DISSERTATIONES MATHEMATICAE UNIVERSITATIS TARTUENSIS

67

DISSERTATIONES MATHEMATICAE UNIVERSITATIS TARTUENSIS
67

MARGUS NIITSOO

Black-box Oracle Separation Techniques with Applications in Time-stamping

TARTU UNIVERSITY PRESS

Institute of Computer Science, Faculty of Mathematics and Computer Science, University of Tartu, Estonia

Dissertation accepted for public defense of the degree of Doctor of Philosophy (PhD) on March 28, 2011 by the Council of the Institute of Computer Science, University of Tartu.

Supervisor:

prof.	Ahto Buldas University of Tartu Tartu, Estonia

Opponents:

prof.	Helger Lipmaa Tallinn University Tallinn, Estonia
dr.	Berry Schoenmakers Eindhoven University of Technology Eindhoven, Netherlands

The public defense will take place on May 20, 2011 at 18:00 in Liivi 2-403.

The publication of this dissertation was financed by Institute of Computer Science, University of Tartu.

ISSN 1024-4212
ISBN 978-9949-19-624-1(trükis)
ISBN 978-9949-19-625-8(PDF)

Tartu Ülikooli Kirjastus
http://www.tyk.ee
Tellimus nr. 208

Contents

LIST OF ORIGINAL PUBLICATIONS

1. Buldas, A., Niitsoo, M.: Can we construct unbounded time-stamping schemes from collision-free hash functions? In: The 2rd International Conference on Provable Security (ProvSec) 2008. LNCS, vol. 4784, pp. 254–267. Springer (2008).

2. Buldas, A., Laur, S., Niitsoo, M.: Oracle separation in the non-uniform model. In: The 3rd International Conference on Provable Security (ProvSec) 2009. LNCS, vol. 5848, pp. 230–244. Springer (2009).

3. Buldas, A., Jürgenson, A., Niitsoo, M.: Efficiency bounds for adversary constructions in black-box reductions. In: Australian Conference on Information Security and Privacy – ACISP 2009. LNCS, vol. 5594, pp. 264–275. Springer (2009).

4. Buldas, A., Niitsoo, M.: Optimally tight security proofs for hash-then-publish time-stamping. In: Australian Conference on Information Security and Privacy – ACISP 2010. LNCS, vol. 6168, pp. 318–355. Springer (2010).

UNPUBLISHED WORK INCLUDED IN THE THESIS

5. Niitsoo, M.: Deterministic random oracles (2011), unpublished.

6. Buldas, A., Niitsoo, M.: Black-box separations and their adaptability to the non-uniform model (2011), unpublished.

PUBLICATIONS NOT INCLUDED IN THE THESIS

7. Niitsoo, M.: Optimal adversary behavior for the serial model of financial attack trees. In: International Workshop on Security – IWSEC 2010. LNCS, vol. 6434, pp. 354–370. Springer (2010)

ABSTRACT

In cryptology, most of the security proofs of systems are not unconditional but rather rely on the security properties of the underlying primitives. In fact, most of the proofs consider the primitives they use as black boxes, assuming only that they work securely in the way specified and nothing else. Black-box proofs have their limitations, however, and there are in fact many cases in which such an approach can be ruled out by a proof method known as Oracle Separation.

This work is mainly concerned with exploring the Oracle Separation paradigm. Firstly, we show how it can be weakened to rule out "generic" constructions from hash functions to time-stamping schemes. Secondly, we describe how changing a few standard proof steps allows the results to be translated into the non-uniform computational model as well. Thirdly, we demonstrate a novel way of upper-bounding the efficiency of security proofs of black-box constructions and then use this approach to prove the optimality of a reduction from collision-resistant hash functions to secure bounded time-stamping schemes. Finally, we also explore the possibility of using a fixed, "algorithmically random" oracle instead of the standard random oracle, which has great potential in simplifying some more technical aspects of separation result proofs.

INTRODUCTION

Cryptology is the science of secure communications. In modern times, however, it has grown to encompass all sorts of different problems related not only to the issue of privacy but also to those of anonymity and authenticity. The topics that are worked on vary greatly, ranging from the classical disciplines of making and breaking new ciphers to the problems of secure protocol design and questions about secure multi-party computation.

One characteristic of modern-day cryptology is its reliance on computational assumptions. As unconditional security is generally very hard to achieve, it often makes sense to try to reduce the security of a system to a computational hardness assumption of a well-studied mathematical problem. This is usually done by showing that if a certain system were to be insecure, it would automatically imply an existence of an efficient algorithm for solving the underlying mathematical problem. This approach works very well for lower-level systems or primitives, whose construction is often based on hard problems in number theory. One of the best known examples of this is probably the Diffie-Hellman key exchange protocol, whose security is proved based on the hardness of finding discrete logarithms.

For more complex protocols, a similar principle is used but in a somewhat different way. Instead of proving security based directly on computational assumptions, these proofs take a more abstract approach and base the proof on the security of some other, simpler, cryptographic system. In these cases, what is shown is that if the new system can be broken, it would automatically imply that the old system is also insecure. There are many famous results of this type, for instance the construction by Merkle [33] and Damgård [13].

Most of these reductions work for any implementation of the underlying cryptographic system. This is so because both the construction and proof only use the underlying system in a so-called black box way where no additional assumptions are made about it besides the fact that it securely does exactly what it is meant to do. This is very convenient, as it allows one to instantiate the system with the best known implementation of an underlying functionality and to switch it out with a newer and better version when the old one becomes either obsolete or insecure.

Such a black-box approach has inherent limitations, however. This was first shown by Impagliazzo and Rudich [27] who proved that secret key exchange (which is a prerequisite for all public-key cryptography) cannot be built in a purely

black-box way by only assuming the existence of one-way permutations (which are enough to do nearly all of secret-key cryptology). This showed that the distinction between public-key and secret-key primitives is an inherent one and that it is possible in theory for the secret-key primitives to exist even when none of the public-key primitives do.

The result was proven by a method called oracle separation, which was introduced by Baker, Gill and Solovay [1] in the context of complexity theory. The main idea of the method is to embed a very powerful computational entity (an oracle) into the underlying primitive and then show how this additional power can be abused to break any implementation of the more complex system while still leaving the underlying primitive itself secure. Such a proof basically implies that although a secure instance of the underlying primitive does exist, it cannot be used to construct an instance of the more complex system as no such secure complex system can exist while this powerful implementation of the underlying primitive is around.

The seminal result was soon followed by a long line of impossibility results [39, 29, 18, 19, 20, 12, 16, 25, 17, 3] which extended the simple approach taken by Impagliazzo and Rudich. For instance, Gennaro and Trevisan [18] showed that separation results can also be proven in the non-uniform model, while Hsiao and Reyzin [25] noticed that considerably weaker separation results can still be used to rule out the existence of fully black-box constructions in certain cases. The approach of proving impossibility results was also extended by Kim, Simon and Tetali [29] to show lower bounds in terms of construction efficiency.

Most of the authors closely follow the proof model of one of the aforementioned papers. However, there are also numerous exceptions where authors depart from these models and often achieve even stronger results. This, however, has resulted in a plurality of models which are usually just slightly different from each other. Reingold, Trevisan and Vadhan [37] attempted to systematize the varying approaches used but considered only one of the possible ways in which the separation results may differ, essentially describing different degrees of black-boxness.

The author's interest in the general methods of oracle separation grew out of studying their application to the concrete problem of time-stamping documents by the scheme proposed by Haber and Stornetta [22]. The scheme is constructed using hash functions and in such a way that collision-resistance may not be sufficient to guarantee the security of the scheme as demonstrated by Buldas et al. [10]. The sufficient criterion, chain-resistance, is however fairly hard to verify directly, which immediately brings up the question whether it can be reduced to more standard assumptions.

In this work, we describe two new additional ways of using oracle separation. Firstly, we describe a very weak separation for tree-based time-stamping schemes which only rules out reductions in a generic model where the adversary is given a 'reasonably limited' access to the oracle. Secondly, we propose a new framework for proving lower bounds on the efficiency of the security reduction, which dif-

fers markedly from the previous approaches that show lower bounds only on the efficiency of the construction of the new primitive. To demonstrate that this new framework can provide interesting and non-trivial results, we again apply it to the problem of time-stamping and show that any black-box reduction from tree-based time-stamping to collision-resistance has an inherent security loss of power 1.5. It turns out that this bound can be reached by a precise combinatorial analysis, allowing us to show the first known reduction provably optimal in terms of security loss.

We then turn to broader questions and try to generalize the previously known reductions to also work in the non-uniform model of computation. It turns out that this is possible for most of the results known, but only to a certain degree. Finally, we try to overcome some of the technical problems associated with random oracles by exploring the limits of using algorithmic randomness instead of classical randomness to produce oracles that seem comparably powerful.

This work is based on four published and two unpublished papers from the period of 2008 to 2011.

1. Buldas, A., Niitsoo, M.: Can we construct unbounded time-stamping schemes from collision-free hash functions? In: The 2rd International Conference on Provable Security (ProvSec) 2008. LNCS, vol. 4784, pp. 254–267. Springer (2008).

   The paper is fully author's own work, with only the problem statement and background information provided by the supervisor.

2. Buldas, A., Laur, S., Niitsoo, M.: Oracle separation in the non-uniform model. In: The 3rd International Conference on Provable Security (ProvSec) 2009. LNCS, vol. 5848, pp. 230–244. Springer (2009).

   The author's main contribution was the proofs of all the theorems, whereas the theorem statements were provided by the other authors.

3. Buldas, A., Jürgenson, A., Niitsoo, M.: Efficiency bounds for adversary constructions in black-box reductions. In: Australian Conference on Information Security and Privacy – ACISP 2009. LNCS, vol. 5594, pp. 264–275. Springer (2009).

   The author came up with the proofs for the framework theorems and the exact preconditions required to actually be able to prove useful lower bounds. He also came up with the details of the proof for the lower bound on division-resistance reduction.

4. Buldas, A., Niitsoo, M.: Optimally tight security proofs for hash-then-publish time-stamping. In: Australian Conference on Information Security and Privacy – ACISP 2010. LNCS, vol. 6168, pp. 318–355. Springer (2010).

   The authors main contribution was a large part of the proof of the tightest security bound which allows for a very precise security analysis.

5. Buldas, A., Niitsoo, M.: Black-box separations and their adaptability to the non-uniform model (2011), unpublished.

   The author is responsible for generalizing the results of Unruh [40] and Simon [39].

6. Niitsoo, M.: Deterministic random oracles (2011), unpublished.

   The author is the only author of this paper and as such, both the idea and the proofs are his.

The copies of papers I–VI are included at the end of the thesis on pages 65 – 168.

The outline of the thesis is the following. Chapter 1 describes the notion of a Black-box reduction and investigates various definitions of different strengths. Chapter 2 concentrates on the oracle separation method and describes the state of the art in the field. Chapter 3 gives a brief overview of hash tree based time-stamping schemes to which some of the results of this thesis apply.

The rest of the Chapters are concerned with expounding the author's own work. Chapter 4 describes how evidence for the non-existence of a reduction can still be achieved by weakening the separation model and allowing only "generic" access to the oracle. Chapter 5 explains how oracle separation results proven in the uniform model can be translated to the non-uniform model and shows the limitations of such an approach. Chapter 6 discusses how meaningful upper bounds on the efficiency of the security proofs can be obtained and shows how to use them to prove the optimality of a reduction from collision-resistance to bounded time-stamping. Chapter 7 takes a more abstract approach and explores the possibility of using algorithmic randomness instead of "true" randomness in the oracles and shows that this approach indeed has promise.

PRELIMINARIES AND TERMINOLOGY

We look at bits as binary digits. As such, $\{0,1\}^*$ is defined to be the set of all finite bit strings while $\{0,1\}^\omega$ is defined as the set of all infinite bit sequences. For bit-strings a and b we define $a\|b$ as their concatenation.

We will use $\cdot$ as a placeholder for function arguments. For example $f = g(\cdot, a, \cdot)$ should be taken to mean that f is a two-argument function defined so that $f(x,y) = g(x,a,y)$. Furthermore, we will assume a standard isomorphism (realized by $\|$) between $\{0,1\}^m \times \{0,1\}^n$ and $\{0,1\}^{m+n}$ and use that to turn multi-parameter functions into single-parameter functions and vice versa as needed.

By $x \leftarrow \mathcal{D}$ we mean that x is chosen randomly according to a distribution $\mathcal{D}$. If A is a probabilistic function or a Turing machine, then $x \leftarrow \mathsf{A}(y)$ means that x is chosen according to the output distribution of A on an input y. If $\mathcal{D}_1, \ldots, \mathcal{D}_m$ are distributions and $F(x_1, \ldots, x_m)$ is a predicate, then $\Pr[x_1 \leftarrow \mathcal{D}_1, \ldots, x_m \leftarrow \mathcal{D}_m : F(x_1, \ldots, x_m)]$ denotes the probability that $F(x_1, \ldots, x_m)$ is true.

We use the Landau notation for describing asymptotic properties of functions. For functions $f, g\colon \mathbb{N} \to \mathbb{R}^+$, we write $f(k) = O(g(k))$ [$f(k) = \Omega(g(k))$] if there is $c \in \mathbb{R}$, so that $f(k) \le cg(k)$ [$f(k) \ge cg(k)$] for sufficiently large k. We write $f(k) = \Theta(g(k)$ if $f(k) = O(g(k))$ and $f(k) = \Omega(g(k))$. We write $f(k) = \omega(g(k))$ if $\lim\limits_{k\to\infty} \frac{g(k)}{f(k)} = 0$ and $f(k) = o(g(k))$ if $g(k) = \omega(f(k))$. In particular, $f(k) = O(1)$ means that f is bounded and $f(k) = k^{-\omega(1)}$ means that $f(k)$ decreases faster than any polynomial, i.e., f is *negligible*. We say that something happens with *overwhelming* probability if the probability of it not happening is negligible. A Turing machine (TM) M is *poly-time* (PTM) if it runs in time $k^{O(1)}$, where k denotes the the *security parameter*.

Unless explicitly stated otherwise, we assume all Turing machines to be probabilistic, i.e., to have access to an additional tape that is filled with an infinite amount of independently and uniformly chosen and random bits.

By an *oracle Turing machine* (OTM) we mean an incompletely specified Turing machine S that can make function calls to an independently specified *oracle* function $\mathcal{O} : \{0,1\}^* \to \{0,1\}^*$. In this case, the machine is denoted by $\mathsf{S}^{\mathcal{O}}$. The oracle function $y \leftarrow \mathcal{O}(x)$ does not have to be computable but it does have a conditional running time $t(x)$, which does not necessarily reflect the actual amount of computations needed to produce y from x. The running time of $\mathsf{S}^{\mathcal{O}}$ comprises the conditional running time of oracle calls – each call $\mathcal{O}(x)$ takes $t(x)$ steps.

