Cryptographic Reductions: Classification and Applications to Ideal Models

Vom Fachbereich Informatik der
Technischen Universität Darmstadt genehmigte

Dissertation

zur Erlangung des Grades
Doctor rerum naturalium (Dr. rer. nat.)
von

Paul Baecher, M.Sc.
geboren in Lich

Referenten:	Prof. Dr. Marc Fischlin
	Prof. Dr. Thomas Shrimpton
Tag der Einreichung:	11. August 2014
Tag der mündlichen Prüfung:	2. Oktober 2014

Darmstadt, 2014
D 17

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit – abgesehen von den in ihr ausdrücklich genannten Hilfen – selbständig verfasst habe.

_____

Wissenschaftlicher Werdegang

Oktober 2004 – April 2008

Studium der Informatik (B.Sc.) an der Technischen Universität Darmstadt

Mai 2008 – Januar 2010

Studium der Informatik (M.Sc.) an der Technischen Universität Darmstadt

März 2010 – Januar 2011

Wissenschaftlicher Mitarbeiter in der Emmy-Noether-Forschungsgruppe „MiniCrypt“ an der Technischen Universität Darmstadt.

seit Februar 2011

Wissenschaftlicher Mitarbeiter in der Forschungsgruppe „Kryptoplexität“ an der Technischen Universität Darmstadt.

List of Publications

[1] Paul Baecher, Christina Brzuska, and Marc Fischlin. Notions of black-box reductions, revisited. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 296–315, Bengalore, India, December 1–5, 2013. Springer, Berlin, Germany. **Part of this thesis**.

[2] Paul Baecher, Christina Brzuska, and Arno Mittelbach. Reset indifferentiability and its consequences. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 154–173, Bengalore, India, December 1–5, 2013. Springer, Berlin, Germany.

[3] Paul Baecher, Pooya Farshim, Marc Fischlin, and Martijn Stam. Ideal-cipher (ir)reducibility for blockcipher-based hash functions. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 426–443, Athens, Greece, May 26–30, 2013. Springer, Berlin, Germany. **Part of this thesis**.

[4] Paul Baecher and Marc Fischlin. Random oracle reducibility. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 21–38, Santa Barbara, CA, USA, August 14–18, 2011. Springer, Berlin, Germany. **Part of this thesis**.

[5] Paul Baecher, Marc Fischlin, and Dominique Schröder. Expedient non-malleability notions for hash functions. In Aggelos Kiayias, editor, *Topics in Cryptology – CT-RSA 2011*, volume 6558 of *Lecture Notes in Computer Science*, pages 268–283, San Francisco, CA, USA, February 14–18, 2011. Springer, Berlin, Germany.

[6] Paul Baecher, Niklas Büscher, Marc Fischlin, and Benjamin Milde. Breaking reCAPTCHA: A holistic approach via shape recognition. In Jan Camenisch, Simone Fischer-Hübner, Yuko Murayama, Armand Portmann, and Carlos Rieder, editors, *SEC*, volume 354 of *IFIP Advances in Informa-*

tion and Communication Technology, pages 56–67, Lucerne, Switzerland, June 7–9, 2011. Springer, Berlin, Germany.

[7] Paul Baecher, Marc Fischlin, Lior Gordon, Robert Langenberg, Michael Lützow, and Dominique Schröder. CAPTCHAs: The good, the bad, and the ugly. In Felix C. Freiling, editor, *Sicherheit 2010: Sicherheit, Schutz und Zuverlässigkeit*, volume 170 of *Lecture Notes in Informatics*, pages 353–365, Berlin, Germany, October 5–7, 2010. Gesellschaft für Informatik, Bonn, Germany.

[8] Heike Busch, Stefan Katzenbeisser, and Paul Baecher. PUF-based authentication protocols - revisited. In Heung Youl Youm and Moti Yung, editors, *WISA 09: 10th International Workshop on Information Security Applications*, volume 5932 of *Lecture Notes in Computer Science*, pages 296–308, Busan, Korea, August 25–27, 2009. Springer, Berlin, Germany.

[9] Jens Ackermann, Paul Baecher, Thorsten Franzel, Michael Goesele, and Kay Hamacher. Massively-parallel simulation of biochemical systems. In Stefan Fischer, Erik Maehle, and Rüdiger Reischuk, editors, *INFORMATIK 2009 – Im Focus das Leben*, volume 154 of *Lecture Notes in Informatics*, pages 739–750, Lübeck, Germany, September 28 – October 2, 2009. Gesellschaft für Informatik, Bonn, Germany.

[10] Paul Baecher, Markus Koetter, Thorsten Holz, Maximillian Dornseif, and Felix C. Freiling. The nepenthes platform: An efficient approach to collect malware. In Diego Zamboni and Christopher Krügel, editors, *Recent Advances in Intrusion Detection*, volume 4219 of *Lecture Notes in Computer Science*, pages 165–184, Hamburg, Germany, September 20–22, 2006. Springer, Berlin, Germany.

Acknowledgments

I am deeply grateful to my advisor, Marc Fischlin, whose guidance and opinions I value well beyond academic matters. Working with him was a great pleasure and, almost always, very fruitful. He provided an environment in which open discussions among peers were the norm, rather than an exception. I also tremendously enjoyed the countless (non-work) conversations we had over lunch and while getting coffee.

Being part of Marc's research group was a great honor and a truly exceptional experience. Experiences largely depend on people. I was fortunate enough to meet many wonderful people during my PhD who made this endeavor enjoyable and fun. In particular, I thank Domi and Özi for encouraging me to pursue a PhD in Marc's group and Michi and Richard for a very hearty welcome to their office during the first months of my PhD. I am thankful to Anja, Christina, Cristina, Heike, Andreas, Bertram, Pooya, Giorgia, Tommaso, Arno, Felix and Victoria for their great moral support and for being terrific companions throughout the years.

Finally, I thank all my coauthors for many successful collaborations and Stefan Katzenbeisser, Max Mühlhäuser, Thomas Shrimpton, and Melanie Volkamer for joining my graduation committee.

Paul Baecher
Darmstadt, October 2014

Abstract

Provable security refers to the ability to give rigorous mathematical proofs towards the security of a cryptographic construction; in some sense one of the best possible security guarantees one can attain. These proofs are most often given through so-called reductions to a simpler construction or to some well-studied number-theoretic assumption. This thesis deals with two aspects of such reductions.

First, since a reduction may be difficult to obtain, many reductions for widely-used signature and encryption schemes are conducted in a model that idealizes some underlying building block of the scheme, for example by replacing a hash function with a truly random function. With these reductions in idealized models, it is difficult to compare requirements of cryptographic schemes because the idealization introduces all desired properties simultaneously and it is inexplicit which ones are used and to what extent. This complicates practical considerations when choosing from multiple candidate constructions for the same task.

We develop a novel mechanism to relate schemes proven in idealized models. In this thesis, we present a reductionist paradigm that allows meaningful comparisons of constructions in idealized models with respect to the idealized part. Some of the idealized constructions considered here are the well-known compression-function constructions from blockciphers by Preneel, Govaerts, and Vandewalle (PGV; CRYPTO, 1993), and the twin ElGamal encryption scheme by Cash, Kiltz, and Shoup (Journal of Cryptology, 2009). Our main results show that the random oracle of the twin ElGamal encryption scheme reduces to the random oracle of the regular ElGamal encryption scheme, the PGV constructions fall into two groups, and the so-called double-block-length constructions reduce to one of the PGV constructions with respect to their ideal cipher.

We can thus conclude that the PGV constructions are essentially equivalent within their respective groups and that double-block-length constructions are strictly superior, not only because of their increased key length. Similarly, the regular ElGamal scheme can be replaced by the twin ElGamal scheme (keeping in mind the reduction's tightness), even though the proofs are in an idealized model. These latter results greatly help designers and implementers of practical cryptographic constructions to select the better of two (or more)

seemingly equivalent options. Ideal-model reducibility as a comparison tool is applicable to any two constructions whose proof is in an idealized model.

The second aspect of reductions we study in this thesis relates to the absence of reductions. Sometimes, insurmountable obstacles in finding a reduction result in a proof that reductions of some kind cannot exists at all. In that case, it is particularly important to carefully understand what the non-existent reductions look like—since, perhaps, a slightly different reduction is feasible.

We develop means that allow us to better understand existing reductions in the literature. This thesis presents a new framework, akin to the one by Reingold, Trevisan, and Vadhan (TCC, 2004), for classifying reductions in a more fine-grained and more systematic way.

The new framework clarifies the role of efficiency of adversaries and primitives within reductions, covers meta-reduction separations, and provides new insights on the power of relativizing reductions. Consequently, a classification within the new framework clearly points out avenues to circumvent existing impossibility results and enables an assessment of their strength. The generality of the framework permits classification of a large body of existing reductions, but it is easily extensible to include further properties.

Zusammenfassung

Beweisbare Sicherheit kryptographischer Konstruktionen bezieht sich auf die Möglichkeit, einen rigorosen mathematischen Sicherheitsbeweis zu führen; in gewisser Hinsicht eine der bestmöglichen erreichbaren Sicherheitsgarantien. Diese Beweise sind meist sogenannte Reduktionen auf eine einfachere Konstruktion oder auf eine gut verstandene zahlentheoretische Annahme. In dieser Dissertation studieren wir zwei Aspekte solcher Reduktionen.

Als ersten Aspekt behandeln wir sogenannte Idealisierungen. Vielen Reduktionen für weit verbreitete Signatur- und Verschlüsselungsverfahren liegt ein Modell zugrunde, das Teile der vom Verfahren verwendeten Bausteine idealisiert, beispielsweise in Form einer echt zufälligen Funktion anstelle einer Hashfunktion. Bei derartigen Reduktionen in idealisierten Modellen ist es dann schwierig, Konstruktionen sinnvoll bezüglich ihrer Anforderungen zu vergleichen, weil die idealisierten Bausteine einen Vergleich „verfälschen“. Diese Verfälschung entsteht dadurch, dass die Idealisierung sämtliche Sicherheitseigenschaften auf einmal einbringt und nicht unmittelbar klar ist, welche davon genutzt werden und in welchem Umfang.

Diese Dissertation präsentiert ein Reduktionsparadigma, welches sinnvolle Vergleiche von Konstruktionen in idealisierten Modellen bezüglich der idealisierten Bausteine erlaubt. Zu den hier betrachteten Konstruktionen gehören insbesondere die Kompressionsfunktionen aus Blockciphern nach Preneel, Govaerts und Vandewalle (PGV; CRYPTO, 1993) und das „twin ElGamal“-Verschlüsselungsverfahren von Cash, Kiltz und Shoup (Journal of Cryptology, 2009). Unsere Resultate zeigen, dass sich das Random-Oracle des „twin ElGamal“-Verschlüsselungsverfahrens auf das Random-Oracle des regulären ElGamal-Verschlüsselungsverfahrens reduzieren lässt, die PGV-Funktionen in zwei Gruppen fallen und diverse Kompressionsfunktionen mit doppelter Blocklänge auf eine der PGV-Funktionen bezüglich des Ideal-Ciphers reduzierbar sind.

Folglich lässt sich feststellen, dass die PGV-Konstruktionen im Wesentlichen äquivalent innerhalb der jeweiligen Gruppe sind und dass Konstruktionen mit doppelter Blocklänge tatsächlich mehr Sicherheit bieten – nicht nur wegen der erhöhten Blocklänge. Ebenso kann anstelle des regulären ElGamal-Verschlüsselungsverfahrens das „twin ElGamal“-Verschlüsselungsverfahren ver-

wendet werden (unter Beachtung der Straffheit der Reduktion), obwohl die jeweiligen Beweise in einem idealisierten Modell geführt werden. Diese Resultate helfen Entwicklern praktischer kryptographischer Lösungen, unter mehreren scheinbar äquivalenten Optionen eine gut informierte Auswahl zu treffen. Reduzierbarkeit in idealisierten Modellen findet darüber hinaus grundsätzlich dort Anwendung, wo zwei Konstruktionen, deren Beweis in einem idealisierten Modell erfolgt, miteinander verglichen werden sollen.

Der zweite in dieser Dissertation beleuchtete Aspekt von Reduktionen bezieht sich auf nicht vorhandene Reduktionen. Manchmal ist es nicht möglich, einen Reduktionsbeweis zu führen, sondern zu zeigen, dass eine bestimmte Art von Reduktion gar nicht existieren kann. In solchen Fällen ist es wichtig, die Art der nichtexistenten Reduktionen genau zu verstehen, da möglicherweise leicht abgeänderte Varianten doch existieren könnten.

Wir entwickeln Hilfsmittel, die uns ein besseres Verständnis von in der Literatur existierenden Reduktionen geben. Diese Dissertation präsentiert ein Framework, ähnlich zu dem von Reingold, Trevisan und Vadhan (TCC, 2004), um Reduktionen feingranularer und systematischer zu klassifizieren.

Das neue Framework verbessert unser Verständnis über die Einordnung von Effizienz bei Angreifern und Primitiven innerhalb von Reduktionen, deckt Meta-Reduktionen ab und liefert neue Erkenntnisse über relativierende Reduktionen. Konsequenterweise zeigt eine Klassifikation einer Reduktion in dem vorgestellten Framework neue Richtungen auf, negative Resultate zu umgehen und ermöglicht es, die relative Stärke solcher negativer Resultate besser zu beurteilen. Die Allgemeinheit des Frameworks erlaubt die Erfassung einer Vielzahl existierender Reduktionen; gleichzeitig ist es jedoch auch aufgrund des systematischen Aufbaus einfach erweiterbar.

Contents

Chapter 1

Introduction

"Is it secure?" is the essential question in the field of IT security and cryptography. Modern cryptography approaches this question with the concept of provable security: working within a precisely defined mathematical model, we seek to find rigorous proofs—or guarantees—that the object in question indeed achieves security in that model. This paradigm allows us to avoid inherent flaws in the basic design of the object without having to rely purely on intuition, which is highly prone to deception. Needless to say, provable security is the only accepted strategy towards answering the opening question whenever cryptanalysis or information-theoretic arguments are not applicable, at least in, but not limited to, the cryptographic community.

Reductions. Provable security typically starts out with a conditional proposition of the form "if $\mathbb{A}$, then $\mathbb{B}$," where $\mathbb{A}$ stands for the security of a well-understood primitive A, and $\mathbb{B}$ captures the idea that some newly-constructed object B is secure. This common structure of propositions induces a common way of proving them. Technically, the vast majority of cryptographic proofs show either the contrapositive ("if not $\mathbb{B}$, then not $\mathbb{A}$") or a contradiction ("$\mathbb{A}$ and not $\mathbb{B}$," where the eventually derived contradiction is often "$\mathbb{A}$ and not $\mathbb{A}$"). More concretely, when proving that we can build some secure object B from a secure primitive A, we usually describe a method to turn any alleged attack on B into an attack on A; thus, given the attack on B, we conclude that "not $\mathbb{A}$." The constructive nature of this proof technique leads to a desirable side effect. Either $\mathbb{A}$ is true and we get a secure object B, or $\mathbb{A}$ is false and any attack on B leads to a concrete method to "break" A—and if $\mathbb{A}$ corresponds to some presumably computationally hard problem like, say, factoring, then the method will help to solve this problem. This method is commonly called a *reduction*.

