polynomial in* $n, \rho(n)$ *s.t. for any* $i \in A$*, the probability* $\mathbf{Prob}[\mathcal{A}(x_1, \ldots, x_{\rho(n)}) = i] = 1 - \epsilon(n, \rho(n))$ *where* ϵ *is a negligible function and* $x_1, \ldots, x_{\rho(n)}$ *are uniformly distributed over* $[D^i]_n$.

Next we reveal the connection between the saturation factor of a hardcore orbit-division and partial random self-reducibility of a function.

Theorem 81 *If for a certain function* f *and an orbit-division that is hardcore for* f *it holds that the saturation factor is polynomial in the input size* n*, then, exactly one of the following is true:*
(i) either f *is computable in probabilistic polynomial time in* n *(for most instances in its domain), or*
(ii) f *is NOT partially random self-reducible w.r.t. the orbit-division.*

Proof. Suppose that $f : D \to R$ is partially random self-reducible w.r.t. the hardcore orbit division $\{[D^i]_n\}_{i \in A_n; n \in \mathbf{N}}$. Let ϕ, σ the two computable functions for the partial random self-reduction of f and $\rho(n)$ be the saturation factor of the orbit-division which is polynomial in n. Given $x \in [D]_n$ with very high probability it belongs to a unique orbit $\mathcal{O}(x)$ (property (ii) of definition 74). For $r_1, \ldots, r_{\rho(n)}$ random bitstrings we compute the instances $x_1 = \sigma(r_1, x), \ldots, x_{\rho(n)} = \sigma(r_{\rho(n)}, x)$. By the properties of σ it holds that $x_1, \ldots, x_{\rho(n)}$ are uniformly distributed in $\mathcal{O}(x)$. Since $\rho(n)$ is the saturation factor of the orbit-division, it holds that given $\rho(n)$ samples from $\mathcal{O}(x)$ we can compute $i \in A_n$ s.t. $\mathcal{O}(x) = [D^i]_n$. It follows that because the orbit-division is hardcore, given i, x we compute in probabilistic polynomial-time $f(x)$. ■

Note that if the saturation factor for a hardcore orbit-division of a function is 1 then it is immediate that the function is computable in polynomial-time. On the other

hand for presumably hard functions (such as χ_{QR} above) that are partially random-self reducible w.r.t. a hardcore orbit-division, it holds that the saturation factor should be superpolynomial in n (otherwise using the partial random self-reducibility over the hardcore orbit-division one would be capable of computing the function in polynomial time):

Fact 1. Under the Quadratic Residuosity Assumption [GM84], the direct hardcore orbit-division for χ_{QR} defined above has superpolynomial saturation factor.

The above remarks suggest a natural connection between the size of the saturation factor for hardcore orbit-divisions and the hardness and self-reducibility of a certain function.

6.2.2 Hardness vs. Self-reducibility in Decoding Linear Codes

A linear code is defined by a generator matrix $A \in \mathbb{F}^{n \times k}$, with $n > k$. The codeword of a message $\mathbf{m} \in \mathbb{F}^k$ is defined as $\mathbf{c} = A \cdot \mathbf{m}$. In our setting, we consider the decoding problem over a *large* finite field where $\log |\mathbb{F}|$ is proportional to n. The decoding problem we consider is defined as follows:

Definition 82 *(*LC-DECODING*) For a fixed linear code A, given a $\mathbf{c} \in \mathbb{F}^n$ so that there exists a $\mathbf{m} \in \mathbb{F}^k$ so that $A \cdot \mathbf{m}$ agrees with $\mathbf{c}$ in at least t positions, recover all $\langle \mathbf{m}, I \rangle$, s.t. $(A \cdot \mathbf{m})_i = (\mathbf{c})_i$ for all $i \in I$ and $|I| \geq t$.*

We denote the set of all instances $\mathbf{c}$ with parameters n, k, t by $\mathcal{G}^A_{n,k,t}$ (where it assumed that $n > t > k$). The function that corresponds to LC-DECODING is defined as follows: $F_{dec} : \mathcal{G}^A_{n,k,t} \to \mathbb{F}^k$ maps instances $\mathbf{c}$ to one of their possible solutions $\langle \mathbf{m}, I \rangle$. Note that F_{dec} is well defined only if there is a unique solution.

We note that the above definition of Decoding is essentially a "promise" version of the typical list-decoding problem where we are given the assurance that the number

of errors is bounded above by $n - t$. Typically, we will consider constant "message-rate" and "error-rate", i.e. $k := \kappa n$ and $t := \tau n$ for some $\kappa, \tau \in \mathbb{Q}^+$ where κ denotes the message-rate and $1 - \tau$ denotes the error-rate (note: $\kappa < \tau$).

Structure of the Instance Space

Let $\mathcal{G}^A_{n,k,t}(I)$ denote the subset of $\mathcal{G}^A_{n,k,t}$ so that every instance $\mathbf{c}$ has a solution of the form $\langle \mathbf{m}, I \rangle$. It is clear that $\mathcal{G}^A_{n,k,t} = \cup_{|I|=t} \mathcal{G}^A_{n,k,t}(I)$.

We consider linear codes that satisfy the following property: any k-minor of the matrix A is non-singular. This property is typically satisfied in many concrete examples of linear codes (e.g. Reed-Solomon codes) and moreover it is satisfied by a random linear code with very high probability when the underlying finite field is large (in particular for a certain random linear code A it is satisfied with probability at least $1 - \frac{1}{|\mathbb{F}|}$).

Lemma 83 *Suppose that* $\log |\mathbb{F}| > 2n$. *Then it holds.*

(i) The number of elements of $\mathcal{G}^A_{n,k,t}$ *can be approximated (within negligible error) by* $\binom{n}{t} |\mathbb{F}|^{n-k+t}$.

(ii) The ratio of the number of instances in $\mathcal{G}^A_{n,k,t}$ *with more than one solution, over* $\#\mathcal{G}^A_{n,k,t}$ *is less than* 2^{-n}.

Proof. (i) For any $I \subseteq \{1, \ldots, n\}$ with $|I| = t$ it holds that $\#\mathcal{G}^A_{n,k,t}(I) = |\mathbb{F}|^{n-t+k}$. This is straightforward since $n - t + k$ are exactly the degrees of freedom that each element of $\mathcal{G}^A_{n,k,t}(I)$ has.

Clearly if an instance $\mathbf{c} \in \mathcal{G}^A_{n,k,t}$ has two distinct solutions $\langle \mathbf{m}_1, I_1 \rangle$ and $\langle \mathbf{m}_2, I_2 \rangle$, it holds that $\mathbf{c} \in \mathcal{G}^A_{n,k,t}(I_1) \cap \mathcal{G}^A_{n,k,t}(I_2)$. To determine the likelihood that a given PR-instance has a single solution or more, the following observation is helpful:
(a) For all $I_1, I_2 \subseteq \{1, \ldots, n\}$, with $|I_1| = |I_2| = t$, $I_1 \neq I_2$, it holds that $\#(\mathcal{G}^A_{n,k,t}(I_1) \cap \mathcal{G}^A_{n,k,t}(I_2)) \leq |\mathbb{F}|^{n-t+k-1}$.

(b) The total number of instances in $\mathcal{G}^A_{n,k,t}$ that have more than one solution is less than $\binom{n}{t}^2|\mathbb{F}|^{n-t+k-1}$.

Proof of (a). Let $|I_1 \cap I_2| = m$: note that $m \in \{0,\ldots,t-1\}$. Clearly the "free" (noise) points contribute $|\mathbb{F}|^{n-2t+m}$ choices. It remains to find the number of choices due to the coordinates of $\mathbf{c}$ that correspond to the positions indexed by $I_1 \cup I_2$. The first solution contributes $|\mathbb{F}|^k$ choices, whereas the second solution, if $m < k$, it contributes $|\mathbb{F}|^{k-m}$. If $m \geq k$ no second solution is feasible. So we have two cases: $m < k$, where $\#(\mathcal{G}^A_{n,k,t}(I_1) \cap \mathcal{G}^A_{n,k,t}(I_2)) = |\mathbb{F}|^{n-2t+2k}$, and $m \geq k$, where $\#(\mathcal{G}^A_{n,k,t}(I_1) \cap \mathcal{G}^A_{n,k,t}(I_2)) = |\mathbb{F}|^{n-2t+m+k}$, with $m \in \{k,\ldots,t-1\}$. As a result, independently of the choice of I_1, I_2, $\#(\mathcal{G}^A_{n,k,t}(I_1) \cap \mathcal{G}^A_{n,k,t}(I_2)) \leq |\mathbb{F}|^{n-t+k-1}$ (recall that $t > k$).

Proof of (b). it follows easily from the fact that the set of all instances of $\mathcal{G}^A_{n,k,t}$ that have more than one solution is a subset of $\cup_{I_1 \neq I_2} \mathcal{G}^A_{n,k,t}(I_1) \cap \mathcal{G}^A_{n,k,t}(I_2)$.

The following observation compares the number of elements of $\mathcal{G}^A_{n,k,t}$ and $\mathcal{G}^A_{n,k,t}(I)$ and in combination with the previous observation, it provides an estimate to the number of elements of $\mathcal{G}^A_{n,k,t}$:

Suppose $\log|\mathbb{F}| \geq 3n$. For any $I \subseteq \{1,\ldots,n\}$, $|I| = t$, it holds that $\binom{n}{t} - 2^{-n} \leq \frac{\#\mathcal{G}^A_{n,k,t}}{\#\mathcal{G}^A_{n,k,t}(I)} \leq \binom{n}{t}$.