Note that though the classical complexity-theoretic oracles only require a single step, this more general notion is appropriate in cryptography where oracles often model abstract adversaries with running time t. We say that S is a *poly-time oracle machine* (POTM) if $\mathsf{S}^{\mathcal{O}}$ runs in poly-time, whenever $\mathcal{O}$ is poly-time. By a *non-uniform* poly-time oracle machine we mean an ordinary poly-time oracle machine S together with a family $\mathcal{A} = \{a_k\}_{k\in\mathbb{N}}$ of (advice) bit-strings a_k with length $k^{O(1)}$. For any oracle $\mathcal{O}$ and any input x, it is assumed that $\mathsf{S}^{\mathcal{O}}(x)$ has access to the advice string $a_{|x|}$. Usually, the advice strings are omitted for simplicity, but their presence must always be assumed when S is non-uniform. One of the most important facts about non-uniform poly-time Turing machines is that there is an uncountable number of them, whereas there are only countably many ordinary Turing machines. We will assume Turing machines to be uniform unless it is explicitly stated otherwise.

A *random oracle* $\mathcal{O}$ is defined as a function chosen uniformly at random from the set of all functions $\mathcal{O} : \{0,1\}^* \to \{0,1\}$ (which is seen to be isomorphic to the set of $\{0,1\}^\omega$). A functionality is said to exist under a random oracle model if it exists for measure 1 of all the oracles.

CHAPTER 1

BLACK-BOX REDUCTIONS

1.1 Black-box Reductions

Black-box reductions play a central role in the modern cryptography. We now describe a simplified form of a very famous reduction (due to Merkle [33] and Damgård [13]) that is meant to illustrate the concept and to help give insight into the formal definitions which are introduced later.

1.1.1 Merkle-Damgård Construction

Modern cryptography uses many different primitives to construct new protocols and schemes. One of the most often used of these is the notion of a hash function, which are employed to condense an arbitrarily long bit string into a short "digest". The digest can then be used in place of the longer bit string for many purposes. There are numerous different notions of security for such hash functions. One of the more common of them is that of "collision-resistance":

Definition 1. We say that a family χ of functions $h : \{0,1\}^* \to \{0,1\}^n$ is (t, ϵ)-*collision resistant* if for any t-time non-uniform adversary A we have

$$\Pr[h \leftarrow \chi, (m_0, m_1) \leftarrow \mathsf{A}(h) \colon m_0 \neq m_1, h(m_0) = h(m_1)] < \epsilon \ . \qquad (1.1)$$

Intuitively, collision resistance implies that for a h chosen randomly from χ, it is hard to find two inputs that map to the same output. This can be used to ensure that once someone computes a "digest" for a bitstring with a given h, it would be hard for him to claim to have actually used some other input since it is hard to find a "digest" that corresponds to two distinct inputs simultaneously.

It is worth noting that no single fixed hash function $h : \{0,1\}^* \to \{0,1\}^n$ could ever be collision resistant – since there are far more inputs than there are outputs, a collision is guaranteed to exist and therefore, an adversary that just has that collision hardwired into it, would trivially break the collision-resistance property. Using a family of hash functions loosely corresponds to the intuition

that a collision has to be computed, as for a large enough family, the adversary cannot have all the collisions hardwired into it any more so it would actually have to compute them somehow.

The construction given in [13, 33] was originally proposed as a way of constructing secure collision-resistant hash functions $h : \{0,1\}^* \to \{0,1\}^m$ for inputs of arbitrary length based on a collision-resistant compressing function $h : \{0,1\}^n \to \{0,1\}^m$ with a fixed-length input. To avoid some technicalities, we will give a simpler proof just showing how a compressing function $h : \{0,1\}^k \to \{0,1\}^m$ with arbitrarily large input length k can be constructed based on a compressing function $f : \{0,1\}^{n+m} \to \{0,1\}^m$ with a shorter fixed input length $n + m < k$.

Theorem 1. *Assume that for fixed $n, m \in \mathbb{N}$ there exists a family $\mathcal{F}_s$ of (t,ϵ)-collision-resistant functions $f : \{0,1\}^{n+m} \to \{0,1\}^m$. Then for every $k \in \mathbb{N}$ there also exist a family $\mathcal{F}_e$ of compressing functions $h\colon \{0,1\}^k \to \{0,1\}^m$ that is ($t + 2lt'$,ϵ)-collision-resistant (where t' is the upper bound on the time it takes to compute $f \in \mathcal{F}_s$).*

Proof. We start by constructing the family $\mathcal{F}_e$. Let $f : \{0,1\}^{n+m} \to \{0,1\}^m$ be any function from $\mathcal{F}_s$ and let $r \in \{0,1\}^m$ be a randomly chosen seed. We now define a function $h_{f,r}\colon \{0,1\}^k \to \{0,1\}^m$ by showing how it works on a fixed input $x \in \{0,1\}^k$. The family $\mathcal{F}_e$ can then be defined as the set of all such functions $h_{f,r}$ where $f \in \mathcal{F}_s$ and $r \in \{0,1\}^m$.

If n does not divide k, we begin by adding zeroes to the end of x until its length is a multiple of n. We then break x into blocks of n bits so $x = x_1 \| x_2 \| \cdots \| x_l$ for $l = \lceil \frac{k}{n} \rceil$. After that, we construct $y_1, \ldots, y_l \in \{0,1\}^m$ by specifying $y_1 = f(r \| x_1)$ and $y_i = f(y_{i-1} \| x_i)$ for $i = 2, \ldots, l$. The value y_l is then returned as the output of $h_{f,r}(x)$.

The preceding description of computation can easily be formalized as an algorithm. It should also be easy to see that if f can be computed in t steps then h can be computed in roughly $lt + n$ steps so it remains relatively efficient. It is also crucial to note that we use f in a black-box manner – we do not know how it works, only that it does. We give it input and it gives us output, but how it computes the output does not concern us.

We now need to show that if $\mathcal{F}_s$ is collision-resistant then so is $\mathcal{F}_e$. Assume the opposite, eg. that $\mathcal{F}_s$ is indeed collision-resistant but that $\mathcal{F}_e$ is not. There then exists an adversary A_l that can break the collision-resistance property for the functions $h \in \mathcal{F}_e$ with more than a negligible probability. Assume A_l can find a collision pair (a, b) for $h_{f,r} \in \mathcal{F}_e$. Let $a = a_1 \| a_2 \| \cdots \| a_l$ and $b = b_1 \| b_2 \| \cdots \| b_l$ where both a_i and b_i are all blocks of length n bits where a and b are padded with zeroes if needed. Then the computation of $h(a)$ yields a sequence $y_1^a, \ldots, y_l^a$ and the computation of $h(b)$ gives $y_1^b, \ldots, y_l^b$. Since (a, b) is a collision, we have $y_l^a = h(a) = h(b) = y_l^b$. This implies that $f(y_{l-1}^a \| a_l) = f(y_{l-1}^b \| b_l)$. If $y_{l-1}^a \| a_l \neq y_{l-1}^b \| b_l$, this results in a collision for f. If not, let r be the smallest such value that

$y^a_{r+1} = y^b_{r+1}$ but $y^a_r \| a_{r+1} \neq y^b_r \| b_{r+1}$. This value has to exist because (a, b) is a collision so $a \neq b$ which implies that a_r and b_r differ at some point r. It is also clear that $(y^a_r \| a_{r+1}, y^b_r \| b_{r+1})$ is then a collision for f.

Therefore, we can create an adversary A_s to break $f \in \mathcal{F}_s$ by having it choose a random seed r, using A_l to find a collision for $h_{f,r}$ and using that to find a collision for f in the way described above. It is easy to see that if A_l succeeds with probability ϵ then so does the new adversary A_s, since each successfully found collision for $h_{f,r}$ is translated to a collision for f. The running time of A_s is equal to A_l plus roughly the time it takes to run h up to $2l$ times. This means that if A_l is efficient, then so is A_s. The existence of such an efficient A_s is a contradiction, since we assumed $\mathcal{F}_s$ to be collision resistant. □

1.1.2 Breakdown of the Proof

The previous proof is a fairly typical example of a proof of security in modern cryptology. We begin with two objects: a function or a family of functions $\mathcal{Q}$ that we assume we already have ($\mathcal{F}_s$ in the example) and a (family of) functions $\mathcal{P}$ that we would like to construct ($\mathcal{F}_e$). In abstract, the proof can be seen as consisting of the following two steps:

1. Show how to construct an instance p^q of $\mathcal{P}$ based on an instance q of $\mathcal{Q}$.

2. Show that if there exists an adversary A_p against $p^q \in \mathcal{P}$ then it can be converted into an adversary A_q against the q that was used in the construction of p^q.

A large proportion of the security proofs in cryptology can be seen to follow the exact same pattern.

To formalize the preceding abstract view, we will introduce the following definition of a primitive given by Reingold et al. [37].

Definition 2. A primitive $\mathcal{P}$ is a pair $(\mathcal{P}_F, \mathcal{P}_R)$, where $\mathcal{P}_F$ is a family of functions $f : \{0,1\}^* \to \{0,1\}^*$, and $\mathcal{P}_R$ is a relation over pairs (f, M) of a function $f \in \mathcal{P}_F$ and a Turing machine M . The set $\mathcal{P}_F$ is required to contain at least one function which is computable by a PTM.

A function $f : \{0,1\}^* \to \{0,1\}^*$ implements $\mathcal{P}$ or is an implementation of $\mathcal{P}$ if $f \in \mathcal{P}_F$. An efficient implementation of $\mathcal{P}$ is an implementation of $\mathcal{P}$ which is computable by a PTM. A machine A $\mathcal{P}$-breaks $f \in \mathcal{P}_F$ if $(f, \mathsf{A}) \in \mathcal{P}_R$. A secure implementation of $\mathcal{P}$ is an implementation of $\mathcal{P}$ such that no PTM $\mathcal{P}$-breaks f. The primitive $\mathcal{P}$ exists if there exists an efficient and secure implementation of $\mathcal{P}$. The primitive $\mathcal{P}$ exists relative to a given oracle $\mathcal{O}$ if there exists a secure implementation f of $\mathcal{P}$ where f is computable by a POTM given access to $\mathcal{O}$ and it is secure against adversaries who also have access to $\mathcal{O}$.

Essentially, $\mathcal{P}_F$ specifies a set of functions that fill the syntactic criteria for the given primitive (such as the set of all compressing functions for the example in the previous section). The set $\mathcal{P}_R$, on the other hand, gives information about whether the primitive is secure by showing which Turing machines break it and which do not.

A few things in the preceding definition may seem counter-intuitive at first. For instance, the notion of a primitive should also capture cryptographic objects that have many parts. As an example, a symmetric cryptosystem consists of an algorithm for key generation, an algorithm for encryption and an algorithm for decryption which are three distinct functions. However, it is easy to see that this can be modeled with just one function, if the first two bits of input specify which of the three functionalities is currently to be used. In such a case, $f(00|r)$ could stand for the call to the key generator $g(r)$, while $f(01|k|x)$ and $f(10|k|y)$ could stand for encryption $e(k, x)$ and decryption $d(k, y)$ respectively.

A similar argument can be used when the security definition is worded in terms of function families – assuming that they are finite, one can just use the first few bits of randomness to select one fixed function from the function family. For instance, if we are talking about a family of 2^k compressing functions of type $f : \{0,1\}^{m+n} \to \{0,1\}^m$, it can be modeled as a single function $f' : \{0,1\}^{k+m+n} \to \{0,1\}^m$ so that $f'(r\|\cdot)$ defines a unique member of that family for every $r \in \{0,1\}^k$.

Secondly, the use of $\mathcal{P}_R$ instead of some well-defined probability-theoretic predicate is there for both convenience and generality, as it allows for arbitrarily complex security criterions while also providing for a fairly simple formalization. For instance in the case of the previous example, $(f, \mathsf{A}) \in (\mathcal{F}_s)_R$ iff $\Pr[r \leftarrow \{0,1\}^k, f' := f(r\|\cdot), (m_0, m_1) \leftarrow \mathsf{A}(f') \colon m_0 \neq m_1, f'(m_0) = f'(m_1)] < \epsilon$ as we are defininining when a given A breaks a given f.

This formalization for a primitive allows us to give a rigorous definition for a *fully* black-box reduction (again, directly following [37]).

Definition 3. There exists a *fully black-box* reduction from a primitive $\mathcal{P}$ to a primitive $\mathcal{Q}$ if there exist polynomial-time oracle machines G and S such that:

- Correctness: For every implementation $f \in \mathcal{Q}_F$ we have that $\mathsf{G}^f \in \mathcal{P}_F$.
- Security: For every implementation $f \in \mathcal{Q}_F$ and every machine A, if A $\mathcal{P}$-breaks G^f then $\mathsf{S}^{\mathsf{A},f}$ $\mathcal{Q}$-breaks f.

Essentially, this is just what happened in the proof of Theorem 1, where G is the construction of $h_{r,f}$ based on f while S should be taken as the construction of A_s using f and A_l. It is easy to verify that both parts of the proof are indeed poly-time. Just as importantly, they make only oracle use of f and A_l which means they only call them and do not make any assumptions about precisely how they work.

1.1.3 Degrees of Black-boxness

The framework of a (fully) black-box reduction is fairly general as many of the cryptographic security proofs can actually be seen to follow exactly the pattern laid out above. However, there are a few rare instances in which additional assumptions are made in either one or both parts of the proof and where black-box oracle access is not enough. For that reason, Reingold et al. [37] described a whole hierarchy of different reductions. We will now proceed to give a brief overview of their taxonomy.

The first thing that can be weakened is the restriction that the security proof be uniform, in the sense that it behave the same way for every adversary provided for $\mathcal{P}$. By allowing the security proof to be more dependent on the adversary construction, we get the following definition for (weak[1]) semi black-box reductions.

Definition 4. There exists a *weak semi black-box* reduction from a primitive $\mathcal{P}$ to a primitive $\mathcal{Q}$ if there exist polynomial-time oracle machines G such that:

- Correctness: For every implementation $f \in \mathcal{Q}_F$ we have that $\mathsf{G}^f \in \mathcal{P}_F$.
- Security: For every implementation $f \in \mathcal{Q}_F$, if there exists an POTM A so that A^f $\mathcal{P}$-breaks G^f, then there exists a POTM $\mathsf{S}_{f,\mathsf{A}}$ so that $\mathsf{S}^f_{f,\mathsf{A}}$ $\mathcal{Q}$-breaks f.

Note that we assume $\mathsf{S}^f_{f,\mathsf{A}}$ has oracle access to f because we assume $\mathsf{S}_{f,\mathsf{A}}$ to exist and to be poly-time for non poly-time implementations of f, in which case f might not be fully embeddable into the circuit of $\mathsf{S}_{f,\mathsf{A}}$ and the oracle access becomes strictly neccessary.

This definition allows the security proof of the reduction to depend on the adversary construction. However, the security proof is still black box in a sense, as it is not allowed to change much relative to the implementation f of the underlying primitive $\mathcal{Q}$, i.e., it has to be the same for all f for which A^f $\mathcal{Q}$-breaks f. If we lose even that restriction, we get the following.

Definition 5. There exists a *weakly black-box* reduction from a primitive $\mathcal{P}$ to a primitive $\mathcal{Q}$ if there exist polynomial-time oracle machines G and S such that:

- Correctness: For every implementation $f \in \mathcal{Q}_F$ we have that $\mathsf{G}^f \in \mathcal{P}_F$.
- Security: For every implementation $f \in \mathcal{Q}_F$, if there exists a POTM A that $\mathcal{P}$-breaks G^f , then there exists a POTM $\mathsf{S}_{\mathsf{A},f}$ such that $\mathsf{S}^f_{f,\mathsf{A}}$ $\mathcal{Q}$-breaks f.