Despite being part of virtually any work in the field, fully understanding reductions, as an object of study themselves, is far from trivial. To appreciate this, consider for a moment the absence of a reduction. An important aspect of cryptography is to study the relationship between different objects in order

to determine if some construction can be based, at all, on some primitive. Being unable to find a reduction is a first indicator that this question may entail a negative answer. Indeed, proving that it is impossible to securely build certain constructions from some primitive, i.e., proving a separation, is a popular theme among theoreticians. The term "impossibility" is a misnomer, though: usually, one merely shows that certain classes of reductions cannot exist. This is a much more restricted statement than showing that no secure construction from that primitive exists whatsoever. There may thus be other types of reductions, or even entirely arbitrary proof techniques, with different properties that do indeed exist (of course, this is less likely the more reductions are being ruled out—especially if that means a new technique needs to be discovered first). Separations are hence an important reason why we should carefully examine the exact properties of a reduction.

Unfortunately, we currently lack a good language to concisely and explicitly communicate those intricacies. In their highly-cited paper, Reingold, Trevisan, and Vadhan (RTV, [RTV04]) laid out some of the basic foundations by distinguishing essentially three types of reductions. For each type, they investigate how the reduction algorithm interacts with the alleged attacker against the newly constructed object B, and how these types are related among each other. The main focus in their classification is on the varying degree of *black boxness*, that is, how much information about the attacker against construction B can the reduction algorithm access. On one end of the spectrum, the reduction can only communicate with the attacker through a fixed input/output interface without being able to look "into the box"; hence the term black box. On the other end of the spectrum, the reduction gets a full description of the attacker, for example in terms of its underlying program code.

The RTV framework is, however, rather coarse. As an example, consider the types of reductions that are ruled out by the increasingly popular meta-reduction separation technique. There, the reduction algorithm treats the attacker against construction B as a black box, but can potentially make use of the program code of the *primitive* whose security is captured by $\mathbb{A}$. For this type of reduction, the RTV framework does not offer a good match. In order to classify this reduction, one is forced to go for a type that is close to the non-black-box end of the spectrum. Thus, the tempting assertion that this type is ruled out by a meta reduction is incorrect, because the type of reduction, in fact, encompasses a much broader class of reductions. It is hence tedious to specify which reductions are ruled out by a meta reduction result, and consequently, it tends to be omitted. That, in turn, makes it easy to misunderstand the entire result.

Apart from these coarseness issues, the binary distinction between black-box and non-black-box use itself seems unrefined. Technically, a reduction that uses information such as the success probability of an alleged attacker (as it is the case in the hardcore-predicate reduction presented in [Gol04] due to Rackoff [Gol04, §2.7.1], for example) is not completely black box any more.

On the other hand, such use of the success probability appears to be a much milder requirement than taking advantage of, say, the code of the adversary. Again, there is currently no established succinct language to convey this.

Idealized models. Occasionally, however, the community at large fails to find neither a proof nor a counterexample for constructions that are widely used in practice. The popular OAEP encryption padding [BR94] (part of PKCS #1) is a prominent example. In these cases, researchers sometimes resort to a heuristic involving an idealized model, where the world is augmented by an (unrealistically) ideal version of some primitive that the construction relies on, in order to make a proof go through. This idealization takes the form of an oracle answering queries according to a specific distribution, but does not reveal its inner workings. A critical consequence of this latter property is that it enables a very liberal reduction during which the reduction algorithm can learn all the queries to the ideal primitive and even set answers (within the rules determined by the distribution).

Not surprisingly, such idealized primitives are considered to be very powerful and controversial [CGH98, KL08, KM07a]. They can neither be implemented efficiently, nor is there any real-world justification for the liberty that a reduction algorithm obtains. Yet, it is unclear how powerful they are exactly and what that means in practice, when the idealized primitive is eventually instantiated by a real-world implementation like AES or SHA-3. An artificial cryptosystem based on an abstract hash function H could, for example, check if $H(0) = \text{SHA-3}(0)$ and then, and only then, behave trivially insecure. Clearly, this equality does not hold with overwhelming probability if H is ideal, i.e., a random function—but it will always hold when H is instantiated with SHA-3. Indeed, in some sense there is no guarantee at all: we know schemes that are secure in an ideal model, but become insecure with any real implementation. Canetti et al. [CGH98] established this result by giving artificial examples and Nielsen [Nie02] identified with non-committing encryption the first natural task where the methodology fails.

The uncertainty regarding the power of idealized primitives has another troubling aspect: directly comparing two constructions that both use the same idealized primitive can be very difficult. In particular, consider first the case where both constructions look very similar in the idealized model. If it is unclear how much power either construction draws from the idealized primitive, then the level of security may be very different once the indealized primitive is instantiated by the same implementation. Prominent concrete examples for such a situation are the various so-called PGV compression functions based on blockciphers [PGV93, BRSS10], which result from various ways to wire a blockcipher and XOR gates. Orthogonally to the case where two or more constructions seem similar, consider that one construction is (provably) strictly better than another one in all respects. When instantiated with one particular implementation, it could turn out that the seemingly superior construction is

actually weaker because it relies more heavily on certain security properties of the primitive. This problem is illustrated best by two versions of hashed ElGamal encryption schemes. Cash, Kiltz, and Shoup [CKS09] improve over the standard hashed ElGamal encryption scheme by weakening the assumption on which the scheme's security is based on. Here, a rigorous declaration that their improvement is strictly better seems problematic at first as both schemes enjoy a proof in an idealized model.

In a somewhat extreme case, one construction could be completely uninstantiable or it may be much harder to do so. Put differently, any comparison in the aforementioned settings is blurred by the idealized primitive. For a practitioner it is thus far from obvious which construction to prefer for an implementation, but even on a theoretical level it would be much more desirable to have sound comparisons. Notwithstanding these drawbacks, it is noteworthy that proofs in idealized models do have value. First, it means that a potential attack *does* have to exploit the inner structure of the instantiation of the idealized primitive. Second, a proof in this model is arguably better than no proof at all, as it provides a basic sanity check for the construction in question.

Contributions of this Thesis In this thesis, we address the aforementioned issues, dealing with the latter ones first.

In Chapter 3, we apply the classical reductionist approach in a novel way. Reducing schemes to each other with respect to an idealized primitive allows us to make sound comparisons on what the requirements on the idealized primitive are. We demonstrate this by revisiting the compression functions initially studied by Preneel, Govaerts, and Vandewalle (PGV, [PGV93]) which serve as building blocks for many hash-function designs. Although we know that 12 of the functions provide optimal security in the ideal-cipher model [BRSS10], little is known about the relationships among these designs. Our treatment shows that the functions partition into two groups of size six where any function immediately reduces to any other function with respect to the cipher within that group. Moreover, we explore the reducibility of three more complex compression function designs based on a cipher using doubled key lengths [LM92, Hir06]. Here we rule out the existence of a broad class of transformations and thereby establish that these designs impose fewer requirements on the blockcipher than the PGV functions when instantiated.

Continuing with our reductionist methodology, we next turn to random oracles in Chapter 4. The classical hashed ElGamal encryption scheme can be proven secure against chosen-ciphertext attacks under the *strong* Diffie–Hellman (DH) assumption in the random-oracle model. An improved variant of this scheme by Cash, Kiltz, and Shoup [CKS09] is only slightly less efficient but relaxes the assumption to the ordinary DH assumption. We formally show that this relaxation is indeed not at the expense of increased requirements from

the random oracle; a possibility that was so far not excluded by the existing work (and, to the best of our knowledge, not considered in any prior work). Thus, apart from the slight overhead, one can safely choose the improved variant for practical considerations. We furthermore discuss several other reducibility results among various signature schemes towards the end of this chapter. These results are interesting insofar as they allow us to argue about the relative instantiability among those schemes.

Chapter 5 finally deals with the taxonomy of cryptographic reductions in general. We present a comprehensive, systematic framework and notation that enables capturing a wide range of different types of reductions. In particular, the framework offers a notion that corresponds accurately to those reductions ruled out by meta reductions. We relate all notions introduced in this chapter to each other by showing implications or providing separating counterexamples. For situations where a binary classification into black-box and non-black-box usage is difficult (e.g., [GL89]), we propose a unified view to reflect this via parametrization. Moreover, we map out dimensions that were previously not—or only implicitly—explored, such as the distinction between efficient/inefficient adversaries and primitives, which eventually leads to a better understanding of the power of relativizing reductions. Due to its systematic design, the framework can be easily extended to further dimensions in order to include more properties of reductions.

Chapter 2

Notation and Definitions

In this chapter, we set the basic notation used throughout the thesis. Readers who are already familiar with cryptographic literature may wish to advance to the next chapter.

General notation. We write $x \leftarrow y$ for assigning value y to variable x. We write $x \leftarrow\!\!\$\ X$ for sampling x from (finite) set X uniformly at random. If $\mathcal{A}$ is a probabilistic algorithm we write $y \leftarrow\!\!\$\ \mathcal{A}(x_1, \ldots, x_n)$ for the action of running $\mathcal{A}$ on inputs $x_1, \ldots, x_n$ with coins chosen uniformly at random, and assigning the result to y. We use "|" for string concatenation, denote the bit-wise complement of $x \in \{0,1\}^\star$ by $\overline{x}$. We set $[n] := \{1, \ldots, n\}$. We say $\epsilon(\lambda)$ is negligible if $|\epsilon(\lambda)| \in \lambda^{-\omega(1)}$. The modifier "ppt" stands for probabilistic polynomial time.

Oracle access and black boxes. If $\mathcal{A}$ is an algorithm, we write $\mathcal{A}^O$ to indicate that $\mathcal{A}$ has "oracle access" to O, where O may be another algorithm, a function, or a distribution. Oracle access usually means that O is a black box: algorithm $\mathcal{A}$ may query O adaptively with chosen inputs $x_1, \ldots, x_n$, but learns no information about O besides the outputs of $O(x_1), \ldots, O(x_n)$. In particular, algorithm $\mathcal{A}$ does not learn how O is implemented, let alone its (possibly non-existing) code.

All this assumes that $\mathcal{A}$ does not "depend" on O in the sense that there is a different version of $\mathcal{A}$ for each O. In that case, oracle O is no longer a black box for $\mathcal{A}$, of course, because $\mathcal{A}$ is then fully aware of what it is dealing with. We indicate this somewhat unusual situation clearly. The formal meaning of the term "black box" in varying degrees is the subject matter of Chapter 5.

Idealized models. The two idealized models we encounter in this thesis are the random-oracle model and the ideal-cipher model. Both models share the same underlying concept. We consider a large set of functionally correct objects and make a probabilistic security statement that involves the uniform choice of one object from the set. More concretely, in the random-oracle model, we randomly sample a function from the set of all admissible functions. Similarly,

in the ideal-cipher model, we sample from the set of all keyed block ciphers. All parties involved in a game that models some security property have (oracle) access to the sampled object and the probability space is augmented by the sampling process of the object. These models are considered ideal, due to the complete absence of any structure and they enjoy in some sense the "best possible security" regarding the sampled object.

Chapter 3

Ideal-Cipher Reducibility

In this first chapter of the thesis, we investigate several compression functions in terms of their relative strengths. All function designs are known to enjoy various security properties in the ideal-cipher model, yet it is unclear which ones to prefer in a practical construction. We show reducibility or irreducibility with respect to the underlying ideal cipher for the so-called PGV compression functions as well as major double-block-length (DBL) compression function designs.

In the following Section 3.1, we recall some facts about the PGV and DBL designs and lay out our reducibility approach. After introducing some chapter-specific notation in Section 3.2, we go on to separate the PGV functions into two groups in Section 3.3. We then relate the DBL constructions one to another and to the PGV functions in Section 3.4. Finally, in Section 3.5, we discuss some concluding remarks and identify avenues for further research.

This work was presented at EUROCRYPT 2013 [BFFS13].

3.1 Introduction

The design of hash functions (or compression functions) from blockciphers has been considered very early in modern cryptography. Preneel, Govaerts, and Vandewalle [PGV93] initiated a systematic study of designing a compression function $\mathsf{F} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ out of a blockcipher $\mathsf{E} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ by analyzing all 64 possible ways to combine the relevant inputs and outputs using xors only. Preneel et al. conjectured only 12 out of these 64 PGV constructions to have certain security properties, including the well-known constructions of Matyas–Meyer–Oseas (MMO) and Davies–Meyer (DM). The idea continues to influence hash-function design till today. Indeed, one of the former five final candidates in the SHA-3 competition, Skein [FLS+08], explicitly refers to this design methodology, and other former candidates like Grøstl [GKM+11] are based on similar principles.

The conjecture about the 12 secure PGV variants was later shown to be true in the *ideal-cipher model* (ICM) by Black et al. [BRS02, BRSS10]. Roughly speaking, Black et al. show that assuming E implements a random blockcipher, the 12 secure PGV compression functions achieve optimal security of $\Theta(q^2 \cdot 2^{-n})$ for collision resistance and $\Theta(q \cdot 2^{-n})$ for preimage resistance, where q is the number of queries to the ideal cipher (and its inverse). Black et al. also discuss 8 further variants which, if used in a hash-iteration mode, attain optimal collision resistance and suboptimal preimage resistance of $\Theta(q^2 \cdot 2^{-n})$. The remaining 44 PGV versions are insecure and of no relevance here.

Idealized models. As pointed out by Black et al. [BRSS10], security proofs for the PGV schemes in the ICM should be treated with care. Such results indicate that, in order to break the security of the PGV scheme, one would need to take advantage of structural properties of the blockcipher. Yet blockciphers such as AES, or the Threefish blockcipher used in Skein, clearly display a structure which is far from an ideal object. For instance, IDEA seems quite unsuitable to base a compression function on [WPS+12], while for AES recent related-key attacks [BK09, BKN09] cast some shadow on its suitability for this purpose. Indeed, Khovratovich [Kho10, Corollary 2] states unambiguously that "AES-256 in the Davies–Meyer hashing mode leads to an insecure hash function," but remarks that it is not known how to attack, for instance, double-block-length constructions. Moreover, it is currently still unknown how to exploit these weaknesses in AES-256 to break the standard collision or preimage security of any AES-instantiated PGV compression function. Consequently it may well be that AES makes some of the 12 PGV constructions secure, whereas others turn out to be insecure, despite a proof in the ICM. Unfortunately, it is very hard to make any security claims about specific PGV constructions with respect to a "real" blockcipher, or to even determine exactly the necessary requirements on the blockcipher for different PGV constructions to be secure.