(proof) By definition it holds that $\mathcal{G}^A_{n,k,t} = \cup_{|I|=t} \mathcal{G}^A_{n,k,t}(I)$. It follows from lemma that $\#\mathcal{G}^A_{n,k,t}(I) = \#\mathcal{G}^A_{n,k,t}(I')$ for all I, I'. Now fix some $I \subseteq \{1,\ldots,n\}$, $|I| = t$. It follows that,

$$\binom{n}{t}\#\mathcal{G}^A_{n,k,t}(I) - \sum_{I_1 \neq I_2} \#(\mathcal{G}^A_{n,k,t}(I_1) \cap \mathcal{G}^A_{n,k,t}(I_2)) \leq \#\mathcal{G}^A_{n,k,t} \leq \binom{n}{t}\#\mathcal{G}^A_{n,k,t}(I)$$

Next using the upper bound on $\sum_{I_1 \neq I_2} \#(\mathcal{G}^A_{n,k,t}(I_1) \cap \mathcal{S}_{n,k,t}(I_2))$ that follows from lemma, it follows that (using the facts $\log|\mathbb{F}| \geq 3n$, $\binom{n}{t} < 2^n$)

$$\sum_{I_1 \neq I_2} \#(\mathcal{G}^A_{n,k,t}(I_1) \cap \mathcal{G}^A_{n,k,t}(I_2)) < \frac{\binom{n}{t}^2 \#\mathcal{G}^A_{n,k,t}(I)}{|\mathbb{F}|} < \frac{\#\mathcal{G}^A_{n,k,t}(I)}{2^n}$$

It follows that

$$(\binom{n}{t} - \frac{1}{2^n})\#\mathcal{G}^A_{n,k,t}(I) \leq \#\mathcal{G}^A_{n,k,t} \leq \binom{n}{t}\#\mathcal{G}^A_{n,k,t}(I)$$

which completes the proof of item (i).

(ii) Now, it holds that $(\binom{n}{t} - 2^{-n})(|\mathbb{F}|)_n|\mathbb{F}|^{n-t+k} \leq \#\mathcal{S}_{n,k,t} \leq \binom{n}{t}(|\mathbb{F}|)_n|\mathbb{F}|^{n-t+k}$. The number of PR-instances of $\mathcal{S}_{n,k,t}$ with more than one solution is less than

$$\binom{n}{t}^2(|\mathbb{F}|)_n|\mathbb{F}|^{n-t+k-1}$$

(from observation (b) above). It follows that the ratio is less than

$$\binom{n}{t}^2(\binom{n}{t} - 2^{-n})^{-1}|\mathbb{F}|^{-1} < 2^{-n}$$

■

It is an immediate corollary from the above lemma that any PPT which samples the uniform distribution over $\mathcal{G}^A_{n,k,t}$ will select an instance $\mathbf{c}$ that has a unique solution with overwhelming probability $1-2^{-n}$. Consequently any instance $\mathbf{c} \in \mathcal{G}^A_{n,k,t}$ uniquely defines a solution $\mathbf{m} \in \mathbb{F}^*$ (with overwhelming probability). As a result F_{dec} is well-defined for most of the instances of $\mathcal{G}_{n,k,t}$ (i.e. F_{dec} has a unique output).

Proposition 84 *The collection* $\{\mathcal{G}^A_{n,k,t}(I)\}_{|I|=t}$ *is a hardcore orbit-division of the instance space* $\mathcal{G}^A_{n,k,t}$ *of the function* F_{dec}.

Hardness vs. Self-Reducibility in LC-DECODING

In this section we establish the fact that LC-DECODING is either solvable in probabilistic polynomial-time with very high probability or it is NOT partially random self-reducible w.r.t. the orbit-division $\{\mathcal{G}^A_{n,k,t}(I)\}_{|I|=t}$. This result is based on the following main theorem that states that the saturation factor of the hardcore orbit-division $\{\mathcal{G}^A_{n,k,t}(I)\}_{|I|=t}$ for F_{dec} is linear in the instance size n.

Theorem 85 *The saturation factor of the orbit-division* $\{\mathcal{G}^{A}_{n,k,t}(I)\}_{|I|=t}$ *is at most* $\lceil(\frac{\kappa}{\tau}+\alpha)n\rceil$. *where* $\alpha \in \mathbb{R}^{+}$.

The proof of the theorem is described in the remaining of this section. Let $\mathbf{c}_1,\ldots,\mathbf{c}_\rho$ be uniformly distributed elements of the orbit $\mathcal{G}^{A}_{n,k,t}(I)$ (ρ is to be determined later). Notice that each $\mathbf{c}_1,\ldots,\mathbf{c}_\rho$ is an instance of LC-DECODING with the same error-pattern. Consider the following $(n \times \rho)$-matrix

$$C = \begin{pmatrix} \uparrow & \uparrow & \cdots & \uparrow \\ \mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_\rho \\ \downarrow & \downarrow & \cdots & \downarrow \end{pmatrix}$$

Here, we introduce the following probabilistic procedure that given the elements $\mathbf{c}_1,\ldots,\mathbf{c}_\rho$, recovers k indices of the index-solution-set (the recovery of all t indices of I follows immediately).

`Find`(C)
Choose ρ rows of C at random to form square matrix M;
define $row[1],\ldots,row[\rho]$ to be the indices of the selected rows.
if $det(M) = 0$ call `Solve`(M);halt; else return(fail);
`Solve`(M)
for i=1 to k do
Perform Gaussian Elimination in M;
If the j-th row of M gets nullified set $sol[i] = row[j]$;
Swap the j-th row with the i-th row of M; swap($row[i], row[j]$);
output $\langle sol[1],\ldots,sol[k]\rangle$

First we compute the probability that the sub-procedure `Find` will actually find a minor M of C that contains at least $k+1$ rows whose index belongs to I. This is possible because the number of "good" rows of M (those with $row[i] \in I$) when selecting rows at random from C to form the minor M follows the hypergeometric distribution (this is because selection is done "without replacement"); if ρ is large enough the probability that less than $k+1$ good rows will be selected can be expressed using the Chvátal tail bounds for the hypergeometric distribution [Chv79].

Lemma 86 *Given that $\rho > \frac{nk}{t}$, the procedure* `Find` *will form a minor M that contains at least $k+1$ of the good rows with probability $1 - e^{-2(\frac{t}{n}-\frac{k}{\rho})^2\rho}$.*

Proof. Denote by λ the number of rows of the selected minor M that belong in I. The random variable λ follows a hypergeometric distribution with initial success probability t/n. We apply the Chvátal bound for the right tail of the hypergeometric distribution: for any $\epsilon > 0$, $\mathbf{Prob}[\lambda \leq (t/n - \epsilon)\rho] \leq \epsilon^{-2\epsilon^2 m}$. Let $\epsilon := \frac{t}{n} - \frac{k}{\rho}$: note that $\epsilon > 0$ by the condition on ρ. Then it holds that $\mathbf{Prob}[\lambda \leq k] \leq e^{-2\epsilon^2\rho}$. As a result we deduce that the probability that `Find` will call `Solve` is $\mathbf{Prob}[\lambda \geq k+1] = 1 - \mathbf{Prob}[\lambda \leq k] = 1 - e^{-2\epsilon^2 m}$. ■

One can show that a minor M that contains $k+1$ good rows is singular. This is because at least one of these rows can be written as a linear combination of the remaining k. Next, we need to show that any row in M that is not among the good rows (i.e. with $row[i] \notin I$) can be expressed as a linear combination of the remaining rows with very small probability. This ensures that the procedure `Solve` will discover all good rows, provided that the given minor M contains $k+1$ of the good rows.

Lemma 87 *Assume that $\rho > k$. The probability that a row i of the minor M with $row[i] \notin I$ can be expressed as a linear combination of the remaining rows of M is at most $1/|\mathbb{F}|$.*

Proof. The probability is taken over all possible choices of the instances $\mathbf{c}_1, \ldots, \mathbf{c}_\rho$ from $\mathcal{G}^A_{n,k,t}(I)$. Suppose we select a minor M of the matrix C defined by the rows $row[1], \ldots, row[\rho]$ and there is an $i \in \{1, \ldots, \rho\}$ with $row[i] \notin I$ s.t. the i-th row of M can be written as a linear combination of the remaining rows. This can happen with probability at most $1/|\mathbb{F}|$ since the i-th row is uniformly distributed over $\mathbb{F}^\rho$. ■

As a consequence of the above lemma a row i of M is nullified only if it is among those with $row[i] \in I$ (with overwhelming probability). As a result the sub-procedure

Solve will recover I with overwhelming probability (provided that M contains $k+1$ good rows).

Now observe that if $k := \kappa n$ and $t := \tau n$ and $\rho = \lceil (\frac{\kappa}{\tau} + \alpha)n \rceil$ it follows that $2(\frac{t}{n} - \frac{k}{\rho})^2 \rho = \Theta(n)$ and as a result the probability stated in lemma 86 is negligible in n. This completes the proof of theorem 85, since we have shown that I can be recovered in probabilistic polynomial-time provided that $\rho = \lceil (\frac{\kappa}{\tau} + \alpha)n \rceil$. ■

Because of theorem 85 we can show:

Theorem 88 *The function F_{dec} that corresponds to the* LC-DECODING *problem with parameters $[n, k := \kappa n, t := \tau n]$, is either*
(i) polynomial-time solvable (for most instances of $\mathcal{G}^{A}_{n,k,t}$), or,
(ii) NOT p-rsr with respect to the orbit-division $\{\mathcal{G}_{n,k,t}(I)\}_{|I|=t}$.

6.2.3 Saturation in the Polynomial Reconstruction Problem

In this section we concentrate on the decoding problem of Reed-Solomon codes. Although our generic results we presented in the previous section apply to the specific case of RS-decoding, here we demonstrate that in RS-Codes decoding the saturation factor of the hardcore orbit-division defined in section 6.2.2 is in fact constant. Recall the definition of the PR problem:

Definition 89 Polynomial Reconstruction (PR). *Given n, k, t and $\{\langle z_i, y_i \rangle\}_{i=1}^{n}$ with $z_i \neq z_j$ for $i \neq j$, output all $\langle p(x), I \rangle$ such that $p \in \mathbb{F}[x]$,* degree$(p) < k$, *$I \subseteq \{1, \ldots, n\}$, $|I| \geq t$ and $\forall i \in I (p(z_i) = y_i)$ (note that we assume that at least one solution exists, i.e. there is a promise that the number of "errors" is less than $n - t$).*

The instance space is denoted by $\mathcal{S}_{n,k,t}$ and contains sets of pairs $\{\langle z_i, y_i \rangle\}_{i=1}^{n}$ as defined above. The decoding function is denoted by F^{RS}_{dec} and maps instances $X \in \mathcal{S}_{n,k,t}$ to a polynomial solution p. Again, F^{RS}_{dec} is well defined only in the case there exists a unique solution.