However, there is another way in which we could weaken black-boxness, this time on the construction side. Namely, we could allow the construction G for $\mathcal{P}$ to actually vary depending on the underlying implementation f of $\mathcal{Q}$. As this is essentially achieved by switching quantifiers in a definition, it can be applied to

[1] See Chapter 5.

generalize all of the above reduction types. Since the change is analogous in all three cases, we will show it only for semi black-box reductions.

Definition 6. There exists a $\forall\exists$-*semi black-box* reduction from a primitive $\mathcal{P}$ to a primitive $\mathcal{Q}$ if for every implementation $f \in \mathcal{Q}_F$ there exists a POTM G such that:

- Correctness: $\mathsf{G}^f_f \in \mathcal{P}_F$.
- Security: If there exists an POTM A so that A^f $\mathcal{P}$-breaks G^f_f, then there exists a POTM $\mathsf{S}_{f,\mathsf{A}}$ so that $\mathsf{S}^f_{f,\mathsf{A}}$ $\mathcal{Q}$-breaks f.

These generalizations may seem somewhat contrived. However, there are a number of known reductions in the literature that do indeed fail to fit into the fully black box framework and such generalizations are thus useful in accommodating for them as well.

The most widespread way a proof ceases to be fully black box is by using a zero-knowledge proof on the correctness of the underlying primitive. In that case, the construction, although guaranteed to exist, is highly dependent on two additional assumptions. Firstly it is then clearly required that the construction of G be allowed to depend on f as in $\forall\exists$ variants of the above definitions. Secondly and more prohibitively, however, they assume that f is computable by a poly-size circuit in the real world so that they could use it in the zero-knoweldge proofs. Such assumptions are somewhat harder to account for, the reasons for which will become obvious in the next chapter.

However, there are proofs that are inherently non-black box in the security proof as well. For instance, Buldas and Laur [5] give a reduction that is not fully black-box in which the security reduction S assumes the availability of a polynomial amount of extra distributional info about the outputs of A. The information used is not, in general, computable in poly-time, marking their reduction as semi black-box. Although such constructions are rare, they do exist and therefore it makes sense to have a framework available that captures them as well.

There is one additional special case of black-box reductions that is worth mentioning. Reductions are often concerned with just proving additional security properties of a primitive based on security properties already known to hold. These *self-reductions* just use a trivial construction in the first part of the proof where $\mathsf{G}^f = f$. However, no restrictions are set on the second half of the proof that deals with security, and as such, the previous classification of different degrees of black-boxness is still relevant. This approach only makes sense in the case where both $\mathcal{P}$ and $\mathcal{Q}$ have the same functional requirements, but there are places for which this is indeed the case [10, 38].

We will now turn from describing the common cryptographic practice of using reductions to a somewhat more specialized field of proving that no reductions can exist. This is usually done via a methodology called Oracle Separation to which the next chapter is devoted.

CHAPTER 2

ORACLE SEPARATION METHODS

2.1 Oracle Separation

Reductions have played a central role in cryptography for nearly 40 years. Nevertheless, a rigorous and general definition for a reduction was only given fairly recently. The reason for that is actually quite simple: it seems that such a definition is needed only in the case where one is ruling out the existence of such reductions. This was indeed the setting in which the work of Reingold et al. [37] was conducted. However, they were already building upon 15 years of previous work in the field that started with the seminal paper by Impagliazzo and Rudich [27].

The main idea of [27] is borrowed from complexity theory (where it was pioneered by Baker, Gill and Solovay [1]) and is essentially quite simple. One starts by augmenting the computational setting with new "oracle" functionality – for instance, extra computational power or access to an exponential-length "true" shared randomness in the form of a random oracle – and then shows that in this new world, certain things can be proven to either surely exist or to not exist at all.

To be precise, Impagliazzo and Rudich [27] proceed in the following way: They assume a world where $\mathbf{P} = \mathbf{NP}$ (which can be achieved by adding a $\mathbf{PSPACE}$ oracle) and then additionally introduce a random oracle into the setting. They then prove that in this setting, one can construct provably secure one-way functions and one-way permutations but no secure key agreement is possible.

In short, their result is the following: in a computational world with an oracle $\mathcal{O}$ that combines a random oracle with a $\mathbf{PSPACE}$ oracle, one-way permutations exist but secure key agreement does not. This in itself might not seem like much. However, it turns out that this is actually sufficient to rule out the existence of a fully black-box reduction from key agreement to one-way permutations. Namely, if such a reduction would exist, it would also exist relative to $\mathcal{O}$ as both G and S that are required by Definition 3 should work equally well even when new oracle functionality is added. However, this clearly cannot be the case, as relative to $\mathcal{O}$, the thing being constructed (key agreement) cannot exist while the thing it would be constructed from (a one-way permutation) definitely does.

Reingold et al. [37] formalize this result in a more general setting and in the following way:

Definition 7. There exists a *relativizing reduction* from a primitive $\mathcal{P}$ to a primitive $\mathcal{Q}$, if for every oracle Π, if $\mathcal{Q}$ exists relative to Π then so does $\mathcal{P}$.

Lemma 1. *For any two primitives $\mathcal{P}$ and $\mathcal{Q}$, if there exists a fully-black-box reduction from $\mathcal{P}$ to $\mathcal{Q}$ then there exists a relativizing reduction from $\mathcal{P}$ to $\mathcal{Q}$ as well. Conversely, if no such relativizing reduction exists then no fully-black-box reduction can exist either.*

However, it turns out that one can usually rule out even stronger forms of reductions if one is capable of embedding the required oracle $\mathcal{O}$ directly inside an implementation f of $\mathcal{Q}$ such that $\mathcal{O}$ can be computed with oracle access to f and vice versa. This technique was first used by Simon [39] but was later strongly generalized by Reingold et al. [37].

Definition 8. We say that a primitive $\mathcal{Q}$ allows embedding if for any $f' \in \mathcal{Q}_f$ and any oracle $\Pi : \{0,1\}^* \to \{0,1\}$ there exists $f \in \mathcal{Q}_f$ such that the following hold:

- There exists a POTM $\mathsf{G}^{f',\Pi}$ that computes f given oracle access to Π.
- There exists a POTM P^{f} that computes Π given oracle access to f.
- If there exists a polynomial-time oracle machine M^{Π} that $\mathcal{Q}$-breaks f then there exists a POTM $(\mathsf{M}')^{\Pi}$ that $\mathcal{Q}$-breaks f'.

As an example, one-way functions are embeddable, as you can construct $f : \{0,1\}^n \to \{0,1\}^m$ by taking $f = f'$ everywhere except on inputs where the first $\frac{n}{2}$ input bits are all 0 in which case the $2^{n/2}$ such inputs are just used to encode the responses to queries of Π. Such an embedding does not interfere with security arguments, as it increases the success probability by only a negligible amount.

Theorem 2 (Reingold et al. 2004)**.** *Let $\mathcal{P}$ be any primitive and $\mathcal{Q}$ be any primitive that allows embedding. Then there exists a relativizing reduction from $\mathcal{P}$ to $\mathcal{Q}$ if and only if there exist a $\forall\exists$-semi-black-box reduction from $\mathcal{P}$ to $\mathcal{Q}$.*

As most primitives seem to allow for embedding, their result can be used to generalize most of the known separation results.

2.2 Historical Perspective

The result of Reingold et al. [37] given in the above represents the state of the art in oracle separation methodologies in cryptology. We will now provide a historical overview of how the subfield developed by giving an overview of four milestone results.

2.2.1 Rudich and Impagliazzo 1989

The first oracle separation result was proven by Rudich and Impagliazzo [27] in 1989. Their proof model, being the oldest known for establishing cryptographic separations, is still in use today, probably because of its simplicity and lack of bothersome technical details. Their proof methodology can be formalized as the following meta-theorem.

Theorem 3 (Rudich, Impagliazzo 1989)**.** *Assume a world where* $\mathbf{P} = \mathbf{NP}$ *that is augmented by the addition of a random oracle* $\mathcal{O}$*. If in such a world a primitive* $\mathcal{Q}$ *exists but a primitive* $\mathcal{P}$ *does not, then no fully black-box reduction can exist from* $\mathcal{P}$ *to* $\mathcal{Q}$*.*

Their model makes two very simple computational assumptions – existence of an exponential amount of public shared randomness (i.e., a random oracle) and computational powers just above our current reach ($\mathbf{NP}$ problems being tractable). Despite its simplicity, it has proven to be a fairly powerful proof model, as it has seen continual use throughout the past two decades [27, 19, 20, 21].

It is also worth noting that if $\mathcal{Q}$ is embeddable, the results in this model can be generalized to also exclude $\forall\exists$-semi black-box reductions by applying Theorem 2. As such an embedding seems possible for most of the primitives, the proof model is fairly strong in terms of the results that can be achieved by following it.

2.2.2 Simon 1998

The result of Simon [39] provided the second conceptual breakthrough in separation methodologies and paved way for most of the following results in the field. The main contribution of that paper was to notice that one can actually custom-design the oracle used for the separation to provide exactly the functionality needed and nothing more.

To be precise, the author constructed an oracle f that was itself an instance of $\mathcal{Q}$ (in his case, a one-way permutation) but that also had embedded into it an adversary for every implementation of $\mathcal{P}$ (a collision-finder for a hash function family with a specified circuit). As f itself was proven to be a secure instance of $\mathcal{Q}$, but would clearly break any instance of $\mathcal{P}$, it would be clear that no black-box reduction could exist from $\mathcal{P}$ to $\mathcal{Q}$.

Essentially, the proof methodology of the paper was exactly the same as formalized in Theorem 2. However, the original paper of Simon [39] concerns just the one separation without giving any thought to the more general meta-level implications. The fully general result was proven only 6 years later by Reingold et al. [37].

2.2.3 Reingold, Trevisan and Vadhan 2004

This paper presented the first fully general framework for separation results by giving formal definitions for primitives and providing for a full hierarchy of reductions of varying degrees of black-boxness. As most of our previous exposition is based directly on their results, we will refrain from discussing them further here.

2.2.4 Hsiao and Reyzin 2004

This paper [25] took a step forward in a yet another direction. Noting that the Rudich-Impagliazzo model can be used to obtain stronger results than originally intended, Hsiao and Reyzin sought to find a simpler proof model which would still be sufficient for ruling out fully black-box reductions. They determined that one can actually introduce an extra oracle into the equation, considerably simplifying the framework for proving separation results.

Theorem 4 (Hsiao and Reyzin 2004)**.** *If there are two oracles A and f such that*

- *There is a poly-time oracle machine Q^f that implements $\mathcal{Q}$.*
- *For every poly-time oracle machine P^f that implements $\mathcal{P}$, there is a poly-time oracle machine $\mathsf{S}^{f,A}$ that breaks P^f.*
- *There is no poly-time oracle machine T such that $\mathsf{T}^{f,A}$ breaks Q^f.*

then there exist no fully black-box reductions from P to Q.

This allowed for conceptual simplification as the oracle could freely be divided into two parts – f that helps to realize $\mathcal{Q}$ and A that directly deals with breaking any instance of $\mathcal{P}$.

That might not seem like much in the first glance. However, this should be put into context by noting that most of the complexity in Simon [39] arose because the universal adversary oracle there actually had to be able to break constructions that had access to that same adversary oracle (i.e., constructions of the form $\mathsf{P}^{f,A}$ instead of just P^f). Avoiding such self-referencing makes proofs considerably easier both to write and to follow and for that reason this model has been used by numerous subsequent authors [14, 3, 7, 28], this in spite of the fact that it only rules out fully black-box reductions instead of semi black-box reductions that can be ruled out with the previously proposed methods.

2.3 Proofs for Lower Bounds

The method of oracle separation has proven quite fruitful in exploring the limits of black-box reductions by showing that there are numerous cases in which such a reduction between two primitives is clearly impossible. Nevertheless, we do have

reductions for many important cases and it would also be very interesting to study the limits of reductions that do actually exist. Two most notable publications from that direction of research are those of by Kim, Simon and Tetali [29] and Gennaro and Trevisan [18].

2.3.1 Kim, Simon and Tetali 1999

A big step in that direction was taken by Kim, Simon and Tetali [29] who noticed that although a reduction (due to Naor and Yung [34]) exists from one-way permutations to universal one-way hash functions, it is so inefficient as to be nearly useless in practice. They showed that some of that inefficiency is inherently unavoidable by proving that any fully black-box reduction for an ϵ-compressing universal one-way hash function would need to call the underlying permutation a number of times proportional to the square root of the security parameter. This was done in a similar fashion to [39] but with the added assumption that the construction makes less than $\Omega(\sqrt{n}/\log(n))$ oracle queries. As all the reductions that make less queries are ruled out, the result does indeed imply that more queries are needed for a black-box reduction to be feasible, hence providing a lower bound on the number of oracle queries.

2.3.2 Gennaro and Trevisan 2000

The bound of [29] was strengthened to linear by Gennaro and Trevisan [18] which could be interpreted as proving the optimality of Naor-Yung construction (up to a constant factor). However, their result was interesting for another reason as well. Namely, all the previous separation results hold only in the uniform model of computation. This is because nearly all of them make essential use of oracle families, where an oracle is chosen randomly, instead of starting off with one fixed oracle. However, to prove the separation results, one needs to fix a suitable oracle and this is usually done via non-constructive argumentation. For that approach to work, one has to assume a countable number of adversary constructions, which is the case in the uniform model but ceases to be true in the non-uniform case.

Gennaro and Trevisan took a different approach. They first proved that under appropriate random oracles, one-way functions and permutations exist that are secure even in the non-uniform model (thus replicating the result of Impagliazzo [26]). They then proceeded to show that the existence of UOWHF that makes less than a linear number of oracle calls (i.e., calls to one-way permutation) would automatically imply the existence of a UOWHF without any assumptions, which would, in turn, imply $\mathbf{P} \neq \mathbf{NP}$. This means that constructing such a reduction is at the very least beyond the current scope of knowledge[1]. Their argumentation cleverly avoids the need to pick one exact oracle for the separation and as such

[1] Although, technically, the proof can be made unconditional by giving the proof relative to a $\mathbf{PSPACE}$-oracle in which case $\mathbf{P} = \mathbf{NP}$

will also work even when non-uniform adversaries are concerned. Their approach can be formalized as follows.

Theorem 5 (Gennaro and Trevisan 2000)**.** *Assume a non-uniform computational world where* $\mathbf{P} = \mathbf{NP}$ *and where there is an oracle* f *that implements* $\mathcal{Q}$ *securely even in the non-uniform model. If* $\mathcal{P}$ *does not exist in such a world, no (potentially non-uniform) semi black-box reduction from* $\mathcal{P}$ *to* $\mathcal{Q}$ *is possible (even relative to non-uniform adversaries).*

This presents a significant step forward, as most of the results that prove the existence of a reduction are shown in the non-uniform model, i.e., they assume security against non-uniform adversaries and show that that is preserved in the reduction. When a separation exists in the uniform model only, it still leaves room for the case where the underlying primitive is insecure against non-uniform adversaries. In that case, however, the separation result becomes nearly meaningless in the non-uniform model since it is essentially just showing that we cannot give a general construction for a secure instance of $\mathcal{P}$ from an insecure instance of$\mathcal{Q}$.