Recently, a similar issue for the random-oracle model, where a monolithic idealized hash function is used, has been addressed by Baecher and Fischlin [BF11] via the so-called random-oracle reducibility. We will explore random-oracle reducibility in greater detail in Chapter 4 of this thesis, but give a brief overview here as we will first work with a simplified notion throughout this chapter. The idea is to relate the idealized hash functions in different (primarily public-key) schemes, allowing to conclude that the requirements on the hash function in one scheme are weaker than those in the other scheme. That is, Baecher and Fischlin consider two cryptographic schemes A and B with related security games in the random-oracle model. They define that the random oracle in scheme B reduces to the one in scheme A, if *any* instantiation H of the random oracle, possibly through an efficient hash function or again by an oracle-based solution, which makes scheme A secure, also makes scheme B secure. As such, the requirements on the hash function for scheme B are weaker than those for the one in scheme A. To be precise, Baecher and Fischlin allow

an efficient but deterministic and stateless transformation $\mathsf{T}^{\mathcal{H}}$ for instantiating the random oracle in scheme B, to account for, say, different input or output sizes of the hash functions in the schemes. Using such transformations they are able to relate the random oracles in some public-key encryption schemes, including some ElGamal-type schemes.

Our results for the PGV constructions. We apply the idea of oracle reducibility to the ideal-cipher model and the PGV constructions. Take any two of the 12 PGV constructions, PGV_i and PGV_j, which are secure in the ICM. The goal is to show that any blockcipher (ideal or not) which makes PGV_i secure, also makes PGV_j secure. Here, security may refer to different games such as standard notion for collision resistance, preimage resistance, or everywhere preimage resistance [RS04], or more elaborate notions such as preimage awareness [DRS09]. One could even ask the same question for indifferentiability from random functions [MRH04], but the PGV constructions, as pointed out in [CDMP05, KM07b], do not achieve this level of security. This, and other points discussed within the body, motivates why we chose the oracle reducibility notion of [BF11] rather than the indifferentiability reducibility notion in [MRH04].

Our first result divides the 12 secure PGV constructions into two groups $\mathcal{G}_1$ and $\mathcal{G}_2$ of size 6, where within each group the ideal cipher in each construction reduces to the ideal cipher in any other construction (with respect to collision resistance, [everywhere] preimage resistance, and preimage awareness). We sometimes call these the PGV_1-group and the PGV_2-group respectively: these two functions are representatives of their respective groups. Across different groups, however, and for any of the security games, our results change. Starting with the ideal cipher, we can derive a blockcipher which makes all schemes in one group secure, whereas any scheme in the other group becomes insecure under this blockcipher. This separates the PGV_1-group and the PGV_2-group in terms of *direct* ideal-cipher reducibility. In direct reducibility we use the blockcipher in question without any modifications in another construction. This was one of the reasons to investigate different PGV constructions in the first place. For *free* reductions allowing arbitrary transformations T of the blockcipher, we show that the PGV constructions can be seen as transformations of each other, and under suitable T all 12 PGV constructions reduce to each other.

Preneel et al. [PGV93] already discussed equivalence classes from an attack perspective. Our work reaffirms these classes and puts them on a solid theoretical foundation. Dividing the 12 constructions into two groups allows us to say that, within each group, one can use a blockcipher in a construction under the same *qualitative* assumptions on the blockcipher; only across the groups this becomes invalid. In other words, the sets (or more formally, distributions) of "good" blockciphers for the groups are not equal, albeit they clearly share the ideal cipher as a common member making both groups simultaneously secure. We note that our results are also *quantitatively* tight in the sense that the

blockciphers within a group are proven to be tightly reducible to each other in terms of the number of queries, running times, and success probabilities.

PGV and double-block-length hashing. Double-block-length (DBL) hash or compression functions aim at surpassing the $2^{n/2}$ upper bound for collision resistance of the PGV constructions by using two "PGV-like" constructions in parallel, doubling the output length. There are three major such compression functions, namely, Tandem-DM (TDM, [LM92]), Abreast-DM (ADM, [LM92]), and Hirose's construction (HDM, [Hir06]). Several results underline the optimality of collision-resistance [Hir06, LK11, LSS11] and preimage-resistance bounds [AFK+11] for these functions in the ICM.

Continuing with ideal-cipher reducibility, we establish a connection between the basic PGV constructions and the double-block-length compression functions. Since all the DBL constructions have a "PGV_1-part" (with twice the key size) built in, it follows that any collision for any of the DBL functions immediately yields a collision for PGV_1 built from a blockcipher with $2n$-bit key. In other words, the ideal cipher in the DBL constructions directly reduces to the one in double-key PGV_1. We also prove that there is a free reduction to single-key PGV_1 from this double-key variant, thereby relating DBL functions to PGV_1 for free transformations. It follows, via a free reduction to PGV_1 and a free reduction from PGV_1 to PGV_2, that DBL functions reduce to PGV_2 for free transformations. An analogous result also applies to the everywhere preimage-resistance game, but, somewhat curiously, we show such a result cannot hold for the (standard) preimage-resistance game.

When it comes to free reducibility from PGV to DBL functions, we present irreducibility results for the collision-resistance and [everywhere] preimage-resistance games. We achieve this by making use of an interesting relationship to (lower bounds for) hash combiners [Her05, HKN+05, Pie08]. Namely, if one can turn a collision (or preimage) for, say, PGV_1 into one for a DBL compression function, then we can think of PGV_1, which has n-bit digests, as a sort of robust hash combiner for the DBL function (which has $2n$-bit outputs). However, known lower bounds for hash combiners [Pie08] tell us that such a combiner—with tight bounds and being black box—cannot exist, and this transfers to ideal-cipher reducibility. More in detail, by combining Pietrzak's techniques [Pie08] with a lower bound on generic collision finders by Bellare and Kohno [BK04] on compression functions, we confirm the irreducibility result formally for the simple case of black-box reductions making only a single call to the PGV collision-finder oracle (as also discussed in [Pie08]). In summary, not only do the DBL functions provide stronger guarantees in terms of quantitative security (as well as efficiency and output length), but they also provably rely on qualitatively weaker assumptions on the blockcipher for the collision-resistance and everywhere preimage-resistance games.

Finally, we demonstrate that for none of the aforementioned DBL constructions the ideal cipher directly reduces to the one in either of the other

schemes. That is, starting with the ideal cipher, for each target DBL function we construct a blockcipher which renders it insecure but preserves collision resistance for the other two functions. We are not aware of an analogous result for free reductions, but can exclude transformations which are involutions.

Practical implications. Our results show that there is "no clear winner" among the PGV constructions in the sense that one construction always relies on weaker assumptions about the blockcipher than the other ones and should be therefore preferred in practice. This depends on the blockcipher in question. As expressed above, settling this for a specific blockcipher may be tedious, though. Nonetheless, our results do show that DBL constructions are superior in this regard, and that one may switch between PGV constructions of the same group in order to match other practical stipulations.

3.2 Notation

Blockciphers. A blockcipher with key length k and block length n is a set of permutations and their inverses on $\{0,1\}^n$ indexed by a key in $\{0,1\}^k$. This set can therefore be thought of as a pair of functions

$$\mathsf{E} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n \quad \text{and} \quad \mathsf{E}^{-1} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n.$$

We denote the set of all such blockciphers by $\mathsf{Block}(k,n)$. A blockcipher is efficient if the above functions can be implemented by a ppt Turing machine.

Ideal and idealized (block)ciphers. An idealized (block)cipher with key length k and block length n is a distribution $\mathcal{E}$ on $\mathsf{Block}(k,n)$. We often consider an $\mathcal{E}$-idealized model of computation where all parties are given oracle access to a blockcipher chosen according to $\mathcal{E}$. *The* ideal-cipher (model) is the $\mathcal{E}$-idealized model where $\mathcal{E}$ is the uniform distribution on $\mathsf{Block}(k,n)$. We denote the set of all idealized ciphers with key length k and block length n (i.e., the set of all distributions on $\mathsf{Block}(k,n)$) by $\mathsf{Ideal}(k,n)$. Below, when saying that one has oracle access to an idealized cipher $\mathcal{E}$ it is understood that a blockcipher is sampled according to $\mathcal{E}$ and that one gets oracle access to this blockcipher.

Compression functions. A compression function is a function mapping $\{0,1\}^l$ to $\{0,1\}^m$ where $m < l$. We are primarily interested in compression functions which are built from a blockcipher. In this case we write $\mathsf{F}^{\mathsf{E},\mathsf{E}^{-1}} : \{0,1\}^l \to \{0,1\}^m$. A compression function is often considered in an idealized model where its oracles are sampled according to an idealized cipher $\mathcal{E}$.

Security notions for compression functions

We now recall a number of fundamental security properties associated with blockcipher-based hashing.

Definition 1 (Everywhere preimage and collision resistance [RS04])**.** *Let* $\mathsf{F}^{\mathsf{E},\mathsf{E}^{-1}} : \{0,1\}^l \to \{0,1\}^m$ *be a compression function with oracle access to a blockcipher in* $\mathsf{Block}(k,n)$*. Let* $\mathcal{E}$ *denote an idealized cipher on* $\mathsf{Block}(k,n)$*. The preimage- (resp., everywhere preimage-, resp., collision-) resistance advantage of an adversary* $\mathcal{A}$ *in the* $\mathcal{E}$*-idealized model against* $\mathsf{F}^{\mathsf{E},\mathsf{E}^{-1}}$ *are defined by*

$$\mathbf{Adv}^{\mathsf{pre}}_{\mathsf{F},\mathcal{E}}(\mathcal{A}) := \Pr\left[\mathsf{F}^{\mathsf{E},\mathsf{E}^{-1}}(X') = Y \;:\; \begin{array}{l}(\mathsf{E},\mathsf{E}^{-1}) \leftarrow\!\!\$\ \mathcal{E};\ X \leftarrow\!\!\$\ \{0,1\}^l;\\ Y \leftarrow \mathsf{F}^{\mathsf{E},\mathsf{E}^{-1}}(X);\ X' \leftarrow\!\!\$\ \mathcal{A}^{\mathsf{E},\mathsf{E}^{-1}}(Y)\end{array}\right],$$

$$\mathbf{Adv}^{\mathsf{epre}}_{\mathsf{F},\mathcal{E}}(\mathcal{A}) := \Pr\left[\mathsf{F}^{\mathsf{E},\mathsf{E}^{-1}}(X) = Y \;:\; \begin{array}{l}(\mathsf{E},\mathsf{E}^{-1}) \leftarrow\!\!\$\ \mathcal{E};\ (Y,\mathrm{st}) \leftarrow\!\!\$\ \mathcal{A}_1;\\ X \leftarrow\!\!\$\ \mathcal{A}_2^{\mathsf{E},\mathsf{E}^{-1}}(\mathrm{st})\end{array}\right],$$

$$\mathbf{Adv}^{\mathsf{coll}}_{\mathsf{F},\mathcal{E}}(\mathcal{A}) := \Pr\left[\begin{array}{l}X_0 \neq X_1 \wedge\\ \mathsf{F}^{\mathsf{E},\mathsf{E}^{-1}}(X_0) = \mathsf{F}^{\mathsf{E},\mathsf{E}^{-1}}(X_1)\end{array} \;:\; \begin{array}{l}(\mathsf{E},\mathsf{E}^{-1}) \leftarrow\!\!\$\ \mathcal{E};\\ (X_0,X_1) \leftarrow\!\!\$\ \mathcal{A}^{\mathsf{E},\mathsf{E}^{-1}}\end{array}\right].$$

For the set S_q of all adversaries which make at most q queries (E queries plus E^{-1} queries) we define

$$\mathbf{Adv}^{\mathsf{pre}}_{\mathsf{F},\mathcal{E}}(q) := \max_{\mathcal{A}\in S_q}\left\{\mathbf{Adv}^{\mathsf{pre}}_{\mathsf{F},\mathcal{E}}(\mathcal{A})\right\},$$

and similarly for the everywhere preimage-resistance and collision-resistance games. We note that although a compression function cannot be collision resistant nor everywhere preimage resistance with respect to a *fixed* blockcipher, reducibility arguments still apply [Rog06].

Some of our results also hold for "more advanced" properties of hash or compression functions like preimage awareness [DRS09]. If so, we mention this briefly. Roughly speaking, preimage awareness [DRS09] states that any adversary which comes up with an image Z for a compression function, already knows a preimage X for it. This is formalized though the existence of an extractor algorithm $\mathcal{X}$ which can recover the value X from Z and the list α of previous queries to E and E^{-1}.

Definition 2 (Preimage awareness [DRS09])**.** *Let* $\mathsf{F}^{\mathsf{E},\mathsf{E}^{-1}} : \{0,1\}^l \to \{0,1\}^m$ *be a compression function with oracle access to a blockcipher in* $\mathsf{Block}(k,n)$*. Let* $\mathcal{E}$ *denote an idealized cipher on* $\mathsf{Block}(k,n)$*. The preimage awareness advantage of an adversary* $\mathcal{A}$ *with respect to the deterministic extractor* $\mathcal{X}$ *in the* $\mathcal{E}$*-idealized model against* $\mathsf{F}^{\mathsf{E},\mathsf{E}^{-1}}$ *is defined by*

$$\mathbf{Adv}^{\mathsf{pra}}_{\mathsf{F},\mathcal{E},\mathcal{X}}(\mathcal{A}) := \Pr\left[\mathbf{Exp}^{\mathsf{pra}}_{\mathsf{F},\mathcal{E},\mathcal{X}}(\mathcal{A}) = 1\right],$$

where $\mathbf{Exp}^{\mathsf{pra}}_{\mathsf{F},\mathcal{E},\mathcal{X}}(\mathcal{A})$ *is shown in Figure 3.1.*

Dodis et al. [DRS09] show that the 12 optimally secure PGV constructions are preimage aware in the ideal-cipher model. Vice versa, preimage awareness (for compressing functions) has been shown to imply collision resistance, and help in proving indifferentiability from a random oracle in certain constructions [DRS09].