PR, interpreted as a coding theoretic problem asks for all messages that agree with at least t positions of the received corrupted codeword. When $t \geq \frac{n+k}{2}$ then PR$[n,k,t]$ has at most one solution and it can be found with the algorithm of Berlekamp and Welch [BW86] ($\frac{n+k}{2}$ is the error-correction bound of the Reed-Solomon codes). When t is beyond the error-correction bound then having more than one solution is possible. Sudan proposed an algorithm that solves the PR beyond the error-correction bound when $t \geq \sqrt{2kn}$ in [Sud97] and later in [GS98]. Guruswami and Sudan presented an algorithm that solves the PR for $t > \sqrt{kn}$. In [GRS95] it was shown that when $t > \sqrt{kn}$ the number of solutions is bounded by a polynomial.

Structure of the Instance Space

Let $I \subseteq \{1,\ldots,n\}$ with $|I| = t$. For some fixed, pairwise different $z_1,\ldots,z_n$, we denote by $\mathcal{S}_{n,k,t}(I)$ the subset of $\mathcal{S}_{n,k,t}$ so that for any $X \in \mathcal{S}_{n,k,t}(I)$ it holds that $X := \{\langle z_i, y_i\rangle\}_{i=1}^{n}$ has a solution of the form $\langle p, I\rangle$.

Following similar arguments as in section 6.2.2, one can show that

Proposition 90 *(i) The collection $\{\mathcal{S}_{n,k,t}(I)\}_{|I|=t}$ is a hardcore orbit-division of the instance space $\mathcal{S}_{n,k,t}$.*
(ii) A random instance over $\mathcal{S}_{n,k,t}$ has a unique solution with probability at least $1 - 2^{-n}$, provided that $\log|\mathbb{F}| \geq 2n$.

Item (ii) suggests that F_{dec}^{RS} is well defined for most instances of $\mathcal{S}_{n,k,t}$.

An immediate corollary of theorem 88 is the following:

Corollary 91 *The function F_{dec}^{PR} which corresponds to the PR problem, with parameters $[n, k := \kappa n, t := \tau n]$, is either*
(i) polynomial-time solvable (for most instances $\mathcal{S}_{n,k,t}$), or,
(ii) NOT p-rsr with respect to the orbit-division $\{\mathcal{S}_{n,k,t}(I)\}_{|I|=t}$.

Hardcore Orbit Divisions in PR with Constant Saturation

This section follows the exposition of section 6.2.2. Due to the additional structure of RS-codes, we can show a much stronger result with respect to the saturation factor of the hardcore orbit-division:

Theorem 92 *The saturation factor of the orbit-division* $\{\mathcal{S}_{n,k,t}(I)\}_{|I|=t}$ *is at most* $\lceil \alpha \frac{1-r}{r-\kappa} \rceil$.

Proof. Let $X_1, \ldots, X_\rho$ be randomly distributed over $\mathcal{S}_{n,k,t}(I)$. Set $X_\ell = \{\langle z_i, y_{i,\ell} \rangle\}_{i=1}^{n}$ for $\ell = 1, \ldots, \rho$. We will determine a value for ρ that allows us to recover I. Define the following matrix $\mathcal{E}$, where $d = t-k-1$, and $m = \rho(t-k-1)+t-1$:

$$\mathcal{E} = \left(\begin{array}{cccc|c|cccc|cccc} y_{1,1} & z_1 y_{1,1} & \cdots & z_1^d y_{1,1} & \cdots & y_{1,r} & z_1 y_{1,r} & \cdots & z_1^d y_{1,r} & 1 & z_1 & \cdots & z_1^{k-1+d} \\ y_{2,1} & z_2 y_{2,1} & \cdots & z_2^d y_{2,1} & \cdots & y_{2,r} & z_2 y_{2,r} & \cdots & z_2^d y_{2,r} & 1 & z_2 & \cdots & z_2^{k-1+d} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ y_{n,1} & z_n y_{n,1} & \cdots & z_n^d y_{n,1} & \cdots & y_{n,r} & z_n y_{n,r} & \cdots & z_n^d y_{n,r} & 1 & z_n & \cdots & z_n^{k-1+d} \end{array} \right)$$

$\mathcal{E}^d[X_1, \ldots, X_r]$ is of dimensions $n \times m$. Consider the following algorithm:

`Find`($\mathcal{E}$)
Choose m rows of the matrix to form square matrix M:
define $row[1], \ldots, row[m]$ to be the indices of the selected rows.
if Det(M) = 0 call `Solve`(M);halt; else return(fail)
`Solve`(M)
for i=1 to t do
Perform Gaussian Elimination in M:
If the j-th row of M gets nullified set $sol[i] = row[j]$;
Swap the j-th row with the i-th row of M; swap($row[i], row[j]$);
output $\langle sol[1], \ldots, sol[t] \rangle$

With similar arguments as in lemma 86, we show

Lemma 93 *Given that* $n > t > k+1$ *and* $\rho > (\frac{n}{t} - 1)\frac{t-1}{t-k-1}$ *the procedure* `Find` *will form a minor* M *that contains at least* t *of the good rows with probability* $1 - e^{-2(\frac{t}{n} - \frac{t-1}{m})^2 m}$.

Proof. Denote by λ the number of rows selected by `Find` that belong in I. The random variable λ follows a hypergeometric distribution with initial success probability t/n. We apply the Chvátal bound for the right tail of the hypergeometric distribution: for any $\epsilon > 0$, $\mathbf{Prob}[\lambda \leq (t/n - \epsilon)m] \leq e^{-2\epsilon^2 m}$. Let $\epsilon := \frac{t}{n} - \frac{t-1}{m}$ then it holds that $\mathbf{Prob}[\lambda \leq t-1] \leq e^{-2\epsilon^2 m}$. As a result we deduce that the probability that `Find` will call `Solve` is $\mathbf{Prob}[\lambda \geq t] = 1 - \mathbf{Prob}[\lambda \leq t-1] = 1 - e^{-2\epsilon^2 m}$. ■

One can show that a minor M of $\mathcal{E}$ that contains t of the good rows will be singular. This is because each one of the good rows in the matrix $\mathcal{E}$ is of the form $\langle P_1(z_i), \ldots, P_m(z_i) \rangle$ where $P_1, \ldots, P_m$ are polynomials of degree at most $t-2$.

Next with similar arguments as in lemma 87 one can show that in such a minor M, the probability that a certain row, that is not among the good ones, can be written as a linear combination of the remaining rows is less than $1/|\mathbb{F}|$.

Now if $k := \kappa n$, $t := \tau n$ and $\rho = \lceil \alpha \frac{1-\tau}{\tau-\kappa} \rceil$ for some $\alpha > 1$. Then, it follows that the probability in lemma 93 is of the form $1 - Be^{-cn}$ for some constants $B, c \in \mathbb{R}$. The proof of theorem 92 follows. ■

Chapter 7

Future Work and Directions

Polynomial reconstruction based Cryptography constitutes a novel paradigm for the exploitation of Decoding Problems of families of Codes in the cryptographic setting. The results of this thesis suggest that the problem of Polynomial Reconstruction, the Decoding Problem of Reed-Solomon codes, is robust in the cryptographic sense under the related decisional Assumption.

We provided various different venues over which this novel intractability assumption yielded primitives and protocols with unique properties and applications, including a probabilistic one-way function with strong concealment properties, commitment schemes with sublinear decommitment witness, stateful cryptosystems, and secure games with polynomial expressions. Also we have investigated the solvability of the Multisample Polynomial Reconstruction, and pointed to its equivalence to the Decoding Problem of Interleaved Reed-Solomon Codes. Finally we discovered interesting hardness/self-reducibility tradeoffs in Decoding Problems and we introduced various new notions for characterizing such tradeoffs in computational complexity problems, such as the notion of hardcore orbit-divisions and the notion of the saturation factor.

There are several directions for future research that are spawned from this thesis. Below we summarize the most important ones:

- Investigation of Polynomial Reconstruction Based Cryptography from a practical viewpoint: concrete implementation and experimentation with parameters.

- Polynomial Reconstruction as an intractability assumption has many natural applications in cases where cryptographic-keys are "fuzzy" (i.e. the same key should be allowed to be slightly different from time to time). This in turn, has a major application in the biometric extraction of cryptographic keys. Our results can provide a formal security foundation for such constructions, and it is an interesting research direction to investigate these interrelations further.

- Considering other codes instead of Reed-Solomon from a cryptographic hardness perspective; examples of codes that can be considered next include Chinese Remaindering Codes and Algebraic Geometric Codes, such as Goppa codes.

- Novel protocol constructions based on Polynomial Reconstruction. To this effect we present in the next section an outline for building a Key-exchange protocol based solely on Polynomial Reconstruction.

7.1 A model for Key-Exchange Using PR

Modern Cryptography started with Diffie and Hellman's fundamental concept of "Asymmetric Cryptography" (aka Public-Key Cryptography) [DH76]. It enables two entities that have never met before to start secure communication. In particular, Asymmetric Cryptography enables the entities to establish a common secure key (which the eavesdroppers cannot get in spite of the fact that the transcript of the entire interaction is public). The key exchange can be done via a specialized protocol (at the end of which the parties agree on a secret value) or via a public-key scheme where one of the parties publishes a public-key cryptosystem that the other party uses to encrypt the common key with. Another introduction of this basic notion was

Merkle's puzzles [Mer78] (that achieved only a polynomial work differential between the two entities and the eavesdroppers). As was noted in these seminal papers, the security of Asymmetric Cryptography, being based on publicly available transcripts, relies on computational complexity and the inability of the eavesdroppers to solve a hard problem (thus, in some sense this field can be viewed as "the applied side of computational complexity").