Formally, to fix that problem, one would just need to show that the underlying primitives are secure under non-uniform adversaries, which is somewhat easier than doing the whole proof in the non-uniform model. Still, the work of Gennaro and Trevisan [18] can be seen as having pointed out a fairly major technical shortcoming in the previous proofs. This has been acknowledged implicitly by other authors [15, 16, 17] who have since used the proof model of [18] with the justification that it can deal with non-uniform reductions.

CHAPTER 3

HASH-TREE BASED TIME-STAMPING

Most of the author's original work is concerned with oracle separation methods in general. However, some of the results do have practical applications to the field of time-stamping schemes. As such, it makes sense to give a brief introduction into that topic as well.

3.1 Model of Time-stamping

The underlying problem for time-stamping is fairly simple. Imagine an inventor who has just invented something that will change the world and give him a large profit margin whilst doing it. However, he fears that someone else may also come up with the same thing in the near future or, even worse, steal the idea and be able to patent it before he himself does. The inventor is therefore interested in binding the document containing the invention with the current time and date so that he would later be able to prove that he has indeed had that document at least since that time.

First of all, it is worth noting that conventional means of achieving this have been in place for hundreds of years already. Firstly – patent offices themselves function in that role, as once they accept something for investigation, they mark down the time they received the documents. If someone later tries to file for a patent for the same idea under a different name, he will be turned down. For documents that do not directly contain inventions, notaries can serve the same role, as in most countries, one of the main functions of notaries is to vouch for the validity of documents and agreements.

However, both of these options have numerous flaws. Firstly, both patents and notary services are fairly expensive, which means that a poor inventor might not be able to defend himself. Secondly, both require you to reveal the document that you are time-stamping, which one might be somewhat reluctant to do in certain cases. Thirdly and possibly most importantly, they are very prone to attack by a corrupted patent clerk or notary, who can later just reformat the same document,

write an earlier date and then put his friends name under it instead of the real inventors.

The first two of these problems can be mitigated by having a cheap trusted authority that is willing to time-stamp closed envelopes. This is usually available in most countries under the guise of the national postal service. Indeed, sending a document to yourself in a sealed envelope and then keeping it sealed til the date of creation needed to be proved (like in court) was often used to settle patent and copyright disputes in earlier times[1]. However, this approach does not solve the third problem, as a postal clerk can still fairly easily use the postal stamp that is a few days old.

3.2 Scheme of Haber and Stornetta

In a modern world, where most documents are stored on digital media rather than paper and where a larger company can easily have thousands of documents created per minute, the preceding solutions are clearly inadequate if one seeks to have everything stamped. As such, a large-scale digital solution is clearly called for. The main ideas for such a scheme were laid out by Haber and Stornetta [23].

Their main idea was to compute a hash value of the document and then combine all these hash values for a whole day into a single (short) bitstring, which can then be published in a daily newspaper with large circulation on the next day. As recalling all the newspapers of any given day is infeasible (except in a truly Orwellian society), this will provide for very strong dating verification. Using hash functions also gives guarantees on privacy. What is then left is only the third problem, i.e., the worry that a malicious time-stamping authority might be able to back-date documents of his choosing.

The scheme therefore tries to make it hard for the authority to later certify time-stamps on documents that he did not actually use while creating the value that was published in the paper. This is done by using a hash tree to create the certificates for time-stamped documents.

The scheme uses two distinct hash functions – $h_c : \{0,1\}^* \to \{0,1\}^k$ to compute the initial hash values of the document (and which can be done by the clients themselves to ensure privacy) and $h_s : \{0,1\}^{2k} \to \{0,1\}^k$ which is used by the server to combine the client hash values into the published value. All the client hash values $x_1, \ldots, x_n \in \{0,1\}^k$ are hashed together in a tree structure (called hash or Merkle tree), yielding one final k-bit hash value $r \in \{0,1\}^k$ that is then published. Each client is sent a certificate $c = (x, n, z)$ where x is the value being certified, $n = n_1 n_2 \ldots n_l$, $n_i \in \{0,1\}$ describes the path from x down to the root and $z = (z_1, \ldots, z_l) \in (\{0,1\}^k)^l$ gives the information to verify that path.

[1] At some point, US court system stopped accepting postal stamps as proof of timing and since then, the practice has gradually declined.

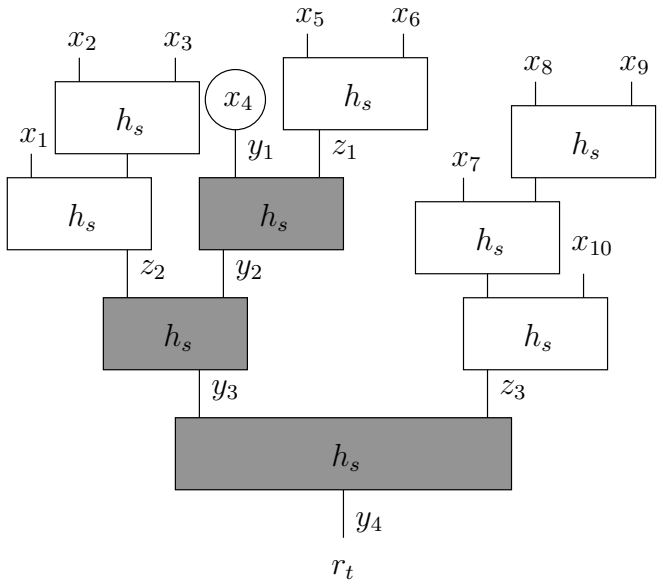

Figure 3.1: A larger tree with a path marked from x_4.

As an example, we describe the certificate of x_4 in fig. 3.1. The sequence n encodes the structure of the path starting from the original value x_4 downwards while the sequence z gives the other inputs used alongside x_4 and values calculated from that. As x_4 is the left input for the first h_s invocation on the path, $n_1 = 1$ and we take z_1 to be the other half of the input for the h_s invocation (which in this case is $h_s(x_5, x_6)$). We now move down to the next invocation on the path and see that the output y_2 of the previous call is now the right input of the new h_s invocation. We thus set $n_2 = 0$ to signify that y_2 is the right input and then take z_2 to be the second input into this box (which in this case is $h_s(x_1, h_s(x_2, x_3))$). The third and final element on the path has the previous h_s calls output y_3 as the left input again so we set $n_3 = 1$ and take z_3 to be the other input again. Since there are no more h_s invocations, the certificate for x_4 for the tree given in fig. 3.1 is $c = (x_4, 101, (z_1, z_2, z_3))$.

Essentially, the certificate just specifies a *hash chain* from the certified value x down to the supposed root value r, providing information both about its shape n and the other values $z_1, \ldots, z_l$ that are used in the hashing. The verification algorithm V just computes the end result of the hash chain. One can then verify the validity by comparing the output of V with the published public root value r.

This motivates the following general definition of a time-stamping scheme.

Definition 9. A time-stamping scheme is a triple $\mathsf{T} = (\mathsf{Com}, \mathsf{Cert}, \mathsf{Ver})$ of efficient algorithms, where:

- Com is a commitment algorithm which, on input a set $\mathcal{X}$ of requests, outputs a commitment $r = \mathsf{Com}(\mathcal{X})$.

- Cert is a certificate generation algorithm which, on input a set $\mathcal{X}$ and an element $x \in \mathcal{X}$, generates a certificate $c = \mathsf{Cert}(x, \mathcal{X})$.
- Ver is a verification algorithm which, on input a request x, a certificate c and a commitment r, outputs 1 or 0, depending on whether x is a member of $\mathcal{X}$ (the set that corresponds to the commitment r). It is assumed that for every set $\mathcal{X}$ of requests and every member-request $x \in \mathcal{X}$ the following correctness condition holds:

$$\mathsf{Ver}(x, \mathsf{Cert}(x, \mathcal{X}), \mathsf{Com}(\mathcal{X})) = 1 \ . \tag{3.1}$$

In the concrete example of tree-based time-stamping, Com outputs the root r of the hash tree, Cert is the procedure of creating z, n, and Ver just checks whether $\mathsf{V}(x, n, z) = r$.

3.2.1 Security Definitions

As mentioned before, the main security concern for such a scheme is for a cheating time-stamping authority who should not be able to back-date any documents. Haber and Stornetta [22] originally defined security in just such a way – they considered their scheme broken if the authority could produce a time-stamp for any document of his choosing that he did not directly use in the hash tree construction.

However, Buldas and Saarepera [10] showed that such a security criterion was obviously flawed as a trivial attack existed against it: it sufficed for the adversary to choose two documents, d_1, d_2, generate their hash values $x_1 = h_c(d_1), x_2 = h_c(d_2)$ and instead of using these values directly in the hash tree construction just perform the first step in private by computing $x = h_s(x_1, x_2)$ and then have the system time-stamp that value x. He now has two distinct time-stamped documents while officially having only queried one stamp.

This flaw is easy to fix by just bounding the shape of the trees by fixing the depth of all the leaves. However, it should be fairly easy to convince oneself that the previous "attack" micht actually a fairly legitimate and perhaps even desired behavior, as it allows the clients to keep secret how many documents they are having stamped. In reality, it does not constitute a real attack against the common understanding of time-stamping security, as it allows one to "back-date" only documents that actually did exist at the time they are back-dated to. Buldas and Saarepera [10] thus concluded that it is only the security definition that is flawed and set out to give a definition more fitting for practical purposes by only considering back-dating of "novel" documents.

Novelty is a somewhat elusive concept, however. In [10] they settled for an entropy-based definition where novel was taken to mean a document that is chosen from any distribution with high-enough max-entropy. To prove security for such a case, they needed to assume a fairly exotic security criterion called chain-resistance from the hash function h_s.

Definition 10. A hash function $h\colon \{0,1\}^{2k} \to \{0,1\}^k$ is (t,ϵ)-*chain-resistant* (relative to a distribution $\mathcal{D}_k$ on $\{0,1\}^k$) if for every t-time adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$

$$\Pr[(r,a) \leftarrow \mathsf{A}_1, x \leftarrow \mathcal{D}_k, (n,z) \leftarrow \mathsf{A}_2(x,a)\colon \mathsf{V}(x,n,z) = r] < \epsilon \ , \qquad (3.2)$$

where x, n, z, r are as in the example above.

Unless specified otherwise, the distribution $\mathcal{D}$ is chosen to be the uniform distribution over all the possible bitstrings of length k.

This notion of chain-resistance is fairly hard to test for, however. As such, one would be very interested in constructing functions secure in this sense from some more sensible security assumptions. Buldas and Saarepera [10] showed that one cannot use black-box methods to show that a collision-resistant function is secure in this sense, but they still left open the question of whether a function is secure in this sense could possibly be constructed based on a collision-resistant function, perhaps with just some minor modifications.

As mentioned before, such a security condition is only required in the case where the tree shape is not fixed, i.e., if certificates of any shape are accepted. It was shown by Buldas and Saarepera [10] that if one restricts the scheme so that only a polynomially bounded number of different valid hash chain shapes are considered valid, it turns out that collision-resistance of h_s is enough to achieve security. Limiting the certificate space is actually fairly easy to do (by, for instance, having the verification algorithm check that the hash chain is of exactly the prescribed length) and the security is then much easier to prove based on standard assumptions. For that reason, such *bounded* time-stamping schemes are the ones actually used in practical implementations (such as GuardTime - www.guardtime.com).

CHAPTER 4

POSSIBILITY OF REDUCING CHAIN-RESISTANCE TO COLLISION-RESISTANCE

In this chapter we cover the work that was presented by the author and his supervisor in the paper "Can We Construct Unbounded Time-Stamping Schemes from Collision-Free Hash Functions?" [7].

The research direction originally proposed for this thesis was to investigate the possibility of constructing chain-resistant hash functions based on collision-resistant ones. Since Buldas and Saarepera [10], it has been known that no black-box self-reduction can exist for that, i.e., that no hash function can be proven to be chain-resistant based purely on the assumption that it is collision-resistant. However, that still left open the possibility of constructing chain-resistant functions from collision-resistant ones. Our aim was to investigate the feasibility of such an approach.

As there seemed to be no obvious ways of giving such a construction, we concentrated on trying to prove that such a black-box reduction is actually impossible. We attempted a fairly straightforward approach based on the framework of Hsiao and Reyzin [25] where we used a random oracle to implement the underlying collision resistant function $h : \{0,1\}^m \rightarrow \{0,1\}^n$. Constructing the adversary part of the oracle turned out to be somewhat tricky, however. To be precise, we needed an oracle $\mathcal{O}$ that would be able to break any construction H^h with respect to the chain-resistance property, which meant it had to be able to provide time-stamping certificates to at least a (reverse) polynomial fraction of all the possible inputs for H^h. As a collision can easily be extracted from two conflicting chains (i.e., two chains that result in the same root value but cannot be part of the same tree), the most obvious candidate for a chain-resistance breaking oracle would be $\mathcal{O} = (\mathcal{O}_1, \mathcal{O}_2)$ where $\mathcal{O}_1(\mathsf{H})$ just returns the root value of a large (exponential-sized) hash tree for H^h which then allows $\mathcal{O}_2(x)$ to honestly deal out a certificate for x, assuming x was indeed used as input at some point inside the tree.

However, this approach has very clear limitations. For instance, it can be shown that if the full Merkle tree where all the k-bit inputs are provided is constructed by the oracle, then H can be built in such a way as to guarantee that a root value contains a collision for h. The way to achieve this is actually quite simple. The idea is to construct $\mathsf{H}^h : \{0,1\}^{4m} \to \{0,1\}^{2m}$ so that each input to H is a pair of inputs for h. H just tests, whether either of the pairs is a collision for h and if that is the case, passes that pair downwards. Since we are dealing with a full tree, each $2m$ bit input is presented, which means that a collision is found in at least one of the leaves. As a collision always propagates down to the root, this guarantees that the root value will indeed yield the desired result.

However, there is no requirement to generate a full tree, as it is completely acceptable for the oracle to break H on a fraction of the inputs, as long as the fraction is non-negligible. However, one would still need to be able to show that the partial adversary is not malleable to similar exploitation.

To prove a full separation result, one would need to show that no adversary construction for collision-resistance could benefit from the chain-resistance adversary oracle. Exploring the limits of Hsiao and Reyzin [25], it turned out that it is actually sufficient if we can construct such an unexploitable oracle once the adversary has already been specified.

Theorem 6. *If for all POTM pairs $R = (\mathsf{G})$ there exist A_R and f_R such that*

(a) f_R implements $\mathcal{Q}$.

(b) There is a polynomial-time oracle machine D^{A_R, f_R} that breaks G^{f_R}.

(c) There is no polynomial-time oracle machine B such that B^{A_R, f_R} breaks f_R,

then there exist no black-box reductions from $\mathcal{P}$ to $\mathcal{Q}$.

This meant that to prove a separation result, one would need to be able to show that for each adversary construction, an oracle can be chosen so that no adversary can be constructed using it.

We assume the adversary makes just one $\mathcal{O}_1$ query, i.e., sees the root of only one large hash tree. This is actually a completely reasonable assumption, as for fully black box reductions, we can assume a fixed construction $\mathsf{H} = \mathsf{P}^h$ for H^h and that is the only hash function to which we need to answer the $\mathcal{O}_1$ query.

We first determined that if only a polynomial fraction of inputs is to be presented, one can always construct an oracle so that the preceding method of exploitation that just checked whether input values produced a collision would not work. Indeed, we showed an even more general result, namely that any construction of H that made all of its h-dependent decisions based solely on collision queries of type $h(x_1) = h(x_2)$ will fail to produce a collision. This was done simply by showing that we can always construct a tree in such a way as to avoid showing any collisions inside the constructed tree.