$\underline{\mathbf{Exp}^{\mathsf{pra}}_{\mathsf{F},\mathcal{E},\mathcal{X}}(\mathcal{A})}$:
$(\mathsf{E}, \mathsf{E}^{-1}) \leftarrow\!\!\$\ \mathcal{E}$
$X \leftarrow\!\!\$\ \mathcal{A}^{\mathbf{E},\mathbf{E^{-1}},\mathbf{Ex}}$
$Z \leftarrow \mathsf{F}^{\mathsf{E},\mathsf{E}^{-1}}(X)$
return $(X \neq \mathsf{V}[Z] \wedge \mathsf{Q}[Z] = 1)$

$\underline{\text{oracle } \mathbf{E}(K, X)}$:
$Y \leftarrow \mathsf{E}(K, X)$
$\alpha \leftarrow \alpha | (K, X, Y)$
return Y

$\underline{\text{oracle } \mathbf{E^{-1}}(K, Y)}$:
$X \leftarrow \mathsf{E}^{-1}(K, Y)$
$\alpha \leftarrow \alpha | (K, X, Y)$
return X

$\underline{\text{oracle } \mathbf{Ex}(Z)}$:
$\mathsf{Q}[Z] \leftarrow 1$
$\mathsf{V}[Z] \leftarrow \mathcal{X}(Z, \alpha)$
return $\mathsf{V}[Z]$

Figure 3.1: Experiment defining preimage awareness.

Reducibility

In order to define what it means for an idealized cipher to reduce to another, we begin with a semantics for security games similar to that in [BR06]. We capture the three security properties above by our notion, but can also extend the framework to cover a larger class of security games, such as complex multi-stage games and simulation-based notions. In the simpler case, we will consider a game between a challenger or a game Game and a sequence $\mathcal{A}_1, \mathcal{A}_2, \ldots$ of admissible adversaries (e.g., those which run in polynomial time). When the game terminates by outputting 1, this is deemed a success for the adversary (in that instance of the game). To determine the overall success of the adversaries, we then measure the success probability with respect to threshold t (e.g., 0 for computational games, or $\frac{1}{2}$ for decisional games). We present our formalism in the concrete setting. However, our definitions can be easily extended to the asymptotic setting by letting the game, its parameters, and adversaries to depend on a security parameter.

Definition 3 (Secure $\mathcal{E}$-idealized games)**.** *An $\mathcal{E}$-idealized game consists of an oracle Turing machine* Game *(also called the challenger) with access to an idealized cipher $\mathcal{E}$ and n adversary oracles, a threshold $t \in [0, 1]$, and a set S of n-tuples of admissible adversaries. The game terminates by outputting a bit. The advantage of adversaries $\mathcal{A}_1, \ldots, \mathcal{A}_n$ against* Game *is defined as*

$$\mathbf{Adv}^{\mathsf{Game}}_{\mathcal{E}}(\mathcal{A}_1, \ldots, \mathcal{A}_n) := \left| \Pr\left[\mathsf{Game}^{\mathsf{E},\mathsf{E}^{-1},\mathcal{A}_1^{\mathsf{E},\mathsf{E}^{-1}},\ldots,\mathcal{A}_n^{\mathsf{E},\mathsf{E}^{-1}}} = 1 \right] - t \right|,$$

where the probability is taken over the coins of Game*, $\mathcal{A}_1, \ldots, \mathcal{A}_n$, and the choice of the cipher $(\mathsf{E}, \mathsf{E}^{-1}) \leftarrow\!\!\$\ \mathcal{E}$. For bounds $\epsilon \in [0, 1]$ and $T, q \in \mathbb{N}$ we say* Game *is* (q, T, ϵ)-secure *if*

$$\forall (\mathcal{A}_1, \ldots, \mathcal{A}_n) \in S : \mathbf{Adv}^{\mathsf{Game}}_{\mathcal{E}}(\mathcal{A}_1, \ldots, \mathcal{A}_n) \leq \epsilon$$

and Game *together with any set of admissible adversaries runs in time at most T (counting adversary runs as unit cost) and makes at most q queries to the sample of the idealized cipher, including those of the adversaries.*

For example, the above notion captures everywhere preimage resistance by having $\mathcal{A}_1$ terminate by outputting (Y, st) with no access to the blockcipher, and $\mathcal{A}_2^{\mathsf{E},\mathsf{E}^{-1}}(\mathrm{st})$ return some X; the challenger then outputs 1 if and only if $\mathsf{F}^{\mathsf{E},\mathsf{E}^{-1}}(X) = Y$. Note that in particular, the construction F is usurped, together with the everywhere preimage experiment, in the general notation Game. We also note that with the above syntax we can combine multiple games into one by having a "master" adversary $\mathcal{A}$ first send a label to the challenger deciding which subgame to play and then invoking the corresponding parties and game. Note also that as in [BF11] we assume that an idealized cipher can be given as an entirely ideal object, as a non-ideal object through a full description of an efficient Turing machine given as input to the parties, or a mixture thereof.

Ideal-cipher transformations. A transformation of ideal ciphers is an oracle function T which maps a blockcipher from $\mathsf{Block}(k, n)$ to another blockcipher in $\mathsf{Block}(k', n')$. Typically, we will only be interested in *efficient* transformations i.e., those which can be implemented by efficient oracle Turing machines in the $\mathcal{E}$-idealized model, written $\mathsf{T}^{\mathsf{E},\mathsf{E}^{-1}}$. Note that the requirement of T being a function implies that, algorithmically, the oracle Turing machine is deterministic and stateless. We discuss this restriction shortly after the definition. Below we envision the (single) transformation T to work in different modes $\mathsf{Enc}, \mathsf{Dec}$ to provide the corresponding interfaces for a blockcipher $(\mathsf{E}', \mathsf{E}'^{-1})$. Slightly abusing notation, we simply write T and T^{-1} for the corresponding interfaces E' and E'^{-1} (instead of $\mathsf{T}_{\mathsf{Enc}}^{\mathsf{E},\mathsf{E}^{-1}}$ for E' and $\mathsf{T}_{\mathsf{Dec}}^{\mathsf{E},\mathsf{E}^{-1}}$ for E'^{-1}). The transformation is written as

$$\mathsf{E}'(K, M) := \mathsf{T}^{\mathsf{E},\mathsf{E}^{-1}}(K, M) \quad \text{and} \quad \mathsf{E}'^{-1}(K, M) := \mathsf{T}^{-1\,\mathsf{E},\mathsf{E}^{-1}}(K, M).$$

Any transformation T also induces a mapping from $\mathsf{Ideal}(k, n)$ to $\mathsf{Ideal}(k', n')$. When E is sampled according to $\mathcal{E}$, then T induces an idealized cipher $\mathcal{E}' \in \mathsf{Ideal}(k', n')$ which we occasionally denote by $\mathsf{T}^{\mathcal{E}}$.

We are now ready to define idealized-cipher reducibility.

Definition 4 (Ideal-cipher reducibility)**.** *Let* Game_1 *and* Game_2 *be two idealized games relying on blockciphers in* $\mathsf{Block}(k, n)$ *and* $\mathsf{Block}(k', n')$ *respectively. We say* the idealized cipher in Game_2 reduces to the idealized cipher in Game_1*, if for any* $\mathcal{E}_1 \in \mathsf{Ideal}(k, n)$ *there is a deterministic, stateless, and efficient transformation* $\mathsf{T} : \mathsf{Block}(k, n) \to \mathsf{Block}(k', n')$ *such that if*

$$\forall (\mathcal{A}_{1,1}, \ldots, \mathcal{A}_{1,n_1}) \in S_1 \ : \ \mathbf{Adv}_{\mathcal{E}_1}^{\mathsf{Game}_1}(\mathcal{A}_{1,1}, \ldots, \mathcal{A}_{1,n_1}) \leq \epsilon_1,$$

whenever Game_1 *runs in time at most* t_1 *and makes at most* q_1 *queries to the block cipher sampled according to* $\mathcal{E}_1$*, then setting* $\mathcal{E}_2 := \mathsf{T}^{\mathcal{E}_1}$*, we have that*

$$\forall (\mathcal{A}_{2,1}, \ldots, \mathcal{A}_{2,n_2}) \in S_2 \ : \ \mathbf{Adv}_{\mathcal{E}_2}^{\mathsf{Game}_2}(\mathcal{A}_{2,1}, \ldots, \mathcal{A}_{2,n_2}) \leq \epsilon_2,$$

where Game *runs in time at most* t_2 *and makes at most* q_2 *queries to the blockcipher sampled according to* $\mathcal{E}_2$*. In this case we say the reduction is*

$(q_1/q_2, T, t_1/t_2, \epsilon_1/\epsilon_2)$-tight, where T is an upper bound on the number of queries that T *places to its oracle per invocation. When $k = k'$, $n = n'$, and* T *is the identity transformation, we say the reduction is* direct*; else it is called* free*.*

Definitional choices. Throughout this chapter, our focus is on reducibility among blockcipher-based hash functions. In this setting, there are often no assumptions beyond the idealized cipher being chosen from a certain distribution. In this case, the strict, strong, and weak reducibility notions as discussed in [BF11] and later in Chapter 3 all collapse to the one given above. Of particular interest to us are two types of transformations. First, *free* transformations which can be arbitrary, and second the identity/dummy transformation which does not change the cipher. This latter type of direct reducibility asks if any idealized cipher making one construction secure makes the other secure too. The former type, however, apart from appropriately modifying the syntactical aspects of the blockcipher (such as the key or the block size), asks if the *model* for which one primitive is secure can be reduced to the model for which the other is secure.

As mentioned earlier, the requirement of stateless transformations is necessary. Indeed, if we were to allow stateful transformations, it would be possible to construct particular contrived transformations that trivialize our notion, in the sense that some ideal-cipher construction always reduces to another one. To get a sense of the problem, consider arbitrary constructions A and B which are secure in the ideal-cipher model. Let T denote the stateful transformation which ignores its oracles E, E^{-1} and (efficiently) implements an ideal cipher via lazy sampling. Since B using T is then clearly secure, construction B is—as per Definition 4—reducible to construction A, despite the arbitrary choice of the two constructions. This relies on the statefulness of T in order to implement the lazy sampling procedure. Thus, in order to rule out such trivial cases, we insist that transformations be stateless. This may seem overly restrictive at first glance, but, in fact, is easily justified because blockciphers are inherently stateless entities.

Relationship with indifferentiability. Ideal-cipher reducibility can be seen in relation with reducibility of systems in the indifferentiability framework [MRH04]. In this framework one says system $\mathcal{U}$ reduces to system $\mathcal{V}$ if there is a deterministic B such that for all cryptosystems $\mathcal{C}$ we have that $\mathcal{C}(B(\mathcal{V}))$ is at least as secure as $\mathcal{C}(\mathcal{U})$. Viewing $\mathcal{C}$ as a security game, indifferentiability reducibility can be seen as oracle reducibility with respect to *all* single-stage games simultaneously. In contrast, we are concerned with a set of fixed games. In fact this restriction is hard to avoid, as the PGV compression functions themselves do not behave like a random function given access to E *and* E^{-1}; see [CDMP05, KM07b]. Also, as demonstrated in [RSS11], the origi-

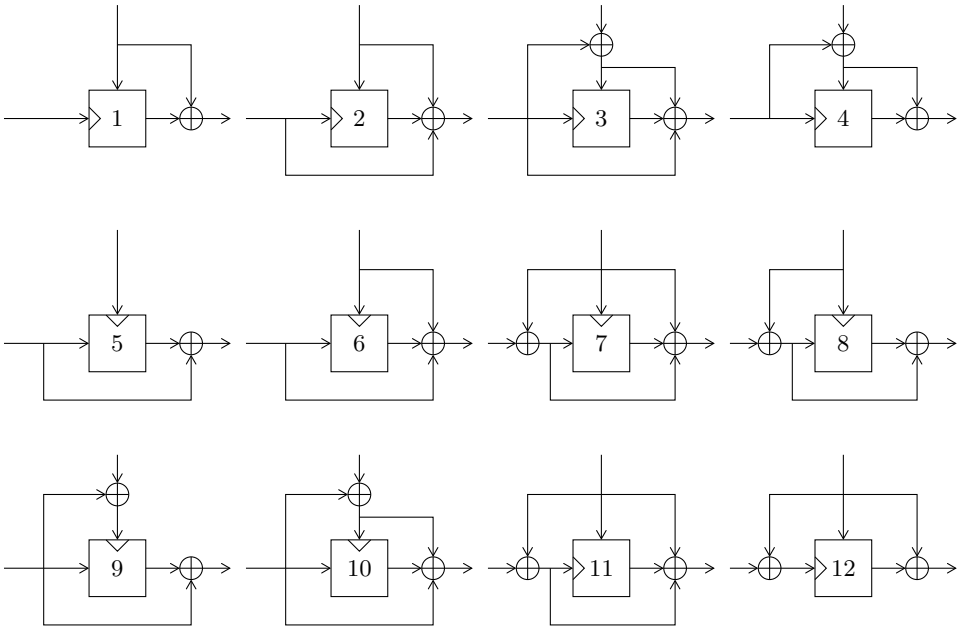


Figure 3.2: The 12 optimally secure PGV constructions $\mathsf{PGV}_i^{\mathsf{E}}$ for $i \in [12]$. A triangle denotes the location of the key input. When used in an iteration mode, the top input is a message block and the left input is the chaining value.

nal indifferentiability framework does not cover arbitrary multi-stage security games well, whereas we can easily cast them in our framework.

3.3 Reducibility among the PGV Functions

We start by recalling the blockcipher-based constructions of hash functions by Preneel et al. [PGV93, BRSS10]. The PGV compression functions rely on a blockcipher $\mathsf{E} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$, and map $\{0,1\}^{2n}$ to $\{0,1\}^n$:

$$\mathsf{PGV}_i^{\mathsf{E}} : \{0,1\}^{2n} \to \{0,1\}^n \quad \text{for} \quad \mathsf{E} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n.$$

There are 64 basic combinations to build such a compression function, of which 12 were first believed [PGV93] (under category "✓" or "FP") and later actually proven to be secure [BRSS10] (under category "group-1"). We denote these secure compression functions by $\mathsf{PGV}_1, \ldots, \mathsf{PGV}_{12}$ and adopt the s-index of [BRSS10, Figure 2]; they are depicted in Figure 3.2. It is worthwhile mentioning that PGV_1 is also known as Matyas–Meyer–Oseas (MMO), PGV_2 as Miyaguchi–Preneel, and PGV_5 as Davies–Meyer (DM). The PGV_1 and PGV_5 functions can be instantiated with a blockcipher whose key length and message

length are not equal. The remaining functions, however, do not natively support this feature but they can be generalized such that they do [Sta09].