Following the above seminal works, the last 25 years led to only a handful of basic mathematical techniques which led themselves to the construction of asymmetric cryptographic methods. Methods based on the index calculus method (discrete logarithm) over a finite group [DH76] (later generalized to other groups such as groups over elliptic curves and recently over non-abelian groups) were introduced. Other methods based on the intractability of factoring have been designed starting from the works of [ARS78, Rab79, GM84] (while another early family of knapsack-based solutions, was later found to be tractable). Other schemes are based on the hardness of decoding random linear codes [McE78], polynomial equations [IM85] and on lattice problems [AD97].

Given this limited choice of mathematical infrastructure to base Asymmetric Cryptography upon, finding novel mathematical techniques and assumptions that can support it, is perhaps the central issue in modern Cryptography research.

Here we propose a variant of the polynomial reconstruction (PR) problem as a sole base for key exchange. Variants of this problem has already been employed within cryptographic protocols in other works beyond this thesis, (e.g. [NP99], however, not by themselves, but rather on top of Oblivious Transfer channels).

Key exchange protocols have been using one of the following paradigms:

1. The generic Merkle puzzle paradigm based on a set of moderately hard problems; however this "generic" method exhibits only a polynomial difference be-

tween the attacker and the legal parties.

2. Public-Key Cryptosystem based, where a problem has a secret "trapdoor" information, however note that for the PR problem no trapdoor is known.

3. The Diffie-Hellman key exchange paradigm, where one employs the commutativity of multiplication of preimages of a one-way function, however no commutativity is present in the PR problem as is.

Our goal is to present a protocol for key exchange where the eavesdropper has to work exponentially (or at least super polynomially) in the work of the users, thus paradigm 1 does not apply. Now, since our underlying problem does not have a structure which allows one of paradigms 2 or 3 to work, we needed a new paradigm. Our paradigm includes the following steps:

- Each party commits to a random set encoded within a public structure which hides "exponentially many subsets."

- Each party generates a "surface" based on its own subset. Each party, then, projects its surface on the structure (all potential subsets) of the other party. Subsequently it partially randomizes this projection.

- The parties exchange these partially randomized projections. Based on the received projection and the actual random set each party employs error-correcting techniques to recover the common key.

While computing on the two projections is, in fact, not commutative, their evaluations along a specific curve generates a computation which is commutative with respect to a common point on the curve. The limited commutativity is generated by the algebraic structure of the set encoding in a PR instance will potentially allow overcoming the obstacle that the PR problem itself (as posed) has no commutative

structure in it. This lack of commutativity in PR is in contrast with the Diffie Hellman problem where exponentiation with two different values is commutative.

7.1.1 Preliminaries and Definitions

We recall some definitions that are useful in this section. An *ensemble* parameterized by $n \in \mathbb{N}$ is a collection of sets $\{A_n\}_{n\in\mathbb{N}}$.

Definition 94 *Two ensembles* D *and* R *parameterized by* $n \in \mathbb{N}$ *are called polynomial time indistinguishable, if for any probabilistic polynomial-time predicate* $\mathcal{A}$ *it holds that,*

$$| \mathbf{Prob}_{X\in_U \mathsf{D}}[\mathcal{A}(X) = 1] - \mathbf{Prob}_{X\in_U \mathsf{R}}[\mathcal{A}(X) = 1] |$$

is a negligible function in n.

7.1.2 The Intractability Assumption

To simplify our exposition we will use pseudorandomness of PR-instances as our intractability assumption in this section (see section 3.3). Let us start by recalling the definition of the problem.

Definition 95 Polynomial Reconstruction (PR). *Given* $n, k, t \in \mathbb{N}$ *and the pairs* $\{\langle z_1, y_1\rangle, \ldots, \langle z_n, y_n\rangle\}$ *. output all* $P(x) \in \mathbb{F}[x]$ *such that* degree(p) $< k$, *and* $P(z_i) = y_i$ *for at least* t *distinct indices* $i \in \{1, \ldots, n\}$.

Our first intractability assumption suggests that it is impossible to distinguish between random sets of points and PR-instances for error-rates beyond what can be decoded with the current knowledge for PR-solvability. We formalize our intractability assumption below. Recall that the parameters $[n, k, t]$ will be called sound for the PR-problem if they satisfy $t < \sqrt{n(k-1)}$, and the brute-force algorithm against PR requires exponential time in n (see section 3.1.2)

Assumption #1 with parameters $[n, k, t]$. Consider the following two ensembles:

$$\mathsf{R}_1 := \left\{ \{\langle z_i, y_i \rangle\}_{i=1}^n \mid z_i, y_i \in \mathbb{F} \right\}$$

$$\mathsf{D}_1 := \left\{ \{\langle z_i, y_i \rangle\}_{i=1}^n \mid z_i \in \mathbb{F},\ \exists I, |I| = t,\ \exists P \in \mathbb{F}[x],\ \forall i \in I : P(z_i) = y_i \right\}$$

where the polynomial P in the ensemble D_1 is of the form $P(x) = \sum_{j=0}^{k-1} c_j x^j$. We assume that whenever the parameters $[n, k, t]$ are sound, the two ensembles $\mathsf{R}_1, \mathsf{D}_1$ are indistinguishable for any probabilistic polynomial-time predicate.

Note that assumption #1 can be derived from the DPR assumption, see section 3.3.

Our second intractability assumption is defined in similar terms, but for bi-variate polynomials. We start with the definition of the Polynomial Reconstruction problem for bi-variate polynomials.

Definition 96 Bi-variate Polynomial Reconstruction (BPR). *Given parameters* $n,\ d_x, d_y \in \mathbb{N}$ *and* $\gamma \in \mathbb{Q}^+$ *with* $\gamma < 1$. *and* $\{\langle z_1, y_1, q_1 \rangle, \ldots, \langle z_n, y_n, q_n \rangle\}$. *output all* $Q(x, y) \in \mathbb{F}[x, y]$ *such that* $\text{degree}(Q, x) \leq d_x, \text{degree}(Q, y) \leq d_y$, *and* $Q(z_i, y_i) = q_i$ *for at least* γn *distinct indices* $i \in \{1, \ldots, n\}$.

The BPR problem is an extension of the univariate case and typically many decoding approaches for PR can be extended to the bi-variate case, see e.g.[Sud97]. On the other hand, there is not an explicitly stated bound for the solvability of the BPR with the above formulation, so we will sketch briefly the main technique of [Sud97],[GS98] (which is the strongest known decoding technique with respect to the error-rate it can correct) and we will extend it to the case of BPR, to obtain a bound on the solvability of BPR. Given a set of points $\{\langle z_i, y_i, q_i \rangle\}_{i=1}^n$ and parameters d_x, d_y, γ we interpolate a non-zero tri-variate polynomial $Q'(x, y, z)$ of degrees d_1, d_2, d_3 so that for any bi-variate polynomial Q of degrees d_x, d_y that fits γn of the given points, it holds that

$Q'(x, y, Q(x, y))$ is identically the 0-polynomial; this method will allow us to use polynomial factorization to recover all possible solutions Q. Observe such a polynomial Q' can be found by standard linear algebra, provided that (i) $Q'(z_i, y_i, q_i) = 0$, for all $i = 1, \ldots, n$, (ii) $(d_1 + d_x d_3)(d_2 + d_y d_3) < \gamma n$, and (iii) $d_1 d_2 d_3 > n$. The condition (ii) suggests that $d_1 d_2$ and $(d_3)^2 d_x d_y$ can be at most γn and since $d_1 d_2 d_3 > n$ it should hold that $\gamma > \sqrt[3]{\xi d_x d_y / n}$, where ξ is some real constant greater equal to 1 (note that we purposely adopt a worst-case approach for this analysis, since we are not interested to solve BPR, but rather find an asymptotic lower bound in the solvability of BPR, using the central ideas of [GS98]). We will call the parameters $[n, d_x, d_y, \gamma]$ sound for the BPR problem if $\gamma < \sqrt[3]{\xi d_x d_y / n}$ and the brute-force approach against BPR requires exponential time in n. Next we formalize our second intractability assumption, that is an extension of assumption #1 for the case of bi-variate Polynomial Reconstruction.

Assumption #2 with parameters $[n, \gamma, d_x, d_y]$. Consider the following two ensembles:

$$\mathsf{R}_2 := \left\{ \{\langle z_i, y_i, q_i \rangle\}_{i=1}^n \mid z_i, y_i, q_i \in \mathbb{F} \right\}$$

$$\mathsf{D}_2 := \left\{ \{\langle z_i, y_i, q_i \rangle\}_{i=1}^n \mid z_i, y_i \in \mathbb{F},\ \exists I, |I| = \gamma n,\ \exists Q \in \mathbb{F}[x, y],\ \forall i \in I : Q(z_i, y_i) = q_i \right\}$$

where the polynomial Q in the ensemble D_2 is of the form

$$Q(x, y) = \sum_{j=0}^{d_x} \sum_{j'=0}^{d_y} c_{j,j'} x^j y^{j'}$$

We assume that whenever the parameters $[n, \gamma, d_x, d_y]$ are sound, the two ensembles $\mathsf{R}_2, \mathsf{D}_2$ are indistinguishable for any probabilistic polynomial-time predicate.

7.1.3 Our Model: Two-Phase Key-Exchange

A two-phase key-exchange is defined by three probabilistic algorithms GEN, ACT, GET.

We give an informal description first. The public-key generation algorithm GEN, on input 1^n (where n is the security parameter) it outputs a tuple $\langle X, I, V \rangle$. The public-portion is X where the parts I, V are kept secret (i.e. the constitute the private-key). The space of public-keys for the security parameter n is subsumed in a space $\mathcal{X}_n$. Similarly the set of private-keys V is contained in a space $\mathcal{V}_n$. We assume that $\mathcal{V}_n$ is a (say) multiplicative group w.r.t. some operation $\odot$.

The action algorithm ACT operates on a public-key X of some player and combines it with the private information I, V of the acting player. The output Y of the ACT procedure is transmitted in the open from the acting player to the receiving player. The set of actions is subsumed in a space $\mathcal{Y}_n$. Finally the receiving player uses GET on the action of another player together with his private-key and obtains the exchanged key.