Theorem 7. *Assume that* $\mathsf{H} : \{0,1\}^{2k} \to \{0,1\}^k$ *is a POTM construction with an oracle* c_h *for* h *where* $h : \{0,1\}^n \to \{0,1\}^m$ *is chosen uniformly from a family of hash functions* $\mathcal{F}$*. We also assume that for uniformly chosen* (x, y)*,* $x \neq y$*, the probability that it forms a collision is* $2^{-\omega(\log(n))}$*. Then the probability of being able to construct a hash tree of size at least* $\frac{2^k}{p(n)}$ *for* H *that doesn't show any collisions of* h *is overwhelming.*

We then tried to investigate, what other types of queries about h would still allow us to prove the same result. The preceding result was fairly easy to generalize to the case where the oracle would also answer questions of the form $h(x) = y$ for given x and y. Using a clever trick, it was also possible to show, that comparison queries of the form $h(x_1) \geq h(x_2)$ for given x_1, x_2 can also be dealt with. In total, we were able to show that if H only uses these types of information about h, then the a suitable choice of oracle is guaranteed to exist.

We will now discuss the implications of the results. First thing we have to note is that we did not rule out the existence of a black-box reduction. However, we did show that if such a reduction were to exist, it would have to be pretty strange. For instance, it would either have to use two separate constructions H and H' in its security proof, or, if it made due with just one, the security proof would have to make essential use of the bit representation of the output of h – as any adversaries that just used ordering information or input-output pair verification are ruled out by our results. In this sense, our result can be seen as somewhat analogous to the impossibility results known for the "generic model of groups" which too only assume limited access to the actual implementation. It is the view of the author that these results can be taken as strong evidence against the existence of a general black-box reduction.

CHAPTER 5

ORACLE SEPARATION IN THE NON-UNIFORM COMPUTATIONAL MODEL

In this chapter we will try to give a brief overview of the work of the author of this thesis and his co-authors that was presented in "Oracle Separation in the Non-Uniform Model" [6] and "Black-Box Separations and their Adaptability to the Non-Uniform Model" [9].

As noted previously, most of the separation results that have been proven to date only work in the uniform model of computation. We were interested in remedying the situation by trying to strengthen the separations that have already been shown in the uniform model into the non-uniform model.

Throughout this whole chapter, we will part with our normal conventions and assume that Turing machines are non-uniform and not randomized, unless otherwise stated.

The main reason why most proofs of separation results fail in the non-uniform model is actually quite simple. Separation oracles that are used are generally not deterministic but rather chosen from a large (usually infinite) family of different possibilities. However, for the separation theorems to hold, one fixed oracle needs to be demonstrated. To do that, the authors usually resort to a technique that is best called "Oracle extraction" where the existence of a fixed oracle that is suitable for the separation is derived using averaging and counting arguments over the sets of all the possible (adversary) constructions. Due to the inherent use of countability in these arguments, this step was irredeemable in the non-uniform model and was replaced with a probability-theoretic argument that worked directly with oracle families rather than one fixed oracle.

However, there were many technical problems that needed to be overcome in order to formalize such an approach. First of all, the definition of a primitive given by Reingold et al. [37] had to be revised to account for adversaries that are successful only with some probability. It also seemed to make sense to draw a

clear distinction between deterministic and randomized primitives. This resulted in the following two definitions.

Definition 11. A *deterministic primitive* $\mathcal{P}$ is a set of functions of type $f\colon \{0,1\}^* \to \{0,1\}^*$. Every primitive has an *advantage function* $\textsc{Adv}_k^{\mathcal{P}}(\cdot,\cdot)$, which given as input the security parameter $k \in \mathbb{N}$, an instance f of $\mathcal{P}$, and an oracle Turing machine $\mathsf{A}^{\mathcal{O}}$ (an *adversary*) returns a real number $\textsc{Adv}_k^{\mathcal{P}}(f, \mathsf{A}^{\mathcal{O}}) \in [0,1]$ (the *advantage* of $\mathsf{A}^{\mathcal{O}}$). The function $\textsc{Adv}_k^{\mathcal{P}}(f,\cdot)$ is extended to probabilistic Turing machines by taking the average over their randomness strings[1]. We say that $\mathsf{A}^{\mathcal{O}}$ *breaks* an instance f of $\mathcal{P}$ if $\textsc{Adv}_k^{\mathcal{P}}(f, \mathsf{A}^{\mathcal{O}}) \neq k^{-\omega(1)}$. If for a fixed oracle $\mathcal{O}$ no probabilistic poly-time oracle Turing machine $\mathsf{A}^{\mathcal{O}}$ breaks f then f is said to be *secure relative to* $\mathcal{O}$.

Definition 12. Let $\mathcal{P}$ be a deterministic primitive. Then the corresponding *randomized primitive* $\mathcal{P}_{\mathrm{r}}$ is the set $\mathcal{P}_{\mathrm{r}} = \{f : \Omega \times \{0,1\}^* \to \{0,1\}^* | \forall r \in \Omega : f(r,\cdot) \in \mathcal{P}\}$ and $\textsc{Adv}_k^{\mathcal{P}_{\mathrm{r}}}(f,\cdot) = \mathop{\mathbf{E}}_{r\in\Omega}\left[\textsc{Adv}_k^{\mathcal{P}_{\mathrm{r}}}(f(r,\cdot),\cdot)\right]$, where Ω is a randomness space.

Both of these definitions are actually fairly natural extensions of those provided in [37]. However, they allow for a more precise handling of reductions by providing convenient means of averaging the success probabilities of adversaries over different oracles.

Drawing the distinction between deterministic and randomized primitives is actually a conceptual simplification suggested by one of the co-authors. Essentially, since no computability assumptions are made in the definitions, a family of oracles all of which implement $\mathcal{Q}$ can be seen as just a single randomized implementation where each specific oracle corresponds to a randomness string. Most of the separation results first essentially prove that no reduction can exist from such a (potentially randomized) instance of $\mathcal{P}$ to such a randomized instance of $\mathcal{Q}$ and then perform the oracle extraction step, which essentially amounts to showing that the non-existence of a reduction in the randomized case implies the non-existence of a reduction in the deterministic case as well.

The key intuition behind the main result of [6] is to prove the same implication, but with more direct probabilistic argumentation that would avoid oracle extraction. However, there seem to be inherent limitations on when that can be accomplished. The notion of semi-black-box reduction was first introduced in [19]. It was later slightly redefined in a somewhat weaker way in [37]. It turns out that a distinction between a stronger and a weaker definition is actually crucial, as the boundary beyond which generic strengthening to non-uniform model becomes impossible seems to occur just on the border of the following (stricter) definition and the Definition 4 given in Chapter 1.

[1]Each fixed randomness string gives a deterministic poly-time Turing machine for which $\textsc{Adv}^{\mathcal{P}}()$ is already defined.

Definition 13. There exists a *strong semi black-box* reduction from a primitive $\mathcal{P}$ to a primitive $\mathcal{Q}$ if there exists a polynomial-time oracle machine G such that:

- Correctness: For every implementation $f \in \mathcal{Q}_f$ we have that $\mathsf{G}^f \in \mathcal{P}_f$.
- Security: For all POTM A there exists a POTM S_A so that for every implementation $f \in \mathcal{Q}_f$, if A^f $\mathcal{P}$-breaks G^f, then S_A^f $\mathcal{Q}$-breaks f.

The only difference between the two notions is that in the weaker, more general case of Definition 4, the construction S_A is allowed to depend on both A and f, while in the stricter case, it has to be the same for all f.

When using the whole family of oracles for separation instead of just a single fixed oracle, one more complication arises. For the averaging argument to work, some uniformity needs to be assumed from the security reductions. This can be formalized by the following notion.

Definition 14. We say that a reduction from $\mathcal{P}$ to $\mathcal{Q}$ is *poly-preserving* if the corresponding mapping $\mathsf{A} \mapsto \mathsf{B}$ in the security guarantee (S) decreases the advantage by at most a polynomial amount, i.e., exists $c \geq 1$ such that

$$\mathrm{Adv}_k^{\mathcal{Q}}(f, \mathsf{B}) \geq \left[\mathrm{Adv}_k^{\mathcal{P}}(\mathsf{P}^f, \mathsf{A})\right]^c \quad .$$

This finally allows us to state the separation theorem for the non-uniform model that deals directly with oracle families.

Theorem 8. *If we consider only poly-preserving reductions, the existence of fully black-box or strong semi black-box reductions for randomized primitives (in the non-uniform world) implies the corresponding existence results for deterministic primitives and vice versa.*

The same line of research is continued in the second paper [9], which takes a critical look at the previous work on the different reduction types and the methods used to rule them out. Firstly, it turned out that the problem statement of [6] had been somewhat misguided. Namely, it turned out that the standard oracle extraction techniques are actually sufficient to rule out non-uniform separations for fully and strong semi black-box reductions in the randomized case, allowing us to drop the poly-preserving assumption for the cases covered in [6]. Nevertheless, it was possible to salvage the averaging-based separation approach to give realistic separation conditions for the weaker reduction types.

This systematic approach results in a unifying framework that extends the taxonomy of Reingold et al. [37], combining their results with those of Hsiao and Reyzin [25] and Buldas et al. [6]. The reductions form a linear hierarchy with Fully Black-box reductions being the strongest and Variable Semi Black-box reductions being the weakest. This means that if a separation result is proven that rules out one of the weaker types of separations, it trivially implies that none of

the stronger types of reductions exist either. For each reduction type, there are two ways to approach the separation – either by oracle extraction or by averaging arguments. The criterions derived by averaging-based methods are usually weaker and easier to apply, but they only work for poly-preserving reductions[2]. The new taxonomy is summarized in Table 5.1.

As is evident from Table 5.1, separations based on oracle extraction will need a countability argument if they are to go any lower than Strong Semi Black-box, and thus will only work for the non-uniform model from that point onward. However, it is worth noting that averaging-based techniques might still apply, provided that strong enough bounds can be proven for non-uniform constructions $S^f_{\varphi(f)}$ that have oracle-dependent advice strings $\varphi(f)$.

Such oracle-dependent advice has only recently come into consideration. Unruh [40] provided a general method whereby most results proven in the standard Random Oracle Model can be transformed to also hold in the model where adversaries have access to a polynomial-length oracle-dependent advice. As separation oracles are rarely pure random oracles, this result on its own is of only limited use.

However, as conjectured by Unruh [40], a similar result can be proven for arbitrary oracles. To be precise, we prove the following.

Theorem 9. *Let $\mathcal{F}$ be any distribution of Oracles and let $f \leftarrow \mathcal{F}$. We say that f is consistent with a matching $M = \{x_1 \to y_1, \ldots, x_m \to y_m\}$ if $f(x_i) = y_i$ for all $i \in \{1, 2, \ldots, m\}$. Let $\varphi(f)$ be an oracle function with an output of length p. Then there is an oracle function S such that S^f is a matching of length n and the following holds: For any probabilistic oracle Turing machine A that makes at most q queries to its oracle, $\Delta(\mathsf{A}^{\mathcal{F}}(\varphi(\mathcal{F})); \mathsf{A}^{\mathcal{F}/S}(\varphi(\mathcal{F}))) \leq \sqrt{\frac{pq}{2m}}$, where $\mathcal{F}/S$ is an oracle sampled according to $\mathcal{F}$ conditioned only on being consistent with $S^{\mathcal{F}}$ (which is also a random variable).*

This theorem allows us to generalize the separation result of Simon [39] to hold in the Weak Semi Black-box sense and in the non-uniform model, thus showing that the averaging-based approach does indeed have benefits over the standard extraction-based methods.

[2]So the Reduction Condition given in the table is somewhat different for the Averaging-based approach, but the quantifier order remains the same.

Table 5.1: Reduction types and separation conditions for oracle extraction and averaging-based separations.

Type	Reduction Condition	Oracle Extraction Condition	Averaging-based Condition
Fully bb	$\underset{\text{of}}{\exists} p \underset{\text{pol}}{\exists} S \forall f \forall A$: A breaks $p(f) \Rightarrow$ $S^{A,f}$ breaks f	$\underset{\text{of}}{\forall} p \underset{\text{pol}}{\forall} S \exists \mathcal{F}$: $\mathop{\mathbf{E}}_{f,A \leftarrow \mathcal{F}} [\textsc{Adv}_k(A, p(f))] = 1 - k^{-\omega(1)}$ $\mathop{\mathbf{E}}_{f,A \leftarrow \mathcal{F}} \left[\textsc{Adv}_k(S^{f,A}, f)\right] = k^{-\omega(1)}$	$\underset{\text{of}}{\forall} p \underset{\text{pol}}{\forall} S \exists \mathcal{F}$: $\mathop{\mathbf{E}}_{f,A \leftarrow \mathcal{F}} [\textsc{Adv}_k(A, p(f))] \neq k^{-\omega(1)}$ $\mathop{\mathbf{E}}_{f,A \leftarrow \mathcal{F}} \left[\textsc{Adv}_k(S^{f,A}, f)\right] = k^{-\omega(1)}$
Strong Semi bb	$\underset{\text{of}}{\exists} p \underset{\text{pol}}{\forall} A \underset{\text{pol}}{\exists} S \forall f$: A^f breaks $p(f) \Rightarrow$ S^f breaks f	$\underset{\text{of}}{\forall} p \underset{\text{pol}}{\exists} A \underset{\text{pol}}{\forall} S \exists \mathcal{F}$: $\mathop{\mathbf{E}}_{f \leftarrow \mathcal{F}} \left[\textsc{Adv}_k(A^f, p(f))\right] = 1 - k^{-\omega(1)}$ $\mathop{\mathbf{E}}_{f \leftarrow \mathcal{F}} \left[\textsc{Adv}_k(S^f, f)\right] = k^{-\omega(1)}$	$\underset{\text{of}}{\forall} p \underset{\text{pol}}{\exists} A \underset{\text{pol}}{\forall} S \exists \mathcal{F}$: $\mathop{\mathbf{E}}_{f \leftarrow \mathcal{F}} \left[\textsc{Adv}_k(A^f, p(f))\right] \neq k^{-\omega(1)}$ $\mathop{\mathbf{E}}_{f \leftarrow \mathcal{F}} \left[\textsc{Adv}_k(S^f, f)\right] = k^{-\omega(1)}$
Weak Semi bb	$\underset{\text{of}}{\exists} p \underset{\text{pol}}{\forall} A \forall f \underset{\text{pol}}{\exists} S$: A^f breaks $p(f) \Rightarrow$ S^f breaks f	$\underset{\text{of}}{\forall} p \underset{\text{pol}}{\exists} A \exists \mathcal{F}$: $\mathop{\mathbf{E}}_{f \leftarrow \mathcal{F}} \left[\textsc{Adv}_k(A^f, p(f))\right] = 1 - k^{-\omega(1)}$ $\underset{\text{pol}}{\forall} S \mathop{\mathbf{E}}_{f \leftarrow \mathcal{F}} \left[\textsc{Adv}_k(S^f, f)\right] = k^{-\omega(1)}$ Countability argument for S	$\underset{\text{of}}{\forall} p \underset{\text{pol}}{\exists} A \exists \mathcal{F}$: $\mathop{\mathbf{E}}_{f \leftarrow \mathcal{F}} \left[\textsc{Adv}_k(A^f, p(f))\right] \neq k^{-\omega(1)}$ $\underset{\text{pol}}{\forall} S \underset{\text{of}}{\forall} \varphi \mathop{\mathbf{E}}_{f \leftarrow \mathcal{F}} \left[\textsc{Adv}_k(S^f_{\varphi(f)}, f)\right] = k^{-\omega(1)}$
Variable Semi bb	$\forall f \underset{\text{pol}}{\exists} P \underset{\text{pol}}{\forall} A \underset{\text{pol}}{\exists} S$: A^f breaks $P^f \Rightarrow$ S^f breaks f	$\exists \mathcal{F} \underset{\text{pol}}{\forall} P \underset{\text{pol}}{\exists} A$: $\mathop{\mathbf{E}}_{f \leftarrow \mathcal{F}} \left[\textsc{Adv}_k(A^f, P^f)\right] = 1 - k^{-\omega(1)}$ $\underset{\text{pol}}{\forall} S \mathop{\mathbf{E}}_{f \leftarrow \mathcal{F}} \left[\textsc{Adv}_k(S^f, f)\right] = k^{-\omega(1)}$ Countability arguments for P and S	$\underset{\text{of}}{\forall} \psi \exists \mathcal{F} \underset{\text{pol}}{\forall} P \underset{\text{pol}}{\exists} A$: $\mathop{\mathbf{E}}_{f \leftarrow \mathcal{F}} \left[\textsc{Adv}_k(A^f, P^f_{\psi(f)})\right] \neq k^{-\omega(1)}$ $\underset{\text{pol}}{\forall} S \underset{\text{of}}{\forall} \varphi \mathop{\mathbf{E}}_{f \leftarrow \mathcal{F}} \left[\textsc{Adv}_k(S^f_{\varphi(f)}, f)\right] = k^{-\omega(1)}$

CHAPTER 6

LOWER BOUNDING SECURITY LOSS

This chapter describes the work presented by the author of this thesis and his co-authors in "Efficiency bounds for adversary constructions in black-box reductions" and "Optimally tight security proofs for hash-then-publish time-stamping" [4, 8] which deals mainly with the efficiency of security proofs.