For $i \in [12]$ and $q \geq 0$, the security bounds for uniform $\mathcal{E}$ according to [BRS02, Sta09, BRSS10] are

$$\mathbf{Adv}^{\mathsf{coll}}_{\mathsf{PGV}_i,\mathcal{E}}(q) \leq \frac{q^2}{2^n}, \qquad \mathbf{Adv}^{\mathsf{pre}}_{\mathsf{PGV}_i,\mathcal{E}}(q) \leq \frac{2q}{2^n}, \quad \text{and} \quad \mathbf{Adv}^{\mathsf{epre}}_{\mathsf{PGV}_i,\mathcal{E}}(q) \leq \frac{2q}{2^n}.$$

These bounds also hold when the key length and block length are not equal. Furthermore, for uniform $\mathcal{E}$, there exist adversaries $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ making q queries to their E and E^{-1} oracles in total such that [BRSS10]

$$\mathbf{Adv}^{\mathsf{coll}}_{\mathsf{PGV}_i,\mathcal{E}}(\mathcal{A}) \geq \frac{1}{8e}\frac{q^2+1}{2^n}, \qquad \mathbf{Adv}^{\mathsf{pre}}_{\mathsf{PGV}_i,\mathcal{E}}(\mathcal{B}) \geq \frac{q+1}{2^{n+1}},$$

$$\text{and} \quad \mathbf{Adv}^{\mathsf{epre}}_{\mathsf{PGV}_i,\mathcal{E}}(\mathcal{C}) \geq \frac{q+1}{2^{n+1}}.$$

(We introduced the "plus one" terms in order to compactly capture the zero-query special case.) As we will show in the two following theorems, when it comes to ideal-cipher reducibility, the 12 secure PGV constructions can be further partitioned into two subgroups as follows, which we call the PGV_1-group and PGV_2-group, respectively.

$$\mathcal{G}_1 := \{\mathsf{PGV}_1, \mathsf{PGV}_4, \mathsf{PGV}_5, \mathsf{PGV}_8, \mathsf{PGV}_9, \mathsf{PGV}_{12}\}$$
$$\mathcal{G}_2 := \{\mathsf{PGV}_2, \mathsf{PGV}_3, \mathsf{PGV}_6, \mathsf{PGV}_7, \mathsf{PGV}_{10}, \mathsf{PGV}_{11}\}$$

The PGV_1 and PGV_2 functions will be representative of their respective groups.

The following proposition shows that, within a group, the compression functions are ideal-cipher reducible to each other in a direct and tight way (i.e., with the identity transformation and preserving the security bounds). It is worth pointing out that Preneel et al. [PGV93] already discussed equivalence classes from an attack perspective. Present work reaffirms these classes and puts them on a solid theoretical foundation. As noted before, we cannot hope that any PGV compression function construction is indifferentiable from random (given access to E and E^{-1}), so we do not cover this property here; we can, however, include the notion of preimage awareness [DRS09] to the games which are preserved.

Proposition 1. *Any two PGV constructions in $\mathcal{G}_1$ (resp., in $\mathcal{G}_2$) directly and $(1,1,1,1)$-tightly reduce the idealized cipher to each other for the [everywhere] preimage-resistance, collision-resistance, and preimage-awareness games.*

Proof. This is straightforward for [everywhere] preimage resistance and collision resistance. To see this, observe that there is a syntactical one-to-one

correspondence with respect to the inputs within any two functions in each group. Relabeling variables immediately turns any collision (or preimage) for one function into one for the other function. For the sake of concreteness, we consider the collision-resistance ideal-cipher reducibility from $\mathsf{PGV}_5^{\mathcal{E}}$ to $\mathsf{PGV}_1^{\mathcal{E}}$ where for any $(\mathsf{E}, \mathsf{E}^{-1})$ sampled according to $\mathcal{E}$ we have

$$\mathsf{PGV}_1^{\mathsf{E}}(K, M) := \mathsf{E}(K, M) \oplus M \quad \text{and} \quad \mathsf{PGV}_5^{\mathsf{E}}(K, M) := \mathsf{E}(M, K) \oplus K.$$

Assume towards contradiction that there is an adversary which outputs a $\mathsf{PGV}_5^{\mathsf{E}}$ collision $(X, Y) \neq (X', Y')$ for this E. Turn this collision into $(Y, X) \neq (Y', X')$, and output it as a $\mathsf{PGV}_1^{\mathsf{E}}$ collision. It is clear that

$$\mathsf{PGV}_1^{\mathsf{E}}(Y, X) = \mathsf{E}(Y, X) \oplus X = \mathsf{E}(Y', X') \oplus X' = \mathsf{PGV}_1^{\mathsf{E}}(Y', X'),$$

where the inner equality holds whenever the alleged $\mathsf{PGV}_5^{\mathsf{E}}$ adversary succeeds. Since this holds for *any* E the claim also follows in particular for any distribution $\mathcal{E}$ on such blockciphers.

For preimage resistance of the same compression functions, the reduction would again simply turn a preimage (X, Y) into (Y, X).

As for preimage awareness, if an adversary $\mathcal{A}_5$ against $\mathsf{PGV}_5^{\mathsf{E}}$ is able to break preimage awareness by outputting (X, Y), we could easily turn this into an adversary $\mathcal{A}_1$ against $\mathsf{PGV}_1^{\mathsf{E}}$ by returning (Y, X). Any extractor $\mathcal{X}_1$ refuting a successful attack of $\mathcal{A}_1$ could be, vice versa, turned into an extractor $\mathcal{X}_5$ against $\mathcal{A}_5$ by swapping the components of $\mathcal{X}_1$'s outputs.

□

Note that since we can combine the individual games into one, we can conclude that any blockcipher making a scheme from one group secure for all games simultaneously, would also make any other scheme in the group simultaneously secure. Also, the above equivalence still holds for PGV_1 and PGV_5 in case they work with a blockcipher with different key and message length.

The next theorem separates the two groups with respect to the collision-resistance and [everywhere] preimage-resistance games.

Theorem 1. *No PGV construction in $\mathcal{G}_1$ (resp., in $\mathcal{G}_2$) directly reduces to any PGV construction in $\mathcal{G}_2$ (resp., in $\mathcal{G}_1$) for any of the collision-resistance and [everywhere] preimage-resistance games.*

For collision resistance and preimage resistance we assume the ideal cipher, whereas for everywhere preimage resistance we only need the minimal property that there exists *some* blockcipher making the schemes in one group secure, in order to achieve the separation.

Proof. Take PGV_1 and PGV_2 as the representatives of their respective groups. Since all the constructions directly reduce to each other within their group, it

suffices to separate these two constructions; by transitivity a reduction between any other combination would otherwise contradict the fact that PGV_1 and PGV_2 have been separated. Recall that

$$\mathsf{PGV}_1^{\mathsf{E}}(K,M) := \mathsf{E}(K,M) \oplus M \quad \text{and} \quad \mathsf{PGV}_2^{\mathsf{E}}(K,M) := \mathsf{E}(K,M) \oplus M \oplus K.$$

Collision resistance. We first show that the compression functions PGV_1 and PGV_2 do not reduce to each other with respect to collision resistance. In order to prove this, take the ideal cipher $\mathcal{E}$ (with the uniform distribution), which is known to make PGV_2 secure for collision resistance, and let (K_0, M_0) and (K_1, M_1) be from $\{0,1\}^{2n}$ with $K_0 \neq K_1$. We show how to transform any blockcipher E in $\mathsf{Block}(n,n)$ (the support of $\mathcal{E}$) into a new cipher $\tilde{\mathsf{E}}$ such that the induced distribution $\tilde{\mathcal{E}}$ on such blockciphers still makes PGV_2 secure, but for which (K_0, M_0) and (K_1, M_1) form a trivial collision under PGV_1 for any $\tilde{\mathsf{E}}$ sampled from $\tilde{\mathcal{E}}$.

Now for a given blockcipher E and arbitrary (component-wise distinct) points (K_0, M_0) and (K_1, M_1), define

$$C_1' := \mathsf{E}(K_0, M_0) \oplus M_0 \oplus M_1, \quad M_1' := \mathsf{E}^{-1}(K_1, C_1'), \quad C_1 := \mathsf{E}(K_1, M_1),$$

and let $\tilde{\mathsf{E}}$ be the blockcipher identical to E, apart from changing the function value for M_1 under key K_1 to C_1', and redirecting the former's preimage M_1' under key K_1 to C_1:

$$\tilde{\mathsf{E}}(K,M) := \begin{cases} C_1' & \text{if } (K,M) = (K_1, M_1); \\ C_1 & \text{if } (K,M) = (K_1, M_1'); \\ \mathsf{E}(K,M) & \text{otherwise.} \end{cases}$$

$$\tilde{\mathsf{E}}^{-1}(K,C) := \begin{cases} M_1 & \text{if } (K,C) = (K_1, C_1'); \\ M_1' & \text{if } (K,C) = (K_1, C_1); \\ \mathsf{E}^{-1}(K,C) & \text{otherwise.} \end{cases}$$

By inspection, $\tilde{\mathsf{E}}$ is again a blockcipher with inverse $\tilde{\mathsf{E}}^{-1}$. Finding a collision for PGV_1 with respect to any $\tilde{\mathsf{E}}$ chosen from a tweaked distribution as above is easy since

$$\begin{aligned} \mathsf{PGV}_1^{\tilde{\mathsf{E}}}(K_0, M_0) = \tilde{\mathsf{E}}(K_0, M_0) \oplus M_0 = \mathsf{E}(K_0, M_0) \oplus M_0 = C_1' \oplus M_1 = \\ = \tilde{\mathsf{E}}(K_1, M_1) \oplus M_1 = \mathsf{PGV}_1^{\tilde{\mathsf{E}}}(K_1, M_1). \end{aligned}$$

For the analysis of the collision resistance of $\mathsf{PGV}_2^{\tilde{\mathsf{E}}}$ where $\mathcal{E}$ is ideal, we recall the prototype PGV proof from [BRSS10]. This proof concentrates on the probability that an adversary creates its first collision on the ith query and subsequently uses a union bound to combine these stepwise probabilities. For this proof all that is needed (to bound the probability of a success at step i) is

(a) that the ith query corresponds to a single compression function evaluation that (over the randomness of the query's answer) is uniformly distributed over a set of size at least $2^n - i$, and (b) that the adversary only knows at most i compression function evaluations prior to making query i. When using $\tilde{\mathsf{E}}$ instead of a sample E from the ideal cipher, we need to take into account that we have introduced a dependency among the points (K_0, M_0), (K_1, M_1), and (K_1, M_1'). We do this by giving these three queries *for free* to the adversary at the beginning of the collision-finding game. If these three points do not cause a collision among themselves, then the original proof goes through as from that moment onwards, (a) $\tilde{\mathsf{E}}$ is identically distributed to the ideal cipher E and (b) the free queries just resulted in three extra compression function evaluations.

For the tweaked points, we look at the $\binom{3}{2} = 3$ possible colliding pairs. Let $C_0 := \mathsf{E}(K_0, M_0)$.

1. The first case is:

$$\begin{aligned} & \mathsf{PGV}_2^{\tilde{\mathsf{E}}}(K_0, M_0) = \mathsf{PGV}_2^{\tilde{\mathsf{E}}}(K_1, M_1) \\ \iff & C_0 \oplus M_0 \oplus K_0 = C_1' \oplus M_1 \oplus K_1 \\ \iff & K_0 = K_1, \end{aligned}$$

   which happens with probability 0 since $K_0 \neq K_1$.

2. The second case is:

$$\mathsf{PGV}_2^{\tilde{\mathsf{E}}}(K_0, M_0) = \mathsf{PGV}_2^{\tilde{\mathsf{E}}}(K_1, M_1') \iff C_0 \oplus M_0 \oplus K_0 = C_1 \oplus M_1' \oplus K_1.$$

   Adding $C_1 \oplus K_1$ to both sides and enciphering with E under K_1 we get that the equation is equivalent to

$$\mathsf{E}(K_1, C_0 \oplus K_0 \oplus C_1 \oplus K_1 \oplus M_0) = \mathsf{E}(K_0, M_0) \oplus M_0 \oplus M_1.$$

   Since $K_0 \neq K_1$ it is clear that the probability of equality is $1/2^n$ as the values of the E on the two sides of the equation are independently and uniformly distributed.

3. The third case is:

$$\mathsf{PGV}_2^{\tilde{\mathsf{E}}}(K_1, M_1) = \mathsf{PGV}_2^{\tilde{\mathsf{E}}}(K_1, M_1') \iff C_1' \oplus M_1 \oplus K_1 = C_1 \oplus M_1' \oplus K_1,$$

   which after rearranging as in the previous case becomes equivalent to

$$\mathsf{E}(K_1, C_0 \oplus C_1 \oplus M_0) = \mathsf{E}(K_0, M_0) \oplus M_0 \oplus M_1.$$

   Once again, since $K_0 \neq K_1$, we have that the probability of a collision is $1/2^n$.

This proves that the idealized cipher $\tilde{\mathcal{E}}$ makes PGV_2 collision resistant.

For the converse separation, start with the ideal blockcipher $\mathcal{E}'$, which makes PGV_1 secure. For any E' in the support of $\mathcal{E}'$ consider the blockcipher E with $\mathsf{E}(K,M) = \mathsf{E}'(K,M) \oplus K$ and $\mathsf{E}^{-1}(K,C) = \mathsf{E}'^{-1}(K, C \oplus K)$. Since

$$\mathsf{PGV}_2^{\mathsf{E}}(K,M) = \mathsf{E}(K,M) \oplus K \oplus M = \mathsf{E}'(K,M) \oplus M = \mathsf{PGV}_1^{\mathsf{E}'}(K,M)$$

this distribution $\mathcal{E}$ on blockciphers E now makes PGV_2 secure. Furthermore, $\mathcal{E}$ itself is again the uniform distribution on all blockciphers. Run the same transformation from E to $\tilde{\mathsf{E}}$ as above, such that $\mathsf{PGV}_2^{\tilde{\mathsf{E}}}$ remains secure, whereas $\mathsf{PGV}_1^{\tilde{\mathsf{E}}}$ is easy to break. Apply now once more the idea of adding the key to the cipher's output and define $\tilde{\mathsf{E}}'$ through

$$\tilde{\mathsf{E}}'(K,M) = \tilde{\mathsf{E}}(K,M) \oplus K, \quad \text{and} \quad \tilde{\mathsf{E}}'^{-1}(K,C) = \tilde{\mathsf{E}}^{-1}(K, C \oplus K),$$

such that again

$$\mathsf{PGV}_2^{\tilde{\mathsf{E}}'}(K,M) = \mathsf{PGV}_1^{\tilde{\mathsf{E}}}(K,M), \quad \text{and} \quad \mathsf{PGV}_2^{\tilde{\mathsf{E}}}(K,M) = \mathsf{PGV}_1^{\tilde{\mathsf{E}}'}(K,M).$$

We conclude that the distribution on blockciphers $\tilde{\mathsf{E}}'$ now makes PGV_1 collision resistant, but any blockcipher allows to find collisions for PGV_2 easily. This proves the separation in the other direction.

Everywhere preimage resistance. For everywhere preimage resistance it is convenient to start with an arbitrary (not necessarily ideal) distribution on blockciphers E which makes PGV_1 secure. We tweak every such E to $\tilde{\mathsf{E}}$ by setting

$$\tilde{\mathsf{E}}(K,M) := \begin{cases} M \oplus K & \text{if } K = \mathsf{E}(0^n, 0^n); \\ \mathsf{E}(K,M) & \text{otherwise.} \end{cases}$$

$$\tilde{\mathsf{E}}^{-1}(K,C) := \begin{cases} C \oplus K & \text{if } K = \mathsf{E}(0^n, 0^n); \\ \mathsf{E}^{-1}(K,C) & \text{otherwise.} \end{cases}$$

Obviously $\tilde{\mathsf{E}}$ together with $\tilde{\mathsf{E}}^{-1}$ constitute a blockcipher.