Let us now describe in more detail how the key exchanged is performed between two players A, B, in two phases.

Phase I. Both A, B use the key-generation algorithm GEN to obtain their public and private keys. In particular, player A obtains $\langle X_A, I_A, V_A \rangle \leftarrow \mathsf{GEN}(1^n)$ and publishes X_A. Player B, similarly, obtains $\langle X_B, I_B, V_B \rangle \leftarrow \mathsf{GEN}(1^n)$ and publishes X_B.

Phase II. Player A obtains the value $Y_A \leftarrow \mathsf{ACT}(X_B, V_A)$ and transmits Y_A to player B. Similarly, player B obtains the value $Y_B \leftarrow \mathsf{ACT}(X_A, V_B)$ and transmits it to player A.

Finally, each player uses GET to compute the common key $V_A \odot V_B$. In particular, player A computes $\mathsf{GET}(Y_B, I_A)$ and player B computes $\mathsf{GET}(Y_A, I_B)$.

Definition 97 *(Correctness) The error probability of a two-phase key-exchange protocol* GEN, ACT, GET, *is defined as*

$$\mathbf{Prob}[\mathsf{GET}(\mathsf{ACT}(X_A, V_B), I_A) \neq V_A \odot V_B]$$

The probability is taken over the internal coin-tosses of ACT *and* GET, *and all choices* $\langle X_A, I_A, V_A\rangle \leftarrow \mathsf{GEN}(1^n)$ *and* $\langle X_B, I_B, V_B\rangle \leftarrow \mathsf{GEN}(1^n)$. *A key-exchange protocol is correct if its error-probability is negligible in* n.

Definition 98 *(Security) A key-exchange protocol* GEN, ACT, GET, *is secure if the following two ensembles are polynomial-time indistinguishable:*

$$\mathsf{R} = \left\{ \left\langle X, X', Y, Y', V \right\rangle \mid X, X' \in \mathcal{X}_n, Y, Y' \in \mathcal{Y}_n, V \in \mathcal{V}_n \right\}$$

$$\mathsf{D} = \left\{ \left\langle X_A, X_B, \mathsf{ACT}(X_B, V_A), \mathsf{ACT}(X_A, V_B), V_A \oplus V_B \right\rangle \mid \right.$$
$$\left. \langle X_A, I_A, V_A\rangle \leftarrow \mathsf{GEN}(1^n), \langle X_B, I_B, V_B\rangle \leftarrow \mathsf{GEN}(1^n) \right\}$$

The above definition suggests that protocol transcripts together with the shared key they define, to be indistinguishable from truly random tuples. The modeling of the definition follows the security formalization of the Diffie-Hellman Key-exchange protocol that requires protocol transcripts and the shared key they define $\langle g^a, g^b, g^{ab}\rangle$, to be indistinguishable from truly random tuples of the form $\langle g^a, g^b, g^c\rangle$ (which is the Decisional Diffie Hellman Assumption).

7.1.4 Key-Exchange Protocol based on Polynomial Reconstruction

In this section we define the key-exchange protocol based on Polynomial Reconstruction following the model of a two-phase key-exchange. In what follows we describe the three probabilistic algorithms GEN, ACT and GET.

The security parameter of the protocol is n, and also we employ the parameters k, t, d_x, d_y, γ thought of as functions in n. Note that $k, t, d_x, d_y \in \mathbb{N}$ and $\gamma \in \mathbb{Q}^+$. $\mathbb{F}$ denotes a large finite field, s.t. $\log|\mathbb{F}|$ is polynomial related in n.

- The $\mathsf{GEN}_{k,t}$ function. Given 1^n, $\mathsf{GEN}_{k,t}$ selects random $z_i \in \mathbb{F}$ for $i = 1, \ldots, n$ and a random $I \subseteq \{1, \ldots, n\}$ with $|I| = t$. Then, it selects $P \in \mathbb{F}[x]$ with degree

less than k, and sets $y_i = P(z_i)$ for all $i \in I$; for all $i \notin I$ a random $y_i \in \mathbb{F}$ is selected. The output of $\mathsf{GEN}_{k,t}$ is the tuple $\langle \{\langle z_i, y_i \rangle\}_{i=1}^n, I, P(0) \rangle$. The general space into which the public-keys are contained is

$$\mathcal{X}_n := \left\{ \{\langle z_i, y_i \rangle\}_{i=1}^n \mid z_i, y_i \in \mathbb{F} \right\}$$

the space for the shared key is $\mathcal{V}_n := \mathbb{F}$, and obviously it is a group with respect to multiplication (with 0 excluded).

- The $\mathsf{ACT}_{\gamma,d_x,d_y}$ function. On input $X := \{\langle z_i, y_i \rangle\}_{i=1}^n$ and $\alpha \in \mathbb{F}$ it operates as follows: first it selects random $J \subseteq \{1, \ldots, n\}$ of size $|J| = \gamma n$. Then, it samples a random polynomial $Q \in \mathbb{F}[x, y]$ so that $Q(x, y) = \alpha y + \sum_{j=1}^{d_x} \sum_{j'=0}^{d_y - 1} c_{j,j'} x^j y^{j'}$ and sets $q_j = Q(z_j, y_j)$ for all $j \in J$. Then it selects all q_j for $j \notin J$ at random from $\mathbb{F}$. Finally it outputs $q_1, \ldots, q_n$. It follows that the general space of the actions is

$$\mathcal{Y}_n := \left\{ \{q_1, \ldots, q_n\} \mid q_i \in \mathbb{F} \right\}$$

Note that for convenience we will assume that the output of

$$\mathsf{ACT}_{\gamma,d_x,d_y}(\{\langle z_i, y_i \rangle\}_{i=1}^n, V)$$

is the set of tuples $\{\langle z_i, y_i, q_i \rangle\}_{i=1}^n$.

- The GET function. Given $Y = \{\langle z_i, y_i, q_i \rangle\}_{i=1}^n$, and I the function GET parses Y to collect the tuples $\{\langle z_i, q_i \rangle\}_{i \in I}$. Subsequently it feeds these tuples to the Guruswami-Sudan polynomial reconstruction algorithm ([GS98]) and collects all polynomials returned by the algorithm of degree at most $\delta := d_x + (k-1)(d_y - 1)$ that cover more than $\sqrt{t\delta}$ points. If there is a single polynomial R as output, GET returns $R(0)$, otherwise GET returns **fail**.

7.1.5 Complexity of the Key-Exchange Protocol

We assume generic implementations of the basic finite field operations: addition is of complexity $\mathcal{O}(\log|\mathbb{F}|)$, and multiplication of complexity $\mathcal{O}((\log|\mathbb{F}|)^2)$. We remark that in many settings multiplication can be optimized further.

Phase I. An execution of the $\mathsf{GEN}_{k,t}$ function requires: (i) generation of $2n-t+k$ random elements of $\mathbb{F}$; (ii) polynomial evaluation of t values of a polynomial of degree less than k over $\mathbb{F}$: each evaluation requires k multiplications (and additions) as a result $\mathcal{O}(tk(\log|\mathbb{F}|)^2)$. (iii) The size of each public-key is $2n$ elements of $\mathbb{F}$, i.e. $\mathcal{O}(n(\log|\mathbb{F}|))$.

Phase II. An execution of the $\mathsf{ACT}_{d_x,d_y,\gamma}$ function requires: (i) The generation of $(d_x-1)(d_y-1)$ random elements of $\mathbb{F}$ and the evaluation of the polynomial Q on n points: each evaluation requires $\mathcal{O}(d_xd_y|(\log|\mathbb{F}|)^2)$ time. In total the time required is $\mathcal{O}(nd_xd_y(\log|\mathbb{F}|)^2)$. The communication complexity of this phase II is $\mathcal{O}(n\log|\mathbb{F}|)$.

Complexity of the GET function. It requires parsing the n elements to collect those in the secret index-set, and the invocation of the GS-algorithm on a set of t points to recover all polynomials of degree at most $\delta := d_x+(k-1)(d_y-1)$ that cover at least $\sqrt{t\delta}$ of the points. The time complexity of the GS-algorithm is deterministic polynomial-time, for more details we refer to [Gur01].

We note that the overall time complexity of the exchange for each player is $\mathcal{O}(t^2\log t+d_xd_yn)$ field operations and the overall communication is $\mathcal{O}(n)$ field elements. The size of the finite field $\log|\mathbb{F}|$ does not interfere with the security of the key-exchange and can be selected independently of the security parameter n.

7.1.6 Correctness of the Key-Exchange

In this section, we will argue that there exist specific choices of the parameters k,t,d_x,d_y,γ, so that the key-exchange protocol defined by $\mathsf{GEN}_{k,t},\mathsf{ACT}_{d_x,d_y,\gamma},\mathsf{GET}$

in section 7.1.4 is correct according to definition 97.

Lemma 99 *Assume that* $d_x + (k-1)(d_y - 1) < \gamma^2 t$. *Let* $\langle X, I, V \rangle \leftarrow \mathsf{GEN}_{k,t}(1^n)$, *and* $Y = \{\langle z_i, y_i, q_i \rangle\}_{i=1}^n \leftarrow \mathsf{ACT}_{d_x,d_y,\gamma}(X, V')$, *where* $V' \in \mathbb{F}$.

1. *The set of points* $\{\langle z_i, q_i \rangle\}_{i \in I}$ *contain a number of points of a polynomial* R *of degree at most* $\delta = d_x + (k-1)(d_y - 1)$ *such that* $R(0) = V \cdot V'$.

2. *The probability that* R *is among the solutions reported by the Guruswami-Sudan algorithm when it is executed on* $\{\langle z_i, q_i \rangle\}_{i \in I}$ *is at least* $1 - e^{-2(\frac{t-\sqrt{t\delta}}{(1-\gamma)n} - \frac{t}{n})^2(1-\gamma)n}$

3. *The probability that the Guruswami-Sudan algorithm on input* $\{\langle z_i, q_i \rangle\}_{i \in I}$ *reports a solution other than* R *is* $(\frac{t}{\sqrt{t\delta}})/|\mathbb{F}|^{\sqrt{t\delta}-\delta}$.