In the previous chapter, we introduced the assumption of poly-preservedness to the adversarial constructions to be able to show stronger results in the non-uniform model. It requires that the advantage of the adversary construction for $\mathcal{P}$ be a polynomial of the advantage of the adversary for $\mathcal{Q}$ that is used in the reduction, i.e., that the success probability does not drop too much within the adversary construction.

We note that at first thought, it would seem that this constraint could be used to show bounds on reduction efficiency. If one managed to prove that reductions for which the advantage drops by less than a power of c cannot exist under certain oracles, he would be able to lower bound the reduction efficiency of the security proof in the sense of success probability. Quick reflection reveals that this might not be the most sensible thing to do, however, as success probability of adversaries can usually be strongly amplified by just running them multiple times in succession and, as such, meaningful bounds in that respect would seem quite improbable.

However, this problem could fairly easily be remedied by also taking into account the running-time of the adversaries. Instead of considering purely the advantage, it makes more sense to consider the time-success ratio of the adversary, i.e., the expected running-time $\text{TIME}_k(\mathsf{A}, f)$ of A on breaking f divided by its success probability $\text{ADV}_k(\mathsf{A}, f)$. This measure is better suited for bounding as it essentially reflects the amount of time (on average) it takes to break the primitive f by repeatedly calling A until it finally succeeds. This measure is robust against the generic success amplification technique and as such, one would hope that meaningful bounds in terms of that measure should be feasible. To give the definition further generality, we actually allow the construction access to some additional (non-black box) information - namely a bound on the success probability

of the underlying adversary A for P used in the security reduction. This results in the following definition for power-c success-specific reductions which can be seen as a generalization of security-preserving reductions as defined in Luby [32].

Definition 15. We say that there exists a power c-secure *success-specific* black-box reduction from primitive $\mathcal{P}$ to primitive $\mathcal{Q}$ iff there is a poly-time oracle machine P and a polynomial p such that for every $\delta > 0$ there is a poly-time oracle machine S_δ so that for all A such that $\textsc{Adv}_k(\mathsf{A}, \mathsf{P}^f) \geq \delta$ the following conditions hold:

1. For any function f that implements $\mathcal{Q}$, the function P^f implements $\mathcal{P}$.

2. For any pair (A, f) of functions we have

$$\frac{\textsc{Time}_k(\mathsf{S}_\delta^{\mathsf{A},f}, f)}{\textsc{Adv}_k(\mathsf{S}_\delta^{\mathsf{A},f}, f)} \leq p(k) \cdot \left[\frac{\textsc{Time}_k(\mathsf{A}, \mathsf{P}^f)}{\textsc{Adv}_k(\mathsf{A}, \mathsf{P}^f)}\right]^c \quad .$$

   for sufficiently large values of k.

We note that the existence of a power-c reduction implies the existence of a poly-preserving reduction. As only fully black-box reductions are considered[1], all the results henceforth mentioned can easily be seen to hold in the non-uniform model by following the methodology outlined in the previous chapter.

The main result of the first paper [4] is the following meta-theorem for proving lower bounds on c for power-c reductions.

Theorem 10. *If for every pair (S, P) of poly-time oracle machines and for every $\delta > 0$ there is a probability distribution $(\mathsf{A}, f) \leftarrow \Omega_{\mathsf{S},\mathsf{P},\delta}$ so that:*

- *f implements $\mathcal{Q}$ and P^f implements $\mathcal{P}$ for every (A, f) in the range of $\Omega_{\mathsf{S},\mathsf{P},\delta}$;*

- *$\textsc{Adv}_k(\mathsf{A}, \mathsf{P}^f) = \delta$ and $\textsc{Time}_k(\mathsf{A}, \mathsf{P}^f) = O(k^{c_0})$ for some c_0 for all (A, f) in the range of $\Omega_{\mathsf{S},\mathsf{P},\delta}$;*

- *for every polynomial $q(k)$ there exists $\delta(k)$ such that $\lim_{k\to\infty} \delta(k) = 0$ and:*

$$\lim_{k\to\infty} \frac{\delta(k)^c}{q(k)} \cdot \frac{\mathop{\mathsf{E}}_{(\mathsf{A},f)\leftarrow\Omega_{\mathsf{S},\mathsf{P},\delta(k)}} \left[\textsc{Time}_k(\mathsf{S}^{\mathsf{A},f}, f)\right]}{\mathop{\mathsf{E}}_{(\mathsf{A},f)\leftarrow\Omega_{\mathsf{S},\mathsf{P},\delta(k)}} \left[\textsc{Adv}_k(\mathsf{S}^{\mathsf{A},f}, f)\right]} > 1 \quad ,$$

then there are no power c-secure success-specific black-box reductions of $\mathcal{P}$ to $\mathcal{Q}$.

[1] Formally, that is not quite correct, due to the increased generality from success-specificity. However, this is straightforward to account for in the argumentation.

The theorem may seem overly technical at the first glance. A simpler statement was initially considered, but sadly it lacked the generality that was required to prove actual lower bounds. Most of the seeming complexity (such as the introduction of δ) in the theorem statement is there because of practical considerations so that the theorem would actually allow us to prove non-trivial lower bounds.

Practical considerations dictate another restriction as well. Namely, it is hard to fulfill the assumptions of Theorem 10 without assuming any sort of time-uniformity from the security reduction. As such, we formulated a fairly mild notion of it called oracle-independence, which just requires the running time to fluctuate with a sub-exponential magnitude relative to different input adversaries.

Definition 16. We say that the reduction is *oracle-independent* if the running time $\textsc{Time}_k(\mathsf{S}_\delta^{\mathsf{A},f}, f)$ of S is between $m(k,\delta)$ and $u(k)m(k,\delta)$ for all oracles A_k that achieve an advantage of δ against P^f where m is polynomial and $u(k) = 2^{o(k)}$ and does not depend on δ.

To show that these techniques can indeed be used, we demonstrated two cases. Firstly, as a simple illustrative example, we showed the impossibility of a sub-linear self-reduction from one-wayness to collision-resistance, thus proving the optimality of the known folklore reduction in which the collision-resistance adversary simply randmly chooses an input x and then runs the one-wayness adversary to invert $h(x)$.

Of more interest is the second result that was proved, which demonstrated the impossibility of power-c reductions from division-resistance to collision resistance for $c < 1.5$.

Definition 17. A cryptographic 2-1 hash function $h = \{h_k\}$ is said to be *division-resistant* if for every poly-time adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ the following probability

$$\Pr\left[r \leftarrow \{0,1\}^{p(k)}, y \leftarrow \mathsf{A}_1(r), x_1 \leftarrow \{0,1\}^k, x_2 \leftarrow \mathsf{A}_2(y, x_1) \colon h_k(r, x_1 \| x_2) = y\right]$$

is negligible.

Theorem 11. *Let $h = \{h_k\}$ be a cryptographic 2-1 hash function that is collision-resistant. Then for $c < 1.5$ there exist no power c-secure success-specific oracle-independent (possibly non-uniform) black-box reductions $\mathsf{S}^{\mathsf{A},f}$ showing that h is also division-resistant.*

This result is important for two reasons. Firstly, it shows the applicability of this approach to even fairly non-trivial cases, where the lower bound for c is not 1 nor even 2. Secondly, the result actually has far-reaching practical implications for tree-based time-stamping schemes, which were discussed in [8].

Division-resistance (which is basically security against random chosen prefix collisions) in itself is a fairly non-standard security requirement for a hash function. However, it is essentially the bare minimum that is required to construct

a secure bounded hash tree based time-stamping scheme, since any adversary that can break division-resistance can (nearly trivially) be converted to one that breaks bounded hash tree based time-stamping schemes[2]. This means that there is an inherent lower bound of $c = 1.5$ for security reductions that prove security of bounded tree-based time-stamping schemes based purely on the collision-resistance assumption of the used hash function.

The next question any theoretician would ask at this point is whether the lower bound is tight, i.e., whether a power-1.5 reduction actually exists. This question is answered affirmatively in [8].

It is fairly easy to verify that the bounded hash-tree based time-stamping scheme described in Chapter 3 easily fits into this framework.

The security criterion for time-stamping that is presented in this paper is somewhat different from the one that can be derived directly from Definition 10.

Definition 18. A time-stamping scheme is secure if for an unpredictable $(\mathsf{A}_1, \mathsf{A}_2)$:

$$\Pr\left[(r,a) \leftarrow \mathsf{A}_1(1^k), (x,c) \leftarrow \mathsf{A}_2(r,a) \colon \mathsf{Ver}(x,c,r) = 1\right] = k^{-\omega(1)} \quad . \qquad (6.1)$$

where unpredictability means that the output x of A_2 is impossible to predict with non-negligible probability by a PTM, even with full information about the internals and the output of A_1.

In order to prove a power-1.5 reduction to collision resistance, all that one has to assume about the time-stamping scheme is that the certificates have only a limited number N of different possible shapes ρ and if one sees two different certificates c_1 and c_2 that have the same shapes $\rho(c_1) = \rho(c_2)$ then a collision can be found for the hash function that the security is being reduced to.

Definition 19. A time-stamping scheme is said to exhibit the *collision-extraction property* if, whenever $\mathsf{Ver}^h(x_1, c_1, r) = \mathsf{Ver}^h(x_2, c_2, r) = 1$, $\rho(c_1) = \rho(c_2)$, and $(x_1, c_1) \neq (x_2, c_2)$, then the h-calls of $\mathsf{Ver}^h(x_i, c_i, r)$ $(i = 1, 2)$ comprise an h-collision.

If one considers the shape of the hash chain of the certificate as "shape" in the previous definition, it is easy to verify[3] that bounded tree-based time-stamping schemes indeed have that property. However, there may be more complicated schemes that also have the same property and as such, the claim is somewhat more general.

Under these assumptions, it is fairly easy and straightforward to show by standard methods that if time-stamping adversary runs in time t and succeeds with probability δ then there exists a construction that breaks collision-resistance which

[2]Division resistance is equivalent to breaking security at the leaves by switching out one of the two inputs to a hash box while leaving the rest of the tree unaltered.

[3]With reasoning completely analogous to the security reduction in Theorem 1.

runs in time $t' \approx 2t$ and succeeds with probability $\delta' = \frac{\delta^2}{N}$, which constitutes a power-2 reduction since $\frac{t'}{\delta'} \approx 2N\frac{t}{\delta^2}$. Best reduction known in the previous literature is that of Buldas and Laur [5] where they showed $\frac{t'}{\delta'} \approx 48\sqrt{N}\frac{t}{\delta^2}$ – improving the reduction considerably in terms of N but leaving it essentially the same in terms of the time-success power ratio. This meant that in the currently known literature, power 1.5 had not been achieved yet.

Nevertheless, such a reduction turned out to be possible. By a precise combinatorial analysis, it was possible to demonstrate that a reduction exists for which the security loss can be described by the formula $\frac{t'}{\delta'} \approx 14\sqrt{N}\frac{t}{\delta^{1.5}}$, presenting a major increase in the reduction increase over the state of the art and also achieving the provably optimal power ratio of 1.5. Efficiency gains here turned out to actually have practical consequences, as they now allow realistic security guarantees for practical global-sized time-stamping services (with N in the order of 2^{56}) while still using only moderately sized hash functions of 256 bit outputs – something that the previous reductions failed to demonstrate but which is crucial for the security of some current implementations already in use (such as GuardTime).

CHAPTER 7

DETERMINISTIC RANDOM ORACLES

This chapter aims to give an overview of the results presented by the author in the paper "Deterministic Random Oracles" [36].

As noted in Chapter 5, one of the main obstacles to generalizing oracle separation results is the fact that oracle families are used instead of a single fixed oracle to introduce randomness into the model. This, however, poses problems when one tries to show the non-existence of separations. In such a case one would normally have to find one fixed oracle relative to which the result holds. If one starts out with a family of oracles, fixing one of them usually involves non-constructive arguments that make heavy use of the countability of the sets involved. Sadly, many of them do not work in the non-uniform model.

Looking at the separation result presented by Gennaro and Trevisan [18] that was the first one proven in the non-uniform model, it becomes apparent that the main difficulty for direct proofs in the non-uniform model is being able to demonstrate the security of an implementation f of the underlying primitive $\mathcal{Q}$ which is usually constructed based on the oracle. To achieve it, they make use of combinatorial arguments over a fixed output length k[1] that basically argue that if A^f can invert $f : \{0,1\}^k \to \{0,1\}^k$ then f has to be chosen from a relatively small number of choices. As the number of one-way functions (or permutations) is considerably larger, any adversary circuit A for that input length can thus be successful on only a small fraction of them. As there is also a limited number of circuits, this is sufficient to show that there exist functions that are secure against all circuits and this happens with larger and larger probability as k increases.

This argument is somewhat reminiscent of the types of proofs done in the field of algorithmic information theory, which is built around the notion of Chaitin-Kolmogorov-Solomonoff complexity. What is interesting, however, is that this theory provides a deterministic notion of randomness. We were interested in whether this could be exploited to yield an alternative model for random oracles that does not rely on a truly random oracle family but rather assumes a fixed oracle

[1] They use the size of the output set N instead of security parameter k. For our purposes, we just define $N = 2^k$.

which nonetheless possesses all the properties commonly associated with random oracles on average. If that were the case, it would perhaps allow one to bypass the oracle extraction step in the separations by assuming a fixed oracle from the beginning.