First observe that we can assume $\mathsf{E}(0^n, 0^n) \neq 0^n$, or else any adversary pair outputting 0^n in the first stage and $(0^n, 0^n)$ in the second stage would refute everywhere preimage resistance for $\mathsf{PGV}_1^{\mathsf{E}}$. Hence, the probability that $\mathsf{E}(0^n, 0^n) = 0^n$ must be negligible, and from now on we condition on this event not happening. We can now show that $\mathsf{PGV}_2^{\tilde{\mathsf{E}}}$ is not secure. For this, let $\mathcal{A}_1(1^n)$ output 0^n, and let $\mathcal{A}_2^{\tilde{\mathsf{E}}, \tilde{\mathsf{E}}^{-1}}(0^n)$ return $(K,M) := (\tilde{\mathsf{E}}(0^n, 0^n), 0^n)$. Then, since $\tilde{\mathsf{E}}(0^n, 0^n) = \mathsf{E}(0^n, 0^n)$ by assumption about $\mathsf{E}(0^n, 0^n) \neq 0^n$, we conclude that

$$\begin{aligned} \mathsf{PGV}_2^{\tilde{\mathsf{E}}}(K,M) &= \tilde{\mathsf{E}}(K,M) \oplus K \oplus M \\ &= \tilde{\mathsf{E}}(\mathsf{E}(0^n, 0^n), 0^n) \oplus \tilde{\mathsf{E}}(0^n, 0^n) \\ &= (\tilde{\mathsf{E}}(0^n, 0^n) \oplus 0^n) \oplus \tilde{\mathsf{E}}(0^n, 0^n) \\ &= 0^n. \end{aligned}$$

Hence, the adversary pair always finds an image/preimage pair with a single query to $\tilde{\mathsf{E}}$.

Next we show that any pair $(\mathcal{A}_1, \mathcal{A}_2)$ against PGV_1 for $\tilde{\mathsf{E}}$ can be immediately turned into a pair against PGV_1 for E. Assume that $\mathcal{A}_1(1^n)$ returns some (Y, st), and that $\mathcal{A}_2^{\tilde{\mathsf{E}},\tilde{\mathsf{E}}^{-1}}(Y, \mathrm{st})$ finds (K, M) such that $\mathsf{PGV}_1^{\tilde{\mathsf{E}}}(K, M) = Y$. There are two cases: If $K = \mathsf{E}(0^n, 0^n)$, then letting the second adversary $\mathcal{A}_2^{\mathsf{E},\mathsf{E}^{-1}}$ (now against E) return $(0^n, 0^n)$ would yield a preimage of Y under $\mathsf{PGV}_1^{\mathsf{E}}$, because then

$$Y = \mathsf{PGV}_1^{\tilde{\mathsf{E}}}(K, M) = M \oplus \mathsf{E}(0^n, 0^n) \oplus M = \mathsf{E}(0^n, 0^n) = \mathsf{PGV}_1^{\mathsf{E}}(0^n, 0^n).$$

In the other case, i.e., when $K \neq \mathsf{E}(0^n, 0^n)$, it is clear that (K, M) is also a preimage under $\mathsf{PGV}_1^{\mathsf{E}}$. Hence, either case must be negligible, and $\mathsf{PGV}_1^{\tilde{\mathsf{E}}}$ must be secure.

For the converse separation, as in the case of collision resistance, we apply the technique of adding the key once to the innermost blockcipher, and another time to the outer blockcipher. This leads to a separating example for PGV_2 from PGV_1 for the everywhere preimage-resistance game.

Preimage resistance. Finally we treat the case of preimage resistance. Given a blockcipher E sampled from the uniform distribution, we let $M_{K,0} := \mathsf{E}^{-1}(K, 0^n)$, $C_{K,0} := \mathsf{E}(K, 0^n)$ for each key K, and define a tweaked blockcipher $\tilde{\mathsf{E}}$ as follows.

$$\tilde{\mathsf{E}}(K, M) := \begin{cases} 0^n & \text{if } M = 0^n; \\ C_{K,0} & \text{if } M = M_{K,0}; \\ \mathsf{E}(K, M) & \text{otherwise.} \end{cases}$$

$$\tilde{\mathsf{E}}^{-1}(K, C) := \begin{cases} 0^n & \text{if } C = 0^n; \\ M_{K,0} & \text{if } C = C_{K,0}; \\ \mathsf{E}^{-1}(K, C) & \text{otherwise.} \end{cases}$$

Note that $\mathsf{PGV}_2^{\tilde{\mathsf{E}}}(K, 0^n) = 0^n \oplus 0^n \oplus K = K$. Hence, any adversary which on input Y outputs $(Y, 0^n)$ succeeds with probability 1 in the preimage-resistance game for $\mathsf{PGV}_2^{\tilde{\mathsf{E}}}$. It remains to show that $\mathsf{PGV}_1^{\tilde{\mathsf{E}}}$ is preimage resistant. An adversary cannot succeed by outputting a pair $(K, 0^n)$ since $\mathsf{PGV}_1^{\tilde{\mathsf{E}}}(K, 0^n) = 0^n$, which would arise as a challenge value with only a negligible probability. Similarly, the challenge digest will originate from $(K, M_{K,0})$ for some K with probability 2^{-n} only. Hence any preimage-resistance adversary must either attack PGV_1 with respect to the original cipher E (which we know to be secure) or recover a preimage using the second branch of $\tilde{\mathsf{E}}$, i.e., output a preimage $(K, M_{K,0})$ for

$$\mathsf{PGV}_1^{\tilde{\mathsf{E}}}(K, M_{K,0}) = C_{K,0} \oplus M_{K,0} = \mathsf{E}(K, 0^n) \oplus \mathsf{E}^{-1}(K, 0^n)$$

for some K. Since E is sampled from the ideal cipher, the two summands are uniformly and independently distributed for each K (unless K is queried). Thus, for a given target digest, any attacker will only have a negligible success probability to recover a preimage of this form.

Applying the transformation which reduces $\mathcal{G}_1$ to $\mathcal{G}_2$ to the cipher $\tilde{\mathsf{E}}$ we obtain an idealized cipher under which PGV_1 is not preimage resistant but PGV_1 is.

□

Proposition 2. *Any two PGV constructions* PGV_i *and* PGV_j *for* $i, j \in [12]$ $(1, 1, 1, 1)$*-tightly reduce the idealized cipher to each other for the [everywhere] preimage-resistance and collision-resistance games (under free transformations).*

To prove this, we first show that there is a transformation such that there is an inter-group reduction, i.e., $\mathsf{PGV}_2 \in \mathcal{G}_2$ reduces to $\mathsf{PGV}_1 \in \mathcal{G}_1$ and vice versa—indeed, we will use the same transformation for either direction. By transitivity we then obtain a reduction for any two constructions through Proposition 1, where we may view the identity transformation as a special case of an arbitrary one.

Proof. Consider PGV_1 and PGV_2. We claim that for the transformation defined through

$$\mathsf{T}^{\mathsf{E}}(K, M) := \mathsf{E}(K, M) \oplus K \quad \text{and} \quad \mathsf{T}^{-1^{\mathsf{E}^{-1}}}(K, C) := \mathsf{E}^{-1}(K, C \oplus K),$$

the security of $\mathsf{PGV}_2^{\mathsf{T}^{\mathsf{E}}}$ reduces to $\mathsf{PGV}_1^{\mathsf{E}}$. This is because both compression functions are identical for any E, implying that the idealized cipher $\mathsf{T}^{\mathcal{E}}$ reduces to the idealized cipher $\mathcal{E}$. This can be easily done vice versa, too, for the same transformation, noting that applying T twice is the identity transformation. Observe that T^{E} is indeed a permutation for any fixed key K; the statement now trivially follows. □

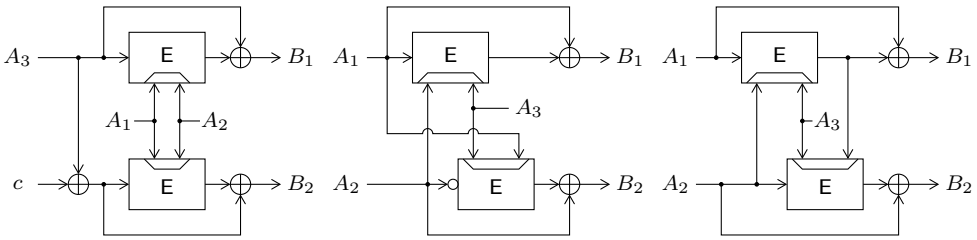
This concludes the treatment of the relations among the PGV functions.

3.4 Double-Block-Length Hashing and PGV

We now widen our scope to the double-block-length compression function designs, comparing them to the PGV functions and one to another.

Reducibility from DBL to PGV

We study the relation between three prominent double-block-length hash function constructions in the literature, namely, Hirose-DM [Hir04, Hir06], Abreast-DM [LM92, LK11], and Tandem-DM [LM92, LSS11, FGL09a], and the



(a) Hirose-DM (b) Abreast-DM (c) Tandem-DM

Figure 3.3: The three double-block-length compression functions. The hollow circle in Abreast-DM denotes bitwise complement.

PGV constructions. All the DBL compression functions under consideration here map $3n$-bit inputs to $2n$-bit outputs, and rely on a blockcipher with $2n$-bit keys and n-bit block. More precisely, these constructions are of the form

$$\mathsf{F}^{\mathsf{E}} : \{0,1\}^{3n} \to \{0,1\}^{2n} \quad \text{where} \quad \mathsf{E} : \{0,1\}^{2n} \times \{0,1\}^{n} \to \{0,1\}^{n}.$$

We denote the Hirose-DM for a constant $c \in \{0,1\}^n \setminus \{0^n\}$, the Abreast-DM, and the Tandem-DM compression functions by HDM_c, ADM, and TDM, respectively. These functions are defined as follows (see Figure 3.3 for pictorial representations).

$$\mathsf{HDM}_c^{\mathsf{E}}(A_1, A_2, A_3) := (\mathsf{E}(A_1|A_2, A_3) \oplus A_3, \mathsf{E}(A_1|A_2, A_3 \oplus c) \oplus A_3 \oplus c)$$
$$\mathsf{ADM}^{\mathsf{E}}(A_1, A_2, A_3) := \left(\mathsf{E}(A_2|A_3, A_1) \oplus A_1, \mathsf{E}(A_3|A_1, \overline{A_2}) \oplus A_2\right)$$
$$\mathsf{TDM}^{\mathsf{E}}(A_1, A_2, A_3) := (\mathsf{E}(A_2|A_3, A_1) \oplus A_1, \mathsf{E}(A_3|\mathsf{E}(A_2|A_3, A_1), A_2) \oplus A_2)$$

The following proposition shows that collisions (resp., somewhere preimages) in the double-block-length functions directly lead to collisions (resp., somewhere preimages) for the double-key versions of PGV_1 and PGV_5 functions; this shows that the DBL functions reduce to the PGV functions.

Proposition 3. *The idealized ciphers in* HDM_c*, for any* $c \in \{0,1\}^n \setminus \{0^n\}$*,* ADM*, and* TDM *compression functions directly and* $(1,1,1,1)$*-tightly reduce to those in the (double-key versions of the)* PGV_1 *and* PGV_5 *functions for the everywhere preimage-resistance and collision-resistance games.*

Proof. We only treat the case of PGV_1 as reducibility to PGV_5 is proved similarly. Note that the first component of any of the DBL constructions is a PGV_1 value (up to relabeling of the variables). This means that any adversary breaking the collision resistance of, say, HDM_c can be used to break the collision resistance of PGV_1. A similar argument applies to the everywhere preimage-resistance game. We take the output of a first-stage adversary which

returns an image value for HDM_c and pass its first component out as the candidate image point for PGV_1. When the second stage of the adversary outputs a preimage, we also use it as our own guess. □

Note that despite the tightness of the reduction, a blockcipher that makes the schemes PGV_1 and PGV_5 ideally secure is not guaranteed to make the double-block-length compression functions secure beyond the implied single-length security bound.

Curiously, the above argument fails for the preimage-resistance game as we cannot extend a challenge value for PGV_1 to a full challenge value for a DBL construction. Indeed, we prove in the next theorem that there is no direct reduction with respect to preimage resistance.

Proposition 4. *The idealized cipher in none of the DBL constructions directly reduces to the idealized cipher in* PGV_1 *(and hence neither to the one in* PGV_5*) for the (standard) preimage-resistance game.*

Direct ideal-cipher reducibility to the other PGV constructions is not syntactically possible as only the PGV_1 and PGV_5 constructions can be natively instantiated with a double-block-length blockcipher. Note that the above proposition leaves open the (im)possibility of free reductions from DBL to PGV, which we consider an interesting open problem.

Proof. Let us start with separating HDM_{1^n} (we briefly discuss how to extend the separation to HDM_c for other nonzero values of c at the end). Recall that there is a natural embedding of $\{0,1\}^{n-1}$ in $\mathrm{GF}(2^{n-1})$ where field addition corresponds to computing exclusive-or, and field multiplication is performed modulo a fixed irreducible polynomial. For an $\alpha \in \mathrm{GF}(2^{n-1}) \setminus \{0,1\}$, we define a distribution on $\mathsf{Block}(2n,n)$ by picking a cipher $\mathsf{E} \leftarrow\!\!\$\ \mathsf{Block}(n/2-1, n/2-1)$, ignoring the key, and essentially enciphering either the left or the right half of the input block, depending on the most significant bit of the input. That is, we parse the input M as $m_1|M_1|m_2|M_2$, where m_i are bits and M_i are of length $n/2-1$, and set

$$\tilde{\mathsf{E}}(K, m_1|M_1|m_2|M_2) := \begin{cases} 0|\mathsf{E}(0^{n/2-1}, M_1)|m_2|(\alpha M_2) & \text{if } m_1 = 0; \\ 1|\mathsf{E}(0^{n/2-1}, M_2)|m_2|(\alpha M_1) & \text{otherwise.} \end{cases}$$

$$\tilde{\mathsf{E}}^{-1}(K, c_1|C_1|c_2|C_2) := \begin{cases} 0|\mathsf{E}^{-1}(0^{n/2-1}, C_1)|c_2|(\alpha^{-1} C_2) & \text{if } c_1 = 0; \\ 1|(\alpha^{-1} C_2)|c_2|\mathsf{E}^{-1}(0^{n/2-1}, C_1) & \text{otherwise.} \end{cases}$$

It is not too difficult to check that $\tilde{\mathsf{E}}$ and $\tilde{\mathsf{E}}^{-1}$ as above define a blockcipher. To see that HDM_{1^n} is not preimage resistant with respect to the distribution on such $\tilde{\mathsf{E}}$, note that with probability $1/4$ in the preimage-resistance game we

have that $m_1 = m_2 = 0$, in which case

$$\mathsf{HDM}^{\tilde{\mathsf{E}}}_{1^n}(A_1, A_2, 0|M_1|0|M_2) = \Big(0|(\mathsf{E}(0^{n/2-1}, M_1) \oplus M_1)|0|(\alpha M_2 \oplus M_2),\\ 1|(\mathsf{E}(0^{n/2-1}, \overline{M_2}) \oplus \overline{M_1})|1|(\alpha\overline{M_1} \oplus \overline{M_2})\Big)\,.$$

Now given a preimage-resistance challenge value as shown above, we can recover M_2 from the second part of the first component, $(\alpha + 1)M_2$. Note that here we use that $\alpha \neq 1$ and thus $\alpha + 1 \neq 0$ over the field of characteristic 2. Then using M_2 and the second part of the second component we can also recover M_1. The tuple $(0^n, 0^n, 0|M_1|0|M_2)$ is a valid preimage (note that A_1 and A_2 do not affect the value of the compression function).