Proof. (*1*). Observe that the polynomial Q selected by $\mathsf{ACT}_{d_x,d_y,\gamma}$ is of degree d_x on the first variable and of degree $d_y - 1$ on the second variable. It follows immediately that the points $\{\langle z_i, q_i \rangle\}_{i \in I}$ contain points of the polynomial $R(x) = Q(x, P(x))$, where $P(x) \in \mathbb{F}[x]$ is the polynomial of degree less than k that was selected by $\mathsf{GEN}_{k,t}$. The polynomial R is of degree $\delta = d_x + (k-1)(d_y - 1)$ and it is easy to verify that since $Q(0, y) = V'y$, it follows that $R(0) = (Q(0), P(0)) = V \cdot V'$ (since $P(0) = V$).
(*2*). Let h denote the number of points $\langle z_i, y_i \rangle$ of X for which it holds $i \in I$ but $i \notin J$ where J is the subset of size γn selected by $\mathsf{ACT}_{d_x,d_y,\gamma}(X, V)$.

Let B be some value s.t. $1 < B < \gamma t$. First we show that,

$$\mathbf{Prob}[h \geq t - B] \leq e^{-2(\frac{t-B}{(1-\gamma)n} - \frac{t}{n})^2(1-\gamma)n}$$

To see this observe that h is a random variable that follows the hypergeometric distribution with initial probability of success t/n. The number of trials is $(1-\gamma)n$. Using the Chvátal bound [] for the hypergeometric distribution we obtain that for any $\epsilon > 0$,

$$\mathbf{Prob}[h \geq (\epsilon + \frac{t}{n})(1-\gamma)n] \leq e^{-2\epsilon^2(1-\gamma)n}$$

We select $\epsilon = \frac{t-B}{(1-\gamma)n} - \frac{t}{n}$. The given condition on B suggests that $\epsilon > 0$, and it follows that

$$\textbf{Prob}[h \geq t - B)] \leq e^{-2[\frac{t-B}{(1-\gamma)n} - \frac{t}{n}]^2(1-\gamma)n}$$

In order for the Guruswami-Sudan to output R, it should hold that the number of points of $\{\langle z_i, q_i \rangle\}_{i \in I}$ that are covered by elements of R should be greater than $\sqrt{t\delta}$. This number is $t - h$, therefore we want to evaluate the probability $\textbf{Prob}[t - h \leq \sqrt{t\delta}] = \textbf{Prob}[h \geq t - \sqrt{t\delta}]$. By setting $B = \sqrt{t\delta}$ above we obtain the result (and we can do that since $B = \sqrt{t\delta} < \gamma t$ according the assumption on d_x, d_y, γ, t given in the statement of the lemma).

(*3*). In order for the described event to happen, it should hold that some polynomial R' of degree at most δ agrees with at least $\sqrt{t\delta}$ points of $\{\langle z_i, q_i \rangle\}_{i \in I}$.

Consider the following experiment: given t points $\{\langle z_i, r_i \rangle\}_{i=1}^{t}$ in the graph of a polynomial $R(x)$ that has degree at most δ, we randomize e of them. We want to compute the probability that the GS-algorithm outputs a list of polynomials that contains a polynomial other than R.

Denote by A_1 the total number of different outcomes of the experiment and by A_2 the total number of outcomes for which the GS-algorithm reports a solution other than R. It is easy to see that $A_1 = |\mathbb{F}|^e$. In order to approximate A_2, observe that at least $\sqrt{t\delta}$ points should belong to the graph of some polynomial R'. Let the overlap of R' and R be $m \in \{0, \ldots, \delta - 1\}$. Then in the experiment one should select $\sqrt{t\delta} - m$ points from the graph of R' as part of the randomization. There is a total of $|\mathbb{F}|^{\delta - m + 1}$ choices for R', and a total of $|\mathbb{F}|^{e - \sqrt{t\delta} + m}$ choices for the remaining randomizations. It follows that $A_2 < \binom{t}{\sqrt{t\delta}} |\mathbb{F}|^{\delta + e - \sqrt{t\delta} + 1}$. The probability of getting a different polynomial solution is:

$$\frac{A_2}{A_1} < \frac{\binom{t}{\sqrt{t\delta}}}{|\mathbb{F}|^{\sqrt{t\delta} - \delta - 1}}$$

This completes the proof. ■

Theorem 100 *The key-exchange protocol of section 7.1.4 with parameters* $k, t, d_x, d_y, \gamma, \log|\mathbb{F}|$ *that satisfy* $d_x + (k-1)(d_y - 1) < \gamma^2 t$ *has error-probability at most*

$$\epsilon^{-(\frac{t-\sqrt{t\delta}}{(1-\gamma)n}-\frac{t}{n})^2(1-\gamma)n} + \frac{\binom{t}{\sqrt{t\delta}}}{|\mathbb{F}|^{\sqrt{t\delta}-\delta-1}}$$

Proof. The proof is immediate based on lemma 99 (*2* and *3*). ■

7.1.7 Security of the Key-Exchange

Lemma 101 *For parameters* n, γ, d_x, d_y, *consider the following two ensembles:*

$$\mathsf{R}_3 := \left\{ \left\langle \{\langle z_i, y_i, q_i\rangle\}_{i=1}^n, \{\langle z_i', y_i', q_i'\rangle\}_{i=1}^n, \rho \right\rangle \mid z_i, y_i, q_i, z_i', y_i', q_i', \rho \in \mathbb{F} \right\}$$

$$\mathsf{D}_3 := \left\{ \left\langle \{\langle z_i, y_i, q_i\rangle\}_{i=1}^n, \{\langle z_i', y_i', q_i'\rangle\}_{i=1}^n, \alpha\alpha' \right\rangle \mid \exists I, I', Q, Q', |I| = |I'| = \gamma n, \right.$$
$$\left. \forall i \in I : q_i = Q(z_i, y_i), \forall i \in I' : q_i' = Q'(z_i, y_i) \right\}$$

where the polynomials Q, Q' *in the ensemble* D_3 *are of the form*

$$Q(x,y) = \alpha y + \sum_{j=1}^{d_x} \sum_{j'=0}^{d_y - 1} c_{j,j'} x^j y^{j'}$$

and

$$Q'(x,y) = \alpha' y + \sum_{j=1}^{d_x} \sum_{j'=0}^{d_y - 1} c'_{j,j'} x^j y^{j'}$$

Under assumption #2 with parameters $[n, \gamma, d_x - 1, d_y - 1]$, *it holds that* R_3 *and* D_3 *are indistinguishable.*

Proof. Let $\mathcal{A}$ be a distinguisher for the ensembles D_3 and R_3. We will describe how to use $\mathcal{A}$ to construct a distinguisher $\mathcal{A}'$ for the ensembles D_2 and R_2.

Let $X := \{\langle z_i, y_i, q_i\rangle\}_{i=1}^n$ be a challenge for assumption #2 with parameters $[n, \gamma, d_x - 1, d_y - 1]$. First $\mathcal{A}'$ computes $q_i^* = z_i q_i + \alpha y_i$ for $i = 1, \ldots, n$, where α is randomly chosen from $\mathbb{F}$. Then, $\mathcal{A}'$ selects a random $M \in \{1, 2\}$.

1. If $M = 1$, $\mathcal{A}'$ selects random $z_i', y_i' \in \mathbb{F}$, and a $I \subseteq \{1, \ldots, n\}$ with $|I| = \gamma n$. Then, it selects a random $\alpha' \in \mathbb{F}$ and random $c_{j,j'} \in \mathbb{F}$ for $j = 1, \ldots, d_x$ and $j' = 0, \ldots, d_{y-1}$, and sets $q_i' = \alpha y_i + \sum_{j=1}^{d_x} \sum_{j'=0}^{d_y-1} c_{j,j'} (z_i)^j (y_i)^{j'}$ for all $i \in I$. Then, it selects q_i' for $i \notin I$ at random from $\mathbb{F}$. Finally it outputs the tuple

$$Y := \left\langle \{\langle z_i, y_i, q_i^* \rangle\}_{i=1}^n, \{\langle z_i', y_i', q_i' \rangle\}_{i=1}^n, \alpha\alpha' \right\rangle$$

2. If $M = 2$, $\mathcal{A}'$ selects random $z_i', y_i', q_i' \in \mathbb{F}$ and random $\rho \in \mathbb{F}$ and outputs the tuple

$$Y := \left\langle \{\langle z_i', y_i', q_i' \rangle\}_{i=1}^n, \{\langle z_i, y_i, q_i^* \rangle\}_{i=1}^n, \rho \right\rangle$$

Finally $\mathcal{A}'$ simulates $\mathcal{A}$ on Y. Now observe the following:

- If $X \in \mathsf{D}_2$ and $M = 1$ it holds that the tuple Y is uniformly distributed over D_3.
- If $X \in \mathsf{R}_2$ and $M = 2$ it holds that the tuple Y is uniformly distributed over R_3.