The main concept in Algorithmic Information Theory (AIT) is that of (self-delimiting or Chaitin) complexity $H(x)$ of a finite bitstring x, first introduced by Levin [30] but popularized by Chaitin [11]. To be precise, $H(x)$ is defined as the length $|x'|$ of the shortest program and input pair $x' = (p, i)$ such that $\mathsf{U}(x') = x$ where U is some fixed (self-delimiting)[2] Universal Turing Machine.

Although deterministic randomness can be defined in many different ways, the most well known of them is probably the following characterization based on Chaitin complexity of finite strings.

Definition 20. A bit sequence $\mathbf{x} \in \{0,1\}^\omega$ is said to be *algorithmically random* when there exists a constant $c_{\mathbf{x}}$ (the *randomness threshold*) such that all of its prefixes $\mathbf{x}_n$ have complexity $H(\mathbf{x}_n) \geq n + c_{\mathbf{x}}$.

Such sequences are analogous to truly random sequences in many ways. As such, it makes sense to define the Algorithmically Random Oracle Model as just replacing the "truly" random oracle with an oracle $\mathcal{O}_{\mathbf{r}} : \{0,1\}^* \to \{0,1\}$ based on such an algorithmically random bit sequence $\mathbf{r}$. In the simplest case, this is achieved by assuming the natural bijection between $\mathbb{N}$ and $\{0,1\}^*$ and setting $\mathcal{O}_{\mathbf{r}}(j)$ equal to the j-th bit of $\mathbf{r}$.

Our first step was, of course, to prove that such algorithmically random oracles possess at least some commonalities with the standard random oracles. To do that, we show that that they can be used to construct very strong one-way functions in the trivial way by defining the one-way function $\{f_k | k \in \mathbb{N}\}$ as $f_n^{\mathbf{r}}(i) = \mathbf{r}_{ni,\ldots,n(i+1)-1}$ (i.e. as the ni-th to $n(i+1)-1$-th bits of $\mathbf{r}$).

Theorem 12. *Let* $\mathbf{r}$ *be an algorithmically random sequence and let* $f_n^{\mathbf{r}}$ *be defined as before. If* $m = m(n)$ *is such that* $m \geq n$ *then* $\{f_n^{\mathbf{r}} | n \in \mathbb{N}\}$ *is a one-way function (secure even in the non-uniform model).*

We also demonstrate that a one-way permutation oracle can be constructed based on $\mathbf{r}$ in an analogous way. Both proofs are fairly simple, presenting a blend of techniques from cryptography and AIT. In short, they show that if an adversary existed, it would imply a shorter description of $\mathcal{O}$ that should be possible, thus refuting the assumption that $\mathbf{r}$ is random. In some sense, they can be seen as formalizing the idea of [18] but within a different framework which makes generalizations potentially simpler.

Both results hold with respect to *any* algorithmically random bitstring $\mathbf{r}$. This has interesting implications as there is a well-known result that states randomly

[2] Self-delimiting string is a string which encodes its own length in some way so that on parsing it the point where it ends can readily be verified. This assumption is essential for most of the results in modern AIT but the exact reasons for it are beyond the scope of this chapter.

chosen bit sequence $\mathbf{x} \in \{0,1\}^\omega$ is algorithmically random with probability 1. This means that anything proven in AROM (i.e. proven relative to an algorithmically random oracle without extra assumptions) will also be true in the standard ROM. More interestingly, however, we prove that the converse also holds, i.e. that under reasonable assumptions, anything proven secure in the standard ROM will also be secure relative to any AROM.

Theorem 13. *Assume that a construction $\mathsf{C}^{\mathcal{O}}$ is an instance of some primitive $\mathcal{P}$ (that has a computable security criterion[3] w.r.t. a random oracle $\mathcal{O}$. Then $\mathsf{C}^{\mathcal{O}}$ is secure in ROM precisely when $\mathsf{C}^{\mathbf{r}}$ is secure for all algorithmically random bit sequences* $\mathbf{r}$.

Our work also shows how one can use the tools of AIT to prove security-related results. This introduces a new and potentially very powerful theoretical tool into the domain of Cryptology. We can only hope that our work will be followed up by others who will strengthen that connection so that it may prove beneficial for both the AIT and cryptology communities.

[3]Computability of the security criterion is an essentially technical restriction stating that the success probability of an adversary has to be computable for all poly-time adversaries.

CONCLUSIONS AND FUTURE RESEARCH

In this work we have explored the limits of the oracle separation technique in many different ways. Initially, we were concerned with just one concrete separation – from chain-resistance to collision-resistance – and explored just one possible oracle which seemed to have implications for that case. Later work was concerned with more general themes, from trying to determine whether the already known results could be generalized to the non-uniform model to seeing what other meaningful lower bounds could be proved, to even describing an alternative models of oracles that would allow one to replace a random oracle family with just one, "deterministically random" oracle.

Nevertheless, there are many open questions that remain in the field of oracle separation. The question that is by far of most interest to the author is whether the results presented in the paper discussing deterministic randomness could be built upon to prove separation results. It would also be interesting to see, what other implications the "algorithmically random oracle model" would have.

The work on lower bounds has gained increasing popularity over the past decade [29, 18, 16, 2, 17, 24, 31, 4]. Although most of this work has concentrated on lower-bounding the number of invocations of the underlying primitive, there are also other things that can be lower-bounded as has been demonstrated in our work with the security loss of reduction. There may well be more parameters of reductions for which meaningful bounds could be obtained and it would indeed be interesting to see any such new types of bounds being introduced and proven.

Socrates is often quoted saying "I know only that I know nothing". Although things are not quite that bad with oracle separation, it is clear that there is much to still discover about the technique. If anything, this thesis has showed us that by pointing to new and interesting possible directions of inquiry. However, this is the nature of science – to pose questions by answering previous ones. In that sense, we hope that the thesis has served its purpose.

Bibliography

[1] Baker, T.P., Gill, J., Solovay, R.: Relativizatons of the $\mathbf{P}$ =? $\mathbf{NP}$ question. SIAM J. Comput. 4(4), 431–442 (1975)

[2] Black, J., Cochran, M., Shrimpton, T.: On the impossibility of highly efficient blockcipher-based hash functions. In: Advances in Cryptology – EUROCRYPT 2005. pp. 526–541 (2005)

[3] Buldas, A., Jürgenson, A.: Does secure time-stamping imply collision-free hash functions? In: The 1st International Conference on Provable Security (ProvSec) 2007. LNCS, vol. 4784, pp. 138–150. Springer (2007)

[4] Buldas, A., Jürgenson, A., Niitsoo, M.: Efficiency bounds for adversary constructions in black-box reductions. In: Australian Conference on Information Security and Privacy – ACISP 2009. LNCS, vol. 5594, pp. 264–275. Springer (2009)

[5] Buldas, A., Laur, S.: Knowledge-binding commitments with applications in time-stamping. In: The International Conference on Theory and Practice of Public-Key Cryptography – PKC 2007. LNCS, vol. 4450, pp. 150–165. Springer (2007)

[6] Buldas, A., Laur, S., Niitsoo, M.: Oracle separation in the non-uniform model. In: The 3rd International Conference on Provable Security (ProvSec) 2009. LNCS, vol. 5848, pp. 230–244. Springer (2009)

[7] Buldas, A., Niitsoo, M.: Can we construct unbounded time-stamping schemes from collision-free hash functions? In: The 2rd International Conference on Provable Security (ProvSec) 2008. LNCS, vol. 4784, pp. 254–267. Springer (2008)

[8] Buldas, A., Niitsoo, M.: Optimally tight security proofs for hash-then-publish time-stamping. In: Australian Conference on Information Security and Privacy – ACISP 2010. LNCS, vol. 6168, pp. 318–355. Springer (2010)

[9] Buldas, A., Niitsoo, M.: Black-box separations and their adaptability to the non-uniform model (2011), unpublished

[10] Buldas, A., Saarepera, M.: On provably secure time-stamping schemes. In: Advances in Cryptology – Asiacrypt 2004. LNCS, vol. 3329, pp. 500–514. Springer (2004)

[11] Chaitin, G.J.: A theory of program size formally identical to information theory. Journal of the ACM 22(3), 329–340 (1975)

[12] Chang, Y., Hsiao, C., Lu, C.: On the impossibilities of basing one-way permutations on central cryptographic primitives. In: Advances in Cryptology – ASIACRYPT 2002. pp. 110–124. Springer-Verlag (2002)

[13] Damgård, I.: A design principle for hash functions. In: Advances in Cryptology – CRYPTO 1989. pp. 416–427 (1989)

[14] Dodis, Y., Oliveira, R., Pietrzak, K.: On the generic insecurity of the full domain hash. In: Advances in Cryptology – CRYPTO 2005. pp. 449–466 (2005)

[15] Dodis, Y., Reyzin, L.: On the power of claw-free permutations. In: Security in Communication Networks, Third International Conference, SCN 2002. pp. 55–73 (2002)

[16] Gennaro, R., Gertner, Y., Katz, J.: Lower bounds on the efficiency of encryption and digital signature schemes. In: Proceedings of the 35th Annual ACM Symposium on Theory of Computing. pp. 417–425 (2003)

[17] Gennaro, R., Gertner, Y., Katz, J., Trevisan, L.: Bounds on the efficiency of generic cryptographic constructions. SIAM Journal of Computing 35(1), 217–246 (2005)

[18] Gennaro, R., Trevisan, L.: Lower bounds on the efficiency of generic cryptographic constructions. Electronic Colloquium on Computational Complexity (ECCC) 7(22) (2000)

[19] Gertner, Y., Kannan, S., Malkin, T., Reingold, O., Viswanathan, M.: The relationship between public key encryption and oblivious transfer. In: FOCS '00: Proceedings of the 41st Annual Symposium on Foundations of Computer Science. pp. 325–337. IEEE Computer Society (2000)

[20] Gertner, Y., Malkin, T., Reingold, O.: On the impossibility of basing trapdoor functions on trapdoor predicates. In: FOCS '01: Proceedings of the 42nd IEEE symposium on Foundations of Computer Science. pp. 126–135. IEEE Computer Society (2001)

[21] Gertner, Y., Malkin, T., Myers, S.: Towards a separation of semantic and cca security for public key encryption. In: Theory of Cryptography Conference – TCC 2007. pp. 434–455 (2007)

[22] Haber, S., Stornetta, W.S.: How to time-stamp a digital document. Journal of Cryptology 3(2), 99–111 (1991)

[23] Haber, S., Stornetta, W.S.: Secure names for bit-strings. In: ACM Conference on Computer and Communications Security. pp. 28–35 (1997)

[24] Horvitz, O., Katz, J.: Bounds on the efficiency of "black-box" commitment schemes. In: ICALP 2005. pp. 128–139 (2005)

[25] Hsiao, C.Y., Reyzin, L.: Finding collisions on a public road, or do secure hash functions need secret coins? In: Advances in Cryptology – CRYPTO 2004. pp. 92–105 (2004)

[26] Impagliazzo, R.: Very strong one-way functions and pseudo-random generators exist relative to a random oracle. Manuscript (1996)

[27] Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: Proceedings of 21st Annual ACM Symposium on the Theory of Computing. pp. 44–61 (1989)

[28] Kiltz, E., Pietrzak, K.: On the security of padding-based encryption schemes — or — why we cannot prove OAEP secure in the standard model. In: Advances in Cryptology – EUROCRYPT 2009. pp. 389–406. Springer-Verlag (2009)

[29] Kim, J.H., Simon, D.R., Tetali, P.: Limits on the efficiency of one-way permutation-based hash functions. In: FOCS: Proceedings of the 40th Annual Symposium on Foundations of Computer Science. pp. 535–542 (1999)

[30] Levin, L.A.: Laws of information conservation (nongrowth) and aspects of the foundation of probability theory. Probl. Peredachi Inf. 10(3), 30–35 (1974)

[31] Lu, C.J.: On the security loss in cryptographic reductions. In: Advances in Cryptology – EUROCRYPT 2009. pp. 72–87. Springer-Verlag (2009)

[32] Luby, M.G.: Pseudorandomness and Cryptographic Applications. Princeton University Press, Princeton, NJ, USA (1994)

[33] Merkle, R.C.: A certified digital signature. In: Advances in Cryptology – CRYPTO 1989. pp. 218–238 (1989)

[34] Naor, M., Yung, M.: Universal one-way hash functions and their cryptographic applications. In: Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing. pp. 33–43 (1989)

[35] Niitsoo, M.: Optimal adversary behavior for the serial model of financial attack trees. In: International Workshop on Security – IWSEC 2010. LNCS, vol. 6434, pp. 354–370. Springer (2010)

[36] Niitsoo, M.: Deterministic random oracles (2011), unpublished

[37] Reingold, O., Trevisan, L., Vadhan, S.: Notions of reducibility between cryptographic primitives. In: Theory of Cryptography Conference – TCC 2004. LNCS, vol. 2951, pp. 1–20. Springer (2004)

[38] Rogaway, P., Shrimpton, T.: Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In: Fast Software Encryption. LNCS, vol. 3017, pp. 371–388. Springer (2004)

[39] Simon, D.R.: Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In: Advances in Cryptology – EUROCRYPT 1998. pp. 334–345 (1998)

[40] Unruh, D.: Random oracles and auxiliary input. In: Advances in Cryptology – CRYPTO 2007. LNCS, vol. 4622, pp. 205–223. Springer (2007)

ACKNOWLEDGMENTS

The author is grateful to his supervisor, Ahto Buldas, for first giving him a masters thesis topic and then allowing him to continue with it into his PhD studies, always encouraging him whenever he needed it. He is also very thankful to his other coauthors Aivo Jürgenson and Sven Laur for many interesting discussions that have takan place over the past three years. He would also like to thank all his other colleagues and friends and especially his mother for supporting him throughout his studies.

The author has been partly financially supported by the European Regional Development Fund through the Estonian Center of Excellence in Computer Science, by the Estonian Science Foundation grant no. 6944, by EU FP6-15964: “AEOLUS", the Estonian Doctoral School in ICT, by the Tiger University Program of the Estonian Information Technology Foundation and, last but not least, Cybernetica AS, one of the few private research companies in Estonia, where he has also been employed as a junior researcher for 2 and a half years.

KOKKUVÕTE
(SUMMARY IN ESTONIAN)

ORAAKLIGA MUSTA KASTI ERALDUSTEHNIKAD RAKENDUSTEGA AJATEMBELDUSELE

Krüptoloogia on teadus turvalisest andmesidest. Tõestatava turvalisuse saavutamiseks tuleb aga reeglina teha eeldusi. Lihtsaimatel juhtudel piirduvad need eeldused lihtsalt turvatava süsteemi kirjeldusega. Reeglina tuleb protokolli turvalisuse näitamiseks teha aga ka arvutuslikke eeldusi. Näiteks piisab tihti eeldusest, et suurte arvude tegurdamine on arvutuslikult raske ning ajamahukas. Vahel on aga lihtsam teha natuke üldisemaid eelduseid, eeldades juba mõne teist tüüpi turvalise süsteemi olemasolu. Näiteks on võimalik praktiliselt kogu salajase võtmega krüptograafia ehitada üles pelgalt eeldusest, et eksisteerib vähemalt üks raskesti pööratav ühesuunaline funktsioon.