It remains to show that $\mathsf{PGV}^{\tilde{\mathsf{E}}}_1$ for such distributed blockciphers $\tilde{\mathsf{E}}$ is preimage resistant. Note that

$$\mathsf{PGV}^{\tilde{\mathsf{E}}}_1(K, m_1|M_1|m_2|M_2) := \begin{cases} 0|(\mathsf{E}_0(M_1) \oplus M_1)|0|(\alpha M_2 \oplus M_2) & \text{if } m_1 = 0; \\ 0|(\mathsf{E}_0(M_2) \oplus M_1)|0|(\alpha M_1 \oplus M_2) & \text{otherwise,} \end{cases}$$

where $\mathsf{E}_0(M)$ is a shorthand for $\mathsf{E}(0^{n/2-1}, M)$. For preimage resistance, observe that the K and m_2 inputs and the 0s in the output can be discarded (cf. [Sta08, Lemma 3]), so for the preimage resistance of $\mathsf{PGV}^{\tilde{\mathsf{E}}}_1$ we can instead regard the two functions

$$\mathsf{F}^{\mathsf{E}}_0(M_1|M_2) := (\mathsf{E}(0^{n/2-1}, M_1) \oplus M_1)|(\alpha M_2 \oplus M_2),$$
$$\mathsf{F}^{\mathsf{E}}_1(M_1|M_2) := (\mathsf{E}(0^{n/2-1}, M_2) \oplus M_1)|(\alpha M_1 \oplus M_2).$$

Using techniques similar to those from [BRSS10, Section 10], one can prove that for either function the uniform distribution for $(M_1|M_2)$ together with the uniform distribution $\mathcal{E}$ for E, induce a close to uniform distribution over the possible challenge digests. Consequently, if both $\mathsf{F}^{\mathsf{E}}_0$ and $\mathsf{F}^{\mathsf{E}}_1$ are everywhere preimage resistant, then $\mathsf{PGV}^{\mathsf{E}}_1$ is preimage resistant (as the adversary against $\mathsf{PGV}^{\mathsf{E}}_1$ needs to find a preimage of a randomly selected digest under either $\mathsf{F}^{\mathsf{E}}_0$ or $\mathsf{F}^{\mathsf{E}}_1$). For the preimage resistance of $\mathsf{F}^{\mathsf{E}}_0$ it suffices to observe that $M_1 \mapsto \mathsf{E}(0^{n/2-1}, M_1) \oplus M_1$ is well known to be everywhere preimage resistant (e.g., [Sta09, Theorem 6]) as appending $(\alpha M_2 \oplus M_2)$ does not affect the security (it is independent of M_1). To prove that $\mathsf{F}^{\mathsf{E}}_1$ is also preimage resistant, we start by considering the auxiliary compression function

$$\mathsf{F}^{\mathsf{E}}(M) := \mathsf{E}(0^k, M) \oplus (\alpha^{-1} \cdot M) \quad \text{for} \quad \alpha \neq 0.$$

This function is preimage resistant for an ideally distributed E (which follows from [Sta09, Theorem 6]). We now show that any preimage-resistance adversary $\mathcal{A}$ against $\mathsf{F}^{\mathsf{E}}_1$ can be used to break the preimage resistance of F^{E}. Given a challenge value Z for F^{E}, choose $Y_2 \leftarrow\!\!\$\ \{0,1\}^{n/2-1}$, set $Y_1 := Z \oplus Y_2$, and run $\mathcal{A}(0|Y_1, 0|\alpha Y_2)$. By a simple code expansion, the challenge value $(Y_1, \alpha Y_2)$

can be seen as being generated by choosing a random (K, M_2) and computing $(\mathsf{E}(0^k, M_2) \oplus M_1, \alpha M_1 \oplus M_2)$ where $M_1 := \alpha^{-1} M_2 \oplus Y_2$. Note that M_1 is uniformly distributed and is independent of (K, M_2). Hence when $\mathcal{A}$ returns a successful preimage (M_1, M_2), the second component, M_2, would be a valid preimage for Z.

We briefly discuss how to extend the above argument to HDM_c for other nonzero values of c. To this end, we need to ensure that adding c in the second component has the same effect of flipping the first bit of the input as above. We do this by first noting the position, i_c, of the most significant nonzero bit of c. Instead of differentiating the two branches of the cipher based on m_1 we do this by inspecting m_{i_c} and leak this bit accordingly. The remaining bits are then used to form what was M_1 and M_2 before.

We now give an idealized cipher separating the preimage resistance of ADM from that of PGV_1. For any blockcipher in $\mathsf{Block}(n/2, n/2)$, define the function $f^{\mathsf{E}}(X) := \mathsf{E}(0^{n/2}, X) \oplus X$. It is straightforward to show that this function is one way in the presence of E and E^{-1} oracles sampled uniformly from $\mathsf{Block}(n/2, n/2)$. With notation as in the previous example, and denoting the most significant bit of K by $K[0]$, based on f^{E} we define the following blockcipher.

$$\tilde{\mathsf{E}}(K_{11}|K_{12}|K_{21}|K_{22}, M_1|M_2) := \begin{cases} (f^{\mathsf{E}}(K_{12}) \oplus M_1)|\alpha M_2 & \text{if } K_{11}[0] = 0; \\ (f^{\mathsf{E}}(K_{22}) \oplus M_1)|\alpha M_2 & \text{if } K_{11}[0] = 1. \end{cases}$$

$$\tilde{\mathsf{E}}^{-1}(K_{11}|K_{12}|K_{21}|K_{22}, C_1|C_2) := \begin{cases} (f^{\mathsf{E}}(K_{12}) \oplus C_1)|\alpha^{-1} C_2 & \text{if } K_{11}[0] = 0; \\ (f^{\mathsf{E}}(K_{22}) \oplus C_1)|\alpha^{-1} C_2 & \text{if } K_{11}[0] = 1. \end{cases}$$

Observe that $\tilde{\mathsf{E}}$ and $\tilde{\mathsf{E}}^{-1}$ as above define a permutation for each key and hence constitute a blockcipher. Let us now look at $\mathsf{ADM}^{\tilde{\mathsf{E}}}$ values conditioned on the event that $A_{21}[0] = 0 \wedge A_{31}[0] = 1$ which occurs with probability $1/4$ for randomly chosen A_2 and A_3:

$$\mathsf{ADM}^{\tilde{\mathsf{E}}}(A_{11}|A_{12}, A_{21}|A_{22}, A_{31}|A_{32}) = \Big(f^{\mathsf{E}}(A_{22})|(\alpha A_{12} \oplus A_{12}), \\ f^{\mathsf{E}}(A_{12}) \oplus 1^{n/2}|(\alpha \overline{A_{22}} \oplus A_{22})\Big).$$

Clearly $\mathsf{ADM}^{\tilde{\mathsf{E}}}$ is not preimage resistant in this case as all the values on which the compression depends can be read off from the digest value. More specifically, given such a value in the preimage-resistance game, the point $(0^{n/2}|A_{12}, 0^{n/2}|A_{22}, 0^n)$ is a valid preimage. To see that $\mathsf{PGV}_1^{\tilde{\mathsf{E}}}$ is preimage resistant for such blockciphers (over the choice of E) observe that any successful preimage-resistance adversary can be immediately used to invert f^{E}, which we have discussed is one way in the ideal-cipher model.

Finally, it turns out that the above blockcipher also separates the preimage resistance of TDM from that of PGV_1: whenever $A_{21}[0] = 0$ and $A_{31}[0] = 1$ (which happens, again, with probability $1/4$ in the preimage game) we have that

$$\mathsf{TDM}^{\tilde{\mathsf{E}}}(A_{11}|A_{12}, A_{21}|A_{22}, A_{31}|A_{32}) = \Big(f^{\mathsf{E}}(A_{22})|(\alpha A_{12} \oplus A_{12}), \\ f^{\mathsf{E}}(\alpha A_{12})|(\alpha\overline{A_{22}} \oplus A_{22})\Big),$$

from which a preimage value can be readily computed since α is public. For the sake of concreteness, a preimage is given by $(0^{n/2}|A_{12}, 0^{n/2}|A_{22}, 0^n)$. The fact that the distribution of blockciphers $\tilde{\mathsf{E}}$ preserves preimage resistance for PGV_1 has been shown before, concluding the proof.

□

We next show that under *free* transformations a double-block-length instantiation of PGV_1 reduces to a single-block-length instantiation of PGV_1. By the transitivity of reductions we obtain reducibility of the idealized cipher in the DBL constructions to that in any of the PGV constructions.

Proposition 5. *The idealized cipher in* PGV_1 *instantiated with an idealized cipher in* $\mathsf{Ideal}(2n, n)$ $(2, 2, 1, 1)$*-tightly reduces to the one in* PGV_1 *when instantiated with an idealized cipher in* $\mathsf{Ideal}(n, n)$ *for the everywhere preimage-resistance and collision-resistance games.*

Proof. We define the required transformation as follows.

$$\mathsf{T}^{\mathsf{E},\mathsf{E}^{-1}}(K_1|K_2, M) := \mathsf{E}(\mathsf{E}(K_1, K_2) \oplus K_2, M)$$
$$\mathsf{T}^{-1\,\mathsf{E},\mathsf{E}^{-1}}(K_1|K_2, C) := \mathsf{E}^{-1}(\mathsf{E}(K_1, K_2) \oplus K_2, C)$$

Note that the above transformed blockcipher, when used in PGV_1 with twice the key length, yields a fixed-length Merkle–Damgård (MD) iteration using a random initialization vector of PGV_1 for cipher E (with single key length):

$$\mathsf{PGV}_1^{\mathsf{T}^{\mathsf{E}}}(K_1|K_2, M) = \mathsf{PGV}_1^{\mathsf{E}}(\mathsf{PGV}_1^{\mathsf{E}}(K_1, K_2), M).$$

As shown in, say [ANPS07], this MD chaining preserves both collision resistance and everywhere preimage resistance of $\mathsf{PGV}_1^{\mathsf{E}}$ (but requires two blockcipher calls per evaluation). This proves the proposition. □

Remark. Although Merkle–Damgård chaining does *not* in general preserve the preimage resistance of the underlying compression function, there exist more sophisticated chaining schemes, such as ROX [ANPS07], which do so. If such chaining rules are used to compress the keys in the proposition above, we also obtain reducibility for the preimage-resistance game.

Separations among the DBL compression functions

We now investigate direct reducibility among the DBL compression functions, as well as reducibility between PGV_1 and DBL functions. We focus on collision resistance, but similar techniques (for separations) may be applicable to the other security games. For this game, there are twelve relations to be considered, three of which have already been settled by Proposition 3. We study the remaining relations by providing separations among all the possible pairs. In doing so, we give blockciphers E such that one of the DBL constructions (and hence by Proposition 3 the PGV_1 function, too) admits a trivial collision, whereas the other two constructions are *simultaneously* secure.

We start with the HDM_c compression function where $c \neq 0^n$. Let E be a blockcipher. Define a modified blockcipher $\tilde{\mathsf{E}}$ as follows.

$$M_c := \mathsf{E}^{-1}(0^n|0^n, \mathsf{E}(0^n|0^n, 0^n) \oplus c), \quad C_0 := \mathsf{E}(0^n|0^n, 0^n), \quad C_c := \mathsf{E}(0^n|0^n, c).$$

$$\tilde{\mathsf{E}}(K_1|K_2, M) := \begin{cases} C_0 \oplus c & \text{if } (K_1|K_2, M) = (0^n|0^n, c); \\ C_c & \text{if } (K_1|K_2, M) = (0^n|0^n, M_c); \\ \mathsf{E}(K_1|K_2, M) & \text{otherwise.} \end{cases}$$

$$\tilde{\mathsf{E}}^{-1}(K_1|K_2, C) := \begin{cases} c & \text{if } (K_1|K_2, C) = (0^n|0^n, C_0 \oplus c); \\ M_c & \text{if } (K_1|K_2, C) = (0^n|0^n, C_c); \\ \mathsf{E}^{-1}(K_1|K_2, C) & \text{otherwise.} \end{cases}$$

Note that $\tilde{\mathsf{E}}$ and $\tilde{\mathsf{E}}^{-1}$ above define a blockcipher and we have $c \neq 0^n$. Hence,

$$\begin{aligned} \mathsf{HDM}_c^{\tilde{\mathsf{E}}}(0^n, 0^n, 0^n) &= (\tilde{\mathsf{E}}(0^n|0^n, 0^n) \oplus 0^n, \tilde{\mathsf{E}}(0^n|0^n, c) \oplus c) \\ &= (C_0, C_0 \oplus c \oplus c) \\ &= (C_0, C_0), \\ \mathsf{HDM}_c^{\tilde{\mathsf{E}}}(0^n, 0^n, c) &= (\tilde{\mathsf{E}}(0^n|0^n, c) \oplus c, \tilde{\mathsf{E}}(0^n|0^n, 0^n) \oplus 0^n) \\ &= (C_0 \oplus c \oplus c, C_0) \\ &= (C_0, C_0). \end{aligned}$$

and the pair $((0^n, 0^n, 0^n), (0^n, 0^n, c))$ thus constitutes a non-trivial collision for $\mathsf{HDM}_c^{\tilde{\mathsf{E}}}$. However, the next lemma shows that ADM and TDM remain collision resistant for this cipher.