These two facts, together with the fact that $\mathcal{A}$ is a distinguisher of D_3 and R_3, suggest that

$$| \mathbf{Prob}_{X \in_U \mathsf{D}_2}[\mathcal{A}'(X) = 1 \mid M = 1] - \mathbf{Prob}_{X \in_U \mathsf{R}_2}[\mathcal{A}'(X) = 1 \mid M = 2] |$$

is non-negligible in n. On the other hand observe that in the two cases: (i) $X \in \mathsf{R}_2$ and $M = 1$, and (ii) $X \in \mathsf{D}_2$ and $M = 2$ the tuple Y generated by $\mathcal{A}'$ is identically distributed. This suggests that

$$\mathbf{Prob}_{X \in_U \mathsf{D}_2}[\mathcal{A}'(X) = 1 \mid M = 2] = \mathbf{Prob}_{X \in_U \mathsf{R}_2}[\mathcal{A}'(X) = 1 \mid M = 1]$$

Now observe that

$$\mathbf{Prob}_{X \in_U \mathsf{D}_2}[\mathcal{A}'(X) = 1] = \frac{1}{2} \sum_{v=1,2} \mathbf{Prob}_{X \in_U \mathsf{D}_2}[\mathcal{A}'(X) = 1 \mid M = v]$$

$$\mathbf{Prob}_{X\in_U \mathsf{R}_2}[\mathcal{A}'(X)=1]=\frac{1}{2}\sum_{v=1,2}\mathbf{Prob}_{X\in_U \mathsf{R}_2}[\mathcal{A}'(X)=1 \mid M=v]$$

(actually, to see this better - and to be more formal — the behavior of $\mathcal{A}'$ should be normalized so that it makes the same coin tosses no matter what is the random choice of M). It follows that

$$\mid \mathbf{Prob}_{X\in_U \mathsf{D}_2}[\mathcal{A}'(X)=1]-\mathbf{Prob}_{X\in_U \mathsf{R}_2}[\mathcal{A}'(X)=1] \mid$$

is non-negligible in n. This completes the proof. ■

Lemma 102 *Fix parameters k, t, γ, d_x, d_y (functions of n). Consider the two ensembles*

$$\mathsf{R}_4 :=$$

$$\left\{\left\langle \{\langle z_i, y_i\rangle\}_{i=1}^n, \{\langle z_i', y_i'\rangle\}_{i=1}^n, \{\langle z_i, y_i, q_i\rangle\}_{i=1}^n, \{\langle z_i', y_i', q_i'\rangle\}_{i=1}^n, \rho\right\rangle \right.$$

$$\left. \mid z_i, y_i, q_i, z_i', y_i', q_i', \rho \in \mathbb{F}\right\}$$

$$\mathsf{D}_4 :=$$

$$\left\{\left\langle \{\langle z_i, y_i\rangle\}_{i=1}^n, \{\langle z_i', y_i'\rangle\}_{i=1}^n, \right.\right.$$

$$\left.\mathsf{ACT}_{\gamma,d_x,d_y}(\{\langle z_i', y_i'\rangle\}_{i=1}^n, \alpha), \mathsf{ACT}_{\gamma,d_x,d_y}(\{\langle z_i, y_i\rangle\}_{i=1}^n, \alpha'), \alpha\alpha'\right\rangle$$

$$\mid \exists I, I', P, P', |I| = |I'| = t, \text{s.t. } P(0)=\alpha, P'(0)=\alpha', \forall i \in I :$$

$$\left. P(z_i)=y_i, \forall i \in I', P'(z_i')=y_i'\right\}$$

where the polynomials P, P' in the ensemble D_4 are of the form $P(x)=\sum_{j=0}^{k-1} c_j x^j$ and $P'(x)=\sum_{j=0}^{k-1} c_j' x^j$. Then, under assumption #1 with parameters $[n, k-1, t]$ and assumption #2 with parameters $[n, d_x-1, d_y-1, \gamma]$ the ensembles D_4 and R_4 are indistinguishable.

Proof. Let $\mathcal{A}$ be a distinguisher for the ensembles D_4 and R_4. We will assume that assumption #2 holds for parameters $[n, d_x - 1, d_y - 1, \gamma]$. and we will describe how to use $\mathcal{A}$ to construct a distinguisher $\mathcal{A}'$ for the ensembles D_1 and R_1 of assumption #1 for parameters $[n, k-1, t]$.

Let $X := \{\langle z_i, y_i \rangle\}_{i=1}^n$ be a challenge for assumption #1 with parameters $[n, k-1, t]$. First $\mathcal{A}'$ computes $y_i^* = z_i y_i + \alpha$ for $i = 1, \ldots, n$ where α is randomly selected from $\mathbb{F}$. Then $\mathcal{A}'$ selects at random $M \in_U \{1, 2\}$.

1. If $M = 1$. then $\mathcal{A}'$ selects z_i' at random from $\mathbb{F}$ for $i = 1, \ldots, n$: then it selects a random $I' \subseteq \{1, \ldots, n\}$ with $|I'| = t$ and a random polynomial $P' \in \mathbb{F}[x]$ of degree less than k. Then, it sets $y_i' = P(z_i)$ for all $i \in I$ and selects y_i' for $i \notin I'$ at random from $\mathbb{F}$. Denote $P'(0)$ by α'. Finally, $\mathcal{A}'$ outputs the tuple

$$Y := \Big\langle \{\langle z_i, y_i^* \rangle\}_{i=1}^n, \{\langle z_i', y_i' \rangle\}_{i=1}^n, \mathsf{ACT}_{\gamma, d_x, d_y}(\{\langle z_i', y_i' \rangle\}_{i=1}^n, \alpha),$$
$$\mathsf{ACT}_{\gamma, d_x, d_y}(\{\langle z_i, y_i^* \rangle\}_{i=1}^n, \alpha'), \alpha\alpha' \Big\rangle$$

2. If $M = 2$. then $\mathcal{A}'$ selects z_i', y_i' at random from $\mathbb{F}$. for $i = 1, \ldots, n$. It outputs the tuple

$$Y := \Big\langle \{\langle z_i', y_i' \rangle\}_{i=1}^n, \{\langle z_i, y_i^* \rangle\}_{i=1}^n, \mathsf{ACT}_{\gamma, d_x, d_y}(\{\langle z_i, y_i^* \rangle\}_{i=1}^n, \rho),$$
$$\mathsf{ACT}_{\gamma, d_x, d_y}(\{\langle z_i', y_i' \rangle\}_{i=1}^n, \alpha), \rho\alpha \Big\rangle$$

where ρ are selected at random from $\mathbb{F}$.

Now observe the following:

- If $X \in \mathsf{D}_1$ and $M = 1$, it follows that Y is uniformly distributed over D_4.
- If $X \in \mathsf{R}_1$ and $M = 2$, it follows that Y is uniformly distributed over (essentially) D_3. Under assumption #2 for parameters $[n, d_x - 1, d_y - 1, \gamma]$ it holds that D_3

is indistinguishable from R_3 (it follows from lemma 101). It follows finally that Y' is indistinguishably from uniform distributed over R_4.

These two facts, together with the fact that $\mathcal{A}$ is a distinguisher of D_4 and R_4, suggest that

$$|\ \mathbf{Prob}_{X\in_U \mathsf{D}_1}[\mathcal{A}'(X)=1 \mid M=1] - \mathbf{Prob}_{X\in_U \mathsf{R}_1}[\mathcal{A}'(X)=1 \mid M=2]\ |$$

is non-negligible in n. On the other hand consider the following two cases: (i) $X \in \mathsf{R}_1$ and $M=1$, and (ii) $X \in \mathsf{D}_1$ and $M=2$. In both of these cases it is easy to see that Y' is identically distributed in both cases. With a similar argument as in the proof of lemma 101, we conclude that

$$|\ \mathbf{Prob}_{X\in_U \mathsf{D}_1}[\mathcal{A}'(X)=1] - \mathbf{Prob}_{X\in_U \mathsf{R}_1}[\mathcal{A}'(X)=1]\ |$$

is non-negligible. ■

Theorem 103 *For parameters n,k,t,d_x,d_y,γ the key-exchange protocol is secure under: (a) assumption #1 for parameters $[n,k-1,t]$ (b) assumption #2 for parameters $[n,d_x-1,d_y-1,\gamma]$*

Proof. Just observe that protocol transcripts are distributed according to the ensemble D_4. From lemma 102 the proof follows immediately. ■

7.1.8 Feasibility of the Parameters

The parameters $\gamma, n, k, t, d_x, d_y, \log|\mathbb{F}|$ used in the protocol should satisfy various constraints for correctness and security to be satisfied. We recall them below:

1. $\delta := d_x + (k-1)(d_y-1) < \gamma^2 t$
2. $(\frac{t-\sqrt{t\delta}}{(1-\gamma)n} - \frac{t}{n})^2(1-\gamma)n = \omega(\log n)$

3. $\frac{\binom{t}{\sqrt{t\delta}}}{|\mathbb{F}|^{\sqrt{t\delta}-\delta-1}}$ should be negligible in n.

4. $t < \sqrt{(k-1)n}$

5. $\gamma < \sqrt[3]{\xi d_x d_y / n}$

If we assign

$$t := \delta n,\ k := \alpha\gamma^2 t + 1,\ d_x := 3\gamma^2 t,\ d_y := 2$$

where $\alpha, \beta, \gamma, \delta$ are constants less than 1, then we can to satisfy conditions 1,2,4,5 by setting e.g. $\alpha = 0.4$, $\beta = 0.5$, $\gamma = 0.96$ provided that ξ is greater equal to 5.3 (condition 3. can be satisfied easily by selecting an appropriately large finite field $\mathbb{F}$). Nevertheless this assignment is conditioned on the choice of parameter ξ and it could be well the case that BPR is solvable for such high choices of ξ. A very interesting research direction is to investigate the solvability of BPR and related problems so that the feasibility question regarding Polynomial Reconstruction based Key-exchange protocol is settled.

Bibliography

[AB00] Michel Abdalla and Mihir Bellare. Increasing the lifetime of a key: A comparative analysis of the security of re-keying techniques. In Tatsuaki Okamoto, editor, *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, volume 1976 of *Lecture Notes in Computer Science*, pages 546–559. Springer, 2000.

[AD97] Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing (STOC '97)*, pages 284–293, New York, USA, May 1997. ACM Press.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems. In *Proceedings of The Twenty-Eighth Annual ACM Symposium On The Theory Of Computing (STOC '96)*, pages 99–108, New York, USA, May 1996. ACM Press.

[ARS78] L. Adleman, R. L. Rivest, and A. Shamir. A method for obtaining digital signature and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

[Bar98] Alexander Barg. Complexity issues in coding theory. In Vera Pless, W. C. Huffman, and Richard A. Brualdi, editors, *Handbook of Coding Theory*, New York, NY, USA, 1998. North Holland.