Sellised teise süsteemi turvalisuse eeldusel põhinevad tõestused on reeglina väga lihtsa struktuuriga. Kõigepealt näidatakse, kuidas seda teist süsteemi saab kasutada alamkomponendina käesoleva süsteemi realiseerimisel. Turvalisuse tõestamiseks näidatakse seejärel, kuidas käesoleva süsteemi murtavusest järeldub automaatselt ka komponendina kasutatud süsteemi murtavus. See annab tõepoolest turvalisuse tõestuse, sest kuna me eeldame, et teist süsteemi murda ei saa, ei saa järelikult murda ka käesolevat. Selliseid turvatõestuseid nimetatakse üldiselt reduktsioonideks ning enamus tänapäevasest krüptograafiast on just nende najal üles ehitatud.

Enamasti kasutatakse reduktsioonides teist süsteemi väga lihtsal moel: eeldatakse, et on olemas mingi turvaline selle teise süsteemi implementatsioon millele on niiöelda *musta kasti* juurdepääs – sellele saab anda sisendeid ning see annab väljundeid, kuid seal sees toimuva kohta igasugune lisainfo puudub. Turvatões-

tuse poole peal tehakse aga vastuväiteline eeldus, et uus süsteem on murtav ning seega on meie käsutuses musta kastina ka mõni seda murda oskav vastane. Seda vastast kasutades konstrueeritakse siis vastane teisele süsteemile, mille turvalisust eeldati. Sellised musta kasti reduktsioonid on oma struktuurilt kõige lihtsamad ning praktiliselt kõik reduktsioonid vastavad tegelikult ka nendele eeldustele.

Musta kasti reduktsioonide abil on võimalik tõestada väga huvitavaid tulemusi, kuid nende kasutamisel on siiski teatavad piirid. Juba üle 20 a tagasi näitasid Rudich ja Impagliazzo [27], et ühesuunalise funktsiooni olemasolu ei ole piisav niinimetatud salajase võtme krüptograafia realiseerimiseks ainult musta kasti reduktsioone kasutades. Seda tõestasid nad niinimetatud *oraakliga eralduse* meetodi abil, rakendades ühesuunalise funktsiooni rollis väga kavalat enda poolt konstrueeritud musta kasti, mille enamus väljundeid on valitud ühtlaselt ja juhuslikult, kuid mille teatud sisendite korral vastab ta muidu väga raskesti vastatavatele, klassi **PSPACE** kuuluvatele küsimustele. Ligipääs sellisele *oraaklile* tekitab ühelt poolt olukorra, kus meie konstrueeritud funktsiooni juhuslikult valitud osa annab pea ideaalse ühesuunalise funktsiooni, kuid kus teiselt poolt $\mathbf{P} = \mathbf{NP}$ mille tõttu avaliku võtme krüptograafia kindlasti välistatud on. Kuigi selline oraakel on võimalik vaid teoreetiliselt, välistab see ometi musta kasti reduktsiooni võimalikkuse. Seda tüüpi reduktsioon peaks ju töötama ka sellist konstrueeritud oraaklit musta kastina kasutades, kuna kasti kohta ei tehta tõestuses lisaeelduseid.

Selline lähenemine osutus väga viljakaks, ning seda arendati peagi edasi teiste autorite poolt. Näiteks tõestasid Kim, Simon ja Tetali [29] seda meetodit kasutades alampiiri sellele, kui mitu korda peab turvalise räsifunktsiooni saavutamiseks ühesuunalist funktsiooni kasutades toda ühesuunalist funktsiooni välja kutsuma – näidati seega mitte reduktsiooni võimatust, vaid selle põhimõttelist efektiivsuse piiri. Igal juhul on oraakliga eralduse näol tegemist praeguseks laialt levinud tõestusmeetodiga [27, 39, 29, 18, 19, 20, 12, 16, 38, 25, 14, 17, 24, 21, 28, 31].

Selles doktoritöös uuritakse oraakliga eralduse meetodi erinevaid uusi rakendusvõimalusi. Töö põhineb neljal avaldatud ja kahel avaldamata artiklil.

Neist esimene artikkel „Kollisioonivabadel räsifunktsioonidel põhinevate piiranguteta ajatempliskeemide võimalikkusest',,vaatleb algselt Hsiao ja Reyzini [25] poolt kasutatud kahe oraakliga eralduse raamistikku, laiendab seda veidi ning kasutab seda seejärel et näidata reduktsiooni võimatust kollisioonivabadusest piiratud ajatempliskeemini teatud lisakitsenduste korral, mida võiks tinglikult nimetada ühe väljakutsega geneeriliseks mudeliks. Töö ei välista küll otseselt musta kasti reduktsiooni, kuid viitab siiski tugevalt selle olemasolu ebatõenäolisusele.

Teine ja kolmas artikkel, „Oraakliga eraldamine mitteühtlases mudelis" ning „Musta kasti eraldused ja nende üldistatavus mitteühtlasesse mudelisse", on juba natukene konkreetsemad. Mõlema artikli aluseks on üks huvitav tehniline tähelepanek. Nimelt tõestatakse enamus reduktsioone niinimetatud mitteühtlases mudelis, kus vastane võib omada teatava koguse lisainfot, näiteks eelmistelt murdmiskatsetelt. Enamus eraldustulemusi tõestatakse aga ainult ühtlases mudelis, kus sellist lisainfot ei võimaldata. Antud kaks artiklit uurivad võimalusi, kuidas ju-

CURRICULUM VITAE

Personal data

Name	Margus Niitsoo
Birth	January 7, 1987 Tallinn, Estonia
Citizenship	Estonian
Marital Status	Single
Languages	Estonian, English, Russian, French
Address	Sõbra 12-2, Tartu 5017 Tartu Estonia
Contact	+372 55 60 3840 margus.niitsoo@ut.ee

Education

2008–	University of Tartu, Ph.D. candidate
2008	University of Tartu, M.Sc. cum laude in Computer Science
2005–2008	University of Tartu, B.Sc. cum laude in Mathematics
2002–2005	Tallinn Secondary Science School, secondary education
1994–2002	Tallinn French School, primary education

Employment

2008–	Cybernetica AS, junior researcher (half time)
2008–2010	University of Tartu, Teaching Assistant (half time)
2005–2006	University of Tartu, programmer (quarter time)

ELULOOKIRJELDUS

Isikuandmed

Nimi	Margus Niitsoo
Sünniaeg ja -koht	7. jaanuar 1987 Tallinn, Eesti
Kodakondsus	eestlane
Perekonnaseis	vallaline
Keelteoskus	eesti, inglise, vene, prantsuse
Aadress	Sõbra 12-2, Tartu 5017 Tartu Estonia
Kontaktandmed	+372 55 60 3840 margus.niitsoo@ut.ee

Haridustee

2008–	Tartu Ülikool, doktorant
2008	Tartu Ülikool, MSc cum laude informaatikas
2005–2008	Tartu Ülikool, BSc cum laude matemaatikas
2002–2005	Tallinna Reaalkool, keskharidus
1994–2002	Tallinna Prantsuse Lütseum, põhiharidus

Teenistuskäik

2008–	Cybernetica AS, nooremteadur (0.5 koht)
2008–2010	Tartu Ülikool, Assistent (0.5 koht)
2005–2006	Tartu Ülikool, programmeerija (0.25 koht)

DISSERTATIONES MATHEMATICAE UNIVERSITATIS TARTUENSIS

1. **Mati Heinloo.** The design of nonhomogeneous spherical vessels, cylindrical tubes and circular discs. Tartu, 1991, 23 p.
2. **Boris Komrakov.** Primitive actions and the Sophus Lie problem. Tartu, 1991, 14 p.
3. **Jaak Heinloo.** Phenomenological (continuum) theory of turbulence. Tartu, 1992, 47 p.
4. **Ants Tauts.** Infinite formulae in intuitionistic logic of higher order. Tartu, 1992, 15 p.
5. **Tarmo Soomere.** Kinetic theory of Rossby waves. Tartu, 1992, 32 p.
6. **Jüri Majak.** Optimization of plastic axisymmetric plates and shells in the case of Von Mises yield condition. Tartu, 1992, 32 p.
7. **Ants Aasma.** Matrix transformations of summability and absolute summability fields of matrix methods. Tartu, 1993, 32 p.
8. **Helle Hein.** Optimization of plastic axisymmetric plates and shells with piece-wise constant thickness. Tartu, 1993, 28 p.
9. **Toomas Kiho.** Study of optimality of iterated Lavrentiev method and its generalizations. Tartu, 1994, 23 p.
10. **Arne Kokk.** Joint spectral theory and extension of non-trivial multiplicative linear functionals. Tartu, 1995, 165 p.
11. **Toomas Lepikult.** Automated calculation of dynamically loaded rigid-plastic structures. Tartu, 1995, 93 p, (in Russian).
12. **Sander Hannus.** Parametrical optimization of the plastic cylindrical shells by taking into account geometrical and physical nonlinearities. Tartu, 1995, 74 p, (in Russian).
13. **Sergei Tupailo.** Hilbert's epsilon-symbol in predicative subsystems of analysis. Tartu, 1996, 134 p.
14. **Enno Saks.** Analysis and optimization of elastic-plastic shafts in torsion. Tartu, 1996, 96 p.
15. **Valdis Laan.** Pullbacks and flatness properties of acts. Tartu, 1999, 90 p.
16. **Märt Põldvere.** Subspaces of Banach spaces having Phelps' uniqueness property. Tartu, 1999, 74 p.
17. **Jelena Ausekle.** Compactness of operators in Lorentz and Orlicz sequence spaces. Tartu, 1999, 72 p.
18. **Krista Fischer.** Structural mean models for analyzing the effect of compliance in clinical trials. Tartu, 1999, 124 p.

19. **Helger Lipmaa.** Secure and efficient time-stamping systems. Tartu, 1999, 56 p.
20. **Jüri Lember.** Consistency of empirical k-centres. Tartu, 1999, 148 p.
21. **Ella Puman.** Optimization of plastic conical shells. Tartu, 2000, 102 p.
22. **Kaili Müürisep.** Eesti keele arvutigrammatika: süntaks. Tartu, 2000, 107 lk.
23. **Varmo Vene.** Categorical programming with inductive and coinductive types. Tartu, 2000, 116 p.
24. **Olga Sokratova.** Ω-rings, their flat and projective acts with some applications. Tartu, 2000, 120 p.
25. **Maria Zeltser.** Investigation of double sequence spaces by soft and hard analitical methods. Tartu, 2001, 154 p.
26. **Ernst Tungel.** Optimization of plastic spherical shells. Tartu, 2001, 90 p.
27. **Tiina Puolakainen.** Eesti keele arvutigrammatika: morfoloogiline ühestamine. Tartu, 2001, 138 p.
28. **Rainis Haller.** *M(r,s)*-inequalities. Tartu, 2002, 78 p.
29. **Jan Villemson.** Size-efficient interval time stamps. Tartu, 2002, 82 p.
30. **Eno Tõnisson.** Solving of expession manipulation exercises in computer algebra systems. Tartu, 2002, 92 p.
31. **Mart Abel.** Structure of Gelfand-Mazur algebras. Tartu, 2003. 94 p.
32. **Vladimir Kuchmei.** Affine completeness of some ockham algebras. Tartu, 2003. 100 p.
33. **Olga Dunajeva.** Asymptotic matrix methods in statistical inference problems. Tartu 2003. 78 p.
34. **Mare Tarang.** Stability of the spline collocation method for volterra integro-differential equations. Tartu 2004. 90 p.
35. **Tatjana Nahtman.** Permutation invariance and reparameterizations in linear models. Tartu 2004. 91 p.
36. **Märt Möls.** Linear mixed models with equivalent predictors. Tartu 2004. 70 p.
37. **Kristiina Hakk.** Approximation methods for weakly singular integral equations with discontinuous coefficients. Tartu 2004, 137 p.
38. **Meelis Käärik.** Fitting sets to probability distributions. Tartu 2005, 90 p.
39. **Inga Parts.** Piecewise polynomial collocation methods for solving weakly singular integro-differential equations. Tartu 2005, 140 p.
40. **Natalia Saealle.** Convergence and summability with speed of functional series. Tartu 2005, 91 p.
41. **Tanel Kaart.** The reliability of linear mixed models in genetic studies. Tartu 2006, 124 p.

42. **Kadre Torn.** Shear and bending response of inelastic structures to dynamic load. Tartu 2006, 142 p.
43. **Kristel Mikkor.** Uniform factorisation for compact subsets of Banach spaces of operators. Tartu 2006, 72 p.
44. **Darja Saveljeva.** Quadratic and cubic spline collocation for Volterra integral equations. Tartu 2006, 117 p.
45. **Kristo Heero.** Path planning and learning strategies for mobile robots in dynamic partially unknown environments. Tartu 2006, 123 p.
46. **Annely Mürk.** Optimization of inelastic plates with cracks. Tartu 2006. 137 p.
47. **Annemai Raidjõe.** Sequence spaces defined by modulus functions and superposition operators. Tartu 2006, 97 p.
48. **Olga Panova.** Real Gelfand-Mazur algebras. Tartu 2006, 82 p.
49. **Härmel Nestra.** Iteratively defined transfinite trace semantics and program slicing with respect to them. Tartu 2006, 116 p.
50. **Margus Pihlak.** Approximation of multivariate distribution functions. Tartu 2007, 82 p.
51. **Ene Käärik.** Handling dropouts in repeated measurements using copulas. Tartu 2007, 99 p.
52. **Artur Sepp.** Affine models in mathematical finance: an analytical approach. Tartu 2007, 147 p.
53. **Marina Issakova.** Solving of linear equations, linear inequalities and systems of linear equations in interactive learning environment. Tartu 2007, 170 p.
54. **Kaja Sõstra.** Restriction estimator for domains. Tartu 2007, 104 p.
55. **Kaarel Kaljurand.** Attempto controlled English as a Semantic Web language. Tartu 2007, 162 p.
56. **Mart Anton.** Mechanical modeling of IPMC actuators at large deformations. Tartu 2008, 123 p.
57. **Evely Leetma.** Solution of smoothing problems with obstacles. Tartu 2009, 81 p.
58. **Ants Kaasik.** Estimating ruin probabilities in the Cramér-Lundberg model with heavy-tailed claims. Tartu 2009, 139 p.
59. **Reimo Palm.** Numerical Comparison of Regularization Algorithms for Solving Ill-Posed Problems. Tartu 2010, 105 p.
60. **Indrek Zolk.** The commuting bounded approximation property of Banach spaces. Tartu 2010, 107 p.
61. **Jüri Reimand.** Functional analysis of gene lists, networks and regulatory systems. Tartu 2010, 153 p.
62. **Ahti Peder.** Superpositional Graphs and Finding the Description of Structure by Counting Method. Tartu 2010, 87 p.

63. **Marek Kolk.** Piecewise Polynomial Collocation for Volterra Integral Equations with Singularities. Tartu 2010, 134 p.
64. **Vesal Vojdani.** Static Data Race Analysis of Heap-Manipulating C Programs. Tartu 2010, 137 p.
65. **Larissa Roots.** Free vibrations of stepped cylindrical shells containing cracks. Tartu 2010, 94 p.
66. **Mark Fišel.** Optimizing Statistical Machine Translation via Input Modification. Tartu 2011, 104 p.