[BDJR97] Mihir Bellare, Anand Desai, E. Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption: Analysis of the DES modes of operation. In *Proceedings of the 38th Symposium on Foundations of Computer Science (FOCS)*, pages 394–403. IEEE Computer Society Press, 1997.

[Ber68] E. R. Berlekamp. *Algebraic Coding Theory.* McGraw-Hill, New York, 1968.

[BG85] M. Blum and S. Goldwasser. An efficient probabilistic public-key encryption scheme which hides all partial information. In George Robert Blakely and David Chaum, editors, *Advances in Cryptology: proceedings of CRYPTO 84*, volume 196 of *Lecture Notes in Computer Science*, pages 289–302, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 1985. Springer-Verlag.

[BLR90] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. In *Proceedings of the twenty-second annual ACM Symposium on Theory of Computing, Baltimore, Maryland, May 14–16, 1990*, pages 73–83, New York, NY, USA, 1990. ACM Press.

[BM84] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal on Computing*, 13(4):850–864, November 1984.

[BN00] Daniel Bleichenbacher and Phong Q. Nguyen. Noisy polynomial interpolation and noisy chinese remaindering. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT ' 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 53–69, Brugge, Belgium, 2000. Springer-Verlag, Berlin Germany.

[BW86] E. R. Berlekamp and L. Welch. Error correction of algebraic block codes. US Patent Number 4,633,470, 1986.

[Chv79] Vašek Chvátal. The tail of the hypergeometric distribution. *Discrete Math*, 25:285–287, 1979.

[Cop02] Don Coppersmith. personal communication, 2002.

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, November 1976.

[FF93] Joan Feigenbaum and Lance Fortnow. Random-self-reducibility of complete sets. *SIAM Journal on Computing*, 22(5):994–1005, October 1993.

[FFLS94] J. Feigenbaum, L. Fortnow, C. Lund, and D. Spielman. The power of adaptiveness and additional queries in random-self-reductions. *Computational Complexity*, 4(2):158–174, 1994. Extended abstract in Proceedings of Structures '92.

[FKN90] Joan Feigenbaum, Sampath Kannan, and Noam Nisan. Lower bounds on random-self-reducibility (extended abstract). In *Proceedings, Fifth Annual Structure in Complexity Theory Conference*, pages 100–109, Barcelona, Spain, 8–11 July 1990. IEEE Computer Society Press.

[For66] G. David Forney, Jr. *Concatenated Codes*. M.I.T. Press, Cambridge, MA, USA, 1966.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In *Proceedings of the 19th Annual Symposium on Theory of Computing (STOC)*, pages 218–229, New York, NY USA, May 1987. ACM Press.

[Gol90] Oded Goldreich. A note on computational indistinguishability. *Information Processing Letters*, 34(6):277–281, May 1990.

[Gol93] Oded Goldreich. A uniform-complexity treatment of encryption and zero-knowledge. *Journal of Cryptology*, 6(1):21–53, 1993.

[Gol97] Shafi Goldwasser. Fault tolerant multi party computations: Past and present. In *Proceedings of the 16th Annual ACM Symposium on Principles of Distributed Computing (PODC '97)*, pages 1–6, New York, August 1997. ACM Press.

[Gol98] Oded Goldreich. Foundations of cryptography (fragments of a book). Manuscript, February 1998. Version 2.03, available from `http://www.wisdom.weizmann.ac.il/ oded/frag.html`.

[GRS95] Oded Goldreich, Ronitt Rubinfeld, and Madu Sudan. Learning polynomials with queries: The highly noisy case. In *36th Annual Symposium on Foundations of Computer Science: October 23–25, 1995, Milwaukee, Wisconsin*, pages 294–303. IEEE Computer Society Press, 1995.

[GS98] Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. In *39th Annual Symposium on*

Foundations of Computer Science: proceedings: November 8–11, 1998, Palo Alto, California, pages 28–37. IEEE Computer Society Press, 1998.

[Gur01] Venkatesan Guruswami. *List Decoding of Error-Correcting Codes*. PhD thesis, MIT, 2001.

[HILL99] Johan Håstad, Russel Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[IM85] Hideki Imai and Tsutomu Matsumoto. Algebraic methods for constructing asymmetric cryptosystems. In Jacques Calmet, editor, *Proceedings of the 3rd International Conference on Algebraic Algorithms and Error-Correcting Codes (AAECC-3)*, volume 229 of *LNCS*, pages 108–119, Grenoble, France, July 1985. Springer.

[Kil90] Joe Kilian. *Uses of Randomness in Algorithms and Protocols*. MIT Press, 1990.

[Kra93] Hugo Krawczyk. Distributed fingerprints and secure information dispersal. In *Proceedings of the twelveth ACM Symposium on Principles of Distributed Computing (PODC '93), Ithaca, New York, USA*, pages 207–218, New York, USA, August 1993. ACM Press.

[KY00] Jonathan Katz and Moti Yung. Complete characterization of security notions for probabilistic private-key encryption. In *Proceedings of the thirty second annual ACM Symposium on Theory of Computing: Portland, Oregon, May 21–23, [2000]*, pages 245–254, New York, NY, USA, 2000. ACM Press.

[KY01a] Jonathan Katz and Moti Yung. Unforgeable encryption and chosen ciphertext secure modes of operation. In Bruce Schneier, editor, *Fast Software Encryption, 7th International Workshop, FSE 2000, New York, NY, USA, April 10-12, 2000, Proceedings*, volume 1978 of *Lecture Notes in Computer Science*, pages 284–299. Springer, 2001.

[KY01b] Aggelos Kiayias and Moti Yung. Polynomial reconstruction based cryptography. In Serge Vaudenay and Amr M. Youssef, editors, *Selected Areas in Cryptography, 8th Annual International Workshop, SAC 2001 Toronto, Ontario, Canada, August 16-17, 2001, Revised Papers*, volume 2259 of *Lecture Notes in Computer Science*, pages 129–133. Springer, 2001.

[KY01c] Aggelos Kiayias and Moti Yung. Secure games with polynomial expressions. In Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen, editors, *Automata, Languages and Programming, 28th International Colloquium, ICALP 2001, Crete, Greece, July 8-12, 2001, Proceedings*, volume 2076 of *Lecture Notes in Computer Science*, pages 939–950. Springer, 2001.

[KY02] Aggelos Kiayias and Moti Yung. Cryptographic hardness based on the decoding of reed-solomon codes. In Peter Widmayer, Francisco Triguero Ruiz, Rafael Morales, Matthew Hennessy, Stephan Eidenbenz, and Ricardo Conejo, editors, *Automata, Languages and Programming, 29th International Colloquium, ICALP 2002, Malaga, Spain, July 8-13, 2002, Proceedings*, volume 2380 of *Lecture Notes in Computer Science*, pages 232–243. Springer, 2002.

[Lip91] Richard J. Lipton. New directions in testing. In *Distributed Computing and Cryptography*, volume 2 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 191–202. American Mathematics Society, 1991.

[Lub96] Michael George Luby. *Pseudorandomness and cryptographic applications*. Princeton computer science notes. Princeton University Press, Princeton, NJ, USA, 1996.

[McE78] R. J. McEliece. *A Public-Key System Based on Algebraic Coding Theory*, pages 114–116. Jet Propulsion Lab, 1978. DSN Progress Report 44.

[Mer78] Ralph C. Merkle. Secure communications over insecure channels. *Communications of the Association for Computing Machinery*, 21(4):294–299, April 1978.

[MRLW02] Fabian Monrose, Michael K. Reiter, Qi Li, and Susanne Wetzel. Password hardening based on keystroke dynamics. *International Journal of Information Security*, 1(2):69–83, 2002.

[MS77] F. J. MacWilliams and N. Sloane. *The Theory of Error Correcting Codes*. North Holland, Amsterdam, 1977.

[MVV97] Alfred J. Menezes, Paul C. Van Oorschot, and Scott A. Vanstone. *Handbook of applied cryptography*. The CRC Press series on discrete mathematics and its applications. CRC Press, 1997.

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, 1991.

[Nat99] National Institute of Standards and Technology (NIST). Data Encryption Standard (DES). Federal Information Processing Standards Publication 46-3 (FIPS PUB 46-3), October 1999.

[Nat01] National Institute of Standards and Technology (NIST). Advanced Encryption Standard (AES). Federal Information Processing Standards Publication 197 (FIPS PUB 197), November 2001.

[NP99] Moni Naor and Benny Pinkas. Oblivious transfer and polynomial evaluation. In *Proceedings of the thirty-first annual ACM Symposium on Theory of Computing: Atlanta, Georgia, May 1–4, 1999*, pages 245–254, New York, NY, USA, 1999. ACM Press.

[NR97] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *Proceedings of the 38th Symposium on Foundations of Computer Science (FOCS)*, pages 458–467. IEEE Computer Society Press, 1997.

[Pap94] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, New York, 1994.

[Ped91] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *Advances in cryptology — CRYPTO '91: proceedings*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 1991. Springer-Verlag.

[Rab79] Michael Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, Laboratory for Computer Science, Massachusetts Institute of Technology, January 1979.

[Rab89] Michael O. Rabin. Efficient dispersal of information for security, load balancing, and fault tolerance. *JACM*, 36(2):335–348, April 1989.

[Sch80] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *JACM*, 27(4):701–717, October 1980.

[Sip97] Michael Sipser. *Introduction to the Theory of Computation*. PWS Publishing Co., Boston, Massachusetts, 1997.

[Sud97] Madhu Sudan. Decoding of Reed Solomon codes beyond the error-correction bound. *Journal of Complexity*, 13(1):180–193, March 1997.

[VV89] S. A. Vanstone and P. C. VanOorschot. *An Introduction to Error Correcting Codes with Applications*. Kluwer Academic Publishers, 1989.

[Yao86] Andrew C. Yao. How to generate and exchange secrets. In *Proceedings of the 27th Symposium on Foundations of Computer Science (FOCS)*, pages 162–167. IEEE Computer Society Press, 1986.

[Zac82] Stathis Zachos. Robustness of probabilistic computational complexity classes under definitional perturbations. *Information and Control*, 54(3):143–154, September 1982.