Lemma 1. *Let $\tilde{\mathsf{E}}$ be a blockcipher as above with a distribution according to $(\mathsf{E}, \mathsf{E}^{-1}) \leftarrow\!\!\$\ \mathsf{Block}(2n, n)$. Then $\mathsf{ADM}^{\tilde{\mathsf{E}}}$ and $\mathsf{TDM}^{\tilde{\mathsf{E}}}$ are both collision resistant.*

Proof. We consider ADM where we first recall the existing proof of collision resistance in the ideal-cipher model by Lee and Kwon [LK11]. We will argue that with only minor modifications, their proof goes through also for the

almost-ideal cipher $\tilde{\mathsf{E}}$. The original proof relies on the observation that queries to the blockcipher can be grouped into cycles by taking into account how they can be used to evaluate the ADM compression function. Suppose an adversary wants to evaluate $\mathsf{ADM}^{\mathsf{E}}(A_1, A_2, A_3)$. This requires the queries $\mathsf{E}(A_2|A_3, A_1)$ and $\mathsf{E}(A_3|A_1, \overline{A_2})$ to be made. Here the second query is intended for the evaluation of the lower half of Fig. 3.3(b), but it could also be used in the upper half, as part of the evaluation of $\mathsf{ADM}^{\mathsf{E}}(\overline{A_2}, A_3, A_1)$. In that case, the lower-half query would be $\mathsf{E}(A_1|\overline{A_2}, \overline{A_3})$. Now this query could also be used in the upper half, leading to lower-half query $\mathsf{E}(\overline{A_2}|\overline{A_3}, \overline{A_1})$. It might seem that this could go on for a while, but after $\mathsf{E}(\overline{A_3}|\overline{A_1}, A_2)$ and $\mathsf{E}(\overline{A_1}|A_2, A_3)$, the next query in this sequence is $\mathsf{E}(A_2|A_3, A_1)$ which we already saw at the very beginning. Thus after at most six steps the cycle is complete; moreover, when distinct, the six blockcipher queries within a cycle uniquely determine six ADM compression function evaluation and they are not used for any other ADM evaluations. This observation is used in the proof by limiting a collision-finding adversary to querying full cycles only: whenever he makes a query, he will get the remaining queries in the cycle for free. For this modified adversary, Lee and Kwon subsequently bound both the probability of finding a collision within a single cycle and the probability of finding a collision between cycles.

For the analysis of the collision resistance of $\mathsf{ADM}^{\tilde{\mathsf{E}}}$ (where E is ideal) we need to take into account possible interdependencies among $(0^n|0^n, 0^n)$, $(0^n|0^n, c)$, or $(0^n|0^n, M_c)$. As in our modified PGV proof of Proposition 1, we will give these three queries for free to the adversary, but in line with the Lee–Kwon proof, we will then have to give *the full cycles* of these points for free as well. For concreteness, these cycles are of the form

$$\{(0^n|0^n, x), (0^n|x, 1^n), (x|1^n, 1^n), (1^n|1^n, \overline{x}), (1^n|\overline{x}, 0^n), (\overline{x}|0^n, 0^n)\}$$

where $x \in \{0^n, c, M_c\}$. It is not always the case that the three choices for x lead to distinct cycles, but this is not an issue. Once we have established that these initial free queries do not cause a collision, the Lee–Kwon proof kicks in (where the free cycles only affect the number of queries made so far).

To ease bounding the probability of a collision due to the free cycles, we will give the corresponding queries for free in a particular order, starting with $\tilde{\mathsf{E}}(0^n|0^n, 0^n)$ and $\tilde{\mathsf{E}}(0^n|0^n, 1^n)$. Potentially both these points are affected by our tweaking (if $1^n \in \{c, M_c\}$), but these two queries only lead to a single compression function evaluation, which is insufficient to find a collision. For the remaining four queries in this cycle it is easy to check that the key will be distinct from $0^n|0^n$, so the outcomes will be as for the ideal cipher. When made in order, the third query leads to one additional compression function evaluation; the probability (over the randomness of the answer of the third query) that this is the same as the already known $\mathsf{ADM}(0^n, 0^n, 0^n)$ is 2^{-n}. The probability the fourth query leads to a success is at most $\frac{2}{2^n}$ (as there are now two targets to aim for and the key is fresh), the fifth query at most $\frac{3}{2^n-1}$ (as

the key has been used once before) and the sixth query at most $\frac{2\cdot 4}{2^n}$ (as it adds two compression function evaluations). For the second cycle, first give query $\tilde{\mathsf{E}}(0^n|0^n, c)$ for free (if $c = 1^n$, move straight to the next cycle). This query on its own cannot add a compression function evaluation, thus it cannot lead to a collision. The remaining queries in the cycle all use non-$(0^n|0^n)$ keys so with similar arguments as for the first cycle, the probability of creating a collision is bounded by $\frac{6}{2^n}$, $\frac{7}{2^n}$, $\frac{8}{2^n-2}$, $\frac{9}{2^n}$, and $\frac{20}{2^n}$ respectively. For the third cycle, start with $\tilde{\mathsf{E}}(0^n|0^n, M_c)$ (if $M_c \in \{1^n, c\}$ this query has already been made at a point where it could not have caused a collision and we are already done). Again, as single query in a cycle it cannot lead to a collision; and all the remaining queries in the cycles each have probability at most $\frac{32}{2^n}$ of creating a collision. Taking a union bound leads to a probability of at most $\frac{150}{2^n-2}$ of the free cycles leading to a collision in $\mathsf{ADM}^{\tilde{\mathsf{E}}}$, which is negligible. This concludes the proof that $\mathsf{ADM}^{\tilde{\mathsf{E}}}$ is collision resistant.

For the analysis of $\mathsf{TDM}^{\tilde{\mathsf{E}}}$ we recall the proof by Lee et al. [LSS11]. In fact, they give two proofs: a short, elegant, and tight one and a second, more tedious and less tight one (in the full version only). As we are not interested in tightness at this point, we will use the second proof as our starting point, as it is easier to adapt for our purposes (in particular, it does not modify the adversary). The proof introduces various auxiliary events that bound the number of certain "bad" configurations, and proceeds by showing that (a) the probability of a collision being found when these auxiliary events do not occur is small, and (b) the probability of these auxiliary events occurring is small.

When we move from the ideal cipher to $\tilde{\mathsf{E}}$, we will give the three queries for which we created an interdependency for free to the adversary. This can have any of three effects: (a) it increases the probability of a collision when the "bad" events do not occur; (b) it increases the probability of the "bad" events; or (c) it directly leads to a collision. By inspection, it can be seen that even in the worst case, the three free queries can only lead to a fixed number of additional bad configurations. Thus by changing the bound on the number of bad configurations, this case is taken care of. For (a) the probability increases slightly due to the changed bound on the bad events, but otherwise nothing of note changes. This leaves the investigation of (c). However, it is impossible that just the queries $(0^n|0^n, 0^n)$, $(0^n|0^n, c)$, and $(0^n|0^n, M_c)$ already lead to a collision, as jointly they determine at most one full TDM compression function evaluation. For neither $(0^n|0^n, c)$ nor for $(0^n|0^n, M_c)$ it is possible to occur on the lower half of Fig.3.3(c) (since the corresponding upper-part query would have a key distinct from $0^n|0^n$ as both $c \neq 0$ and $M_c \neq 0$). Moreover, $(0^n|0^n, 0^n)$ can occur on the lower half, but it would only match with $\tilde{\mathsf{E}}^{-1}(0^n|0^n, 0^n)$. This could correspond to either of the free three queries (with very low probability), but never to several. Thus with some modifications (affecting tightness), the full Lee et al. proof goes through also for $\mathsf{TDM}^{\tilde{\mathsf{E}}}$.

□

Theorem 2. *Let $c \in \{0,1\}^n \setminus \{0^n\}$. Then among the compression functions HDM_c, ADM, and TDM neither one directly reduces the idealized cipher in either one of the other two functions for the collision-resistance game.*

Proof. Let us start by separating Abreast-DM from the other two DBL compression functions. For a blockcipher E, define the modified blockcipher $\tilde{\mathsf{E}}$ as follows.

$$M_1 := \mathsf{E}^{-1}(1^n|0^n, \mathsf{E}(0^n|1^n, 0^n) \oplus 1^n), \quad C_0 := \mathsf{E}(0^n|1^n, 0^n),$$

$$C_1 := \mathsf{E}(1^n|0^n, 1^n).$$

$$\tilde{\mathsf{E}}(K_1|K_2, M) := \begin{cases} C_0 \oplus 1^n & \text{if } (K_1|K_2, M) = (1^n|0^n, 1^n); \\ C_1 & \text{if } (K_1|K_2, M) = (1^n|0^n, M_1); \\ \mathsf{E}(K_1|K_2, M) & \text{otherwise.} \end{cases}$$

$$\tilde{\mathsf{E}}^{-1}(K_1|K_2, C) := \begin{cases} 1^n & \text{if } (K_1|K_2, C) = (1^n|0^n, C_0 \oplus 1^n); \\ M_1 & \text{if } (K_1|K_2, C) = (1^n|0^n, C_1); \\ \mathsf{E}^{-1}(K_1|K_2, C) & \text{otherwise.} \end{cases}$$

Note that $\tilde{\mathsf{E}}$ and $\tilde{\mathsf{E}}^{-1}$ as above define a blockcipher. We have

$$\begin{aligned} \mathsf{ADM}^{\tilde{\mathsf{E}}}(0^n, 0^n, 1^n) &= (\tilde{\mathsf{E}}(0^n|1^n, 0^n) \oplus 0^n, \tilde{\mathsf{E}}(1^n|0^n, 1^n) \oplus 0^n) = (C_0, C_0 \oplus 1^n), \\ \mathsf{ADM}^{\tilde{\mathsf{E}}}(1^n, 1^n, 0^n) &= (\tilde{\mathsf{E}}(1^n|0^n, 1^n) \oplus 1^n, \tilde{\mathsf{E}}(0^n|1^n, 0^n) \oplus 1^n) \\ &= (C_0 \oplus 1^n \oplus 1^n, C_0 \oplus 1^n) = (C_0, C_0 \oplus 1^n). \end{aligned}$$

Hence the pair $((0^n, 0^n, 1^n), (1^n, 1^n, 0^n))$ constitutes a collision for ADM with respect to $\tilde{\mathsf{E}}$. Using a case analysis as in Lemma 1, it is possible to prove that for $\tilde{\mathsf{E}}$ a blockcipher as above, with $(\mathsf{E}, \mathsf{E}^{-1}) \leftarrow\!\!\$\ \mathsf{Block}(2n, n)$, the $\mathsf{TDM}^{\tilde{\mathsf{E}}}$ and $\mathsf{HDM}^{\tilde{\mathsf{E}}}_{1^n}$ compression functions are both collision resistant.

We now turn to Tandem-DM. Let E be a blockcipher. For this separation it is easier to derive the separation by tweaking the cipher at two points. This is due to the nested call that the TDM compression function places to E. Set

$$M_0 := \mathsf{E}^{-1}(0^n|0^n, 0^n), \quad M_1 := \mathsf{E}^{-1}(1^n|1^n, 1^n), \quad C_0 := \mathsf{E}(0^n|0^n, 0^n),$$

$$\text{and} \quad C_1 := \mathsf{E}(1^n|1^n, 1^n).$$

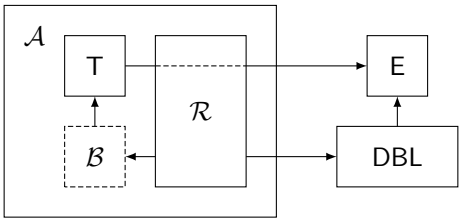


Figure 3.4: The proof of Theorem 3 is essentially a meta reduction ($\mathcal{A}$) simulating an adversary $\mathcal{B}$ for the reduction $\mathcal{R}$. Queries by the meta reduction to E are transformed via T and possibly programmed by the reduction $\mathcal{R}$.

Now define a modified blockcipher $\tilde{\mathsf{E}}$ as follows.

$$\tilde{\mathsf{E}}(K_1|K_2,M) := \begin{cases} 0^n & \text{if } (K_1|K_2,M) = (0^n|0^n,0^n); \\ 1^n & \text{if } (K_1|K_2,M) = (1^n|1^n,1^n); \\ C_0 & \text{if } (K_1|K_2,M) = (0^n|0^n,M_0); \\ C_1 & \text{if } (K_1|K_2,M) = (1^n|1^n,M_1); \\ \mathsf{E}(K_1|K_2,M) & \text{otherwise.} \end{cases}$$

$$\tilde{\mathsf{E}}^{-1}(K_1|K_2,C) := \begin{cases} 0^n & \text{if } (K_1|K_2,C) = (0^n|0^n,0^n); \\ 1^n & \text{if } (K_1|K_2,C) = (1^n|1^n,1^n); \\ M_0 & \text{if } (K_1|K_2,C) = (0^n|0^n,C_0); \\ M_1 & \text{if } (K_1|K_2,C) = (1^n|1^n,C_1); \\ \mathsf{E}^{-1}(K_1|K_2,C) & \text{otherwise.} \end{cases}$$

Note that $\tilde{\mathsf{E}}$ and $\tilde{\mathsf{E}}^{-1}$ as above define a blockcipher. We have

$$\begin{aligned} \mathsf{TDM}^{\tilde{\mathsf{E}}}(0^n,0^n,0^n) &= (\tilde{\mathsf{E}}(0^n|0^n,0^n) \oplus 0^n, \tilde{\mathsf{E}}(0^n|0^n,0^n) \oplus 0^n) \\ &= (0^n \oplus 0^n, 0^n \oplus 0^n) = (0^n,0^n), \\ \mathsf{TDM}^{\tilde{\mathsf{E}}}(1^n,1^n,1^n) &= (\tilde{\mathsf{E}}(1^n|1^n,1^n) \oplus 1^n, \tilde{\mathsf{E}}(1^n|1^n,1^n) \oplus 1^n) \\ &= (1^n \oplus 1^n, 1^n \oplus 1^n) = (0^n,0^n). \end{aligned}$$

Hence the pair $((0^n,0^n,0^n),(1^n,1^n,1^n))$ constitutes a collision for TDM with respect to $\tilde{\mathsf{E}}$. Using a case analysis as in Lemma 1, it is possible to prove that for $\tilde{\mathsf{E}}$ a blockcipher as above, with $(\mathsf{E},\mathsf{E}^{-1}) \leftarrow\!\!\$\ \mathsf{Block}(2n,n)$, the $\mathsf{HDM}^{\tilde{\mathsf{E}}}_{1^n}$ and $\mathsf{ADM}^{\tilde{\mathsf{E}}}$ compression functions are both collision resistant. □

As a corollary of the above results we get that there is no direct reduction from PGV to any of the DBL compression functions: otherwise we also obtain direct reducibility to any other DBL compression function via Theorem 3, which we have shown to be impossible in the above theorem. In the next section we will extend this irreducibility result to free reductions.

Irreducibility of PGV to DBL

We now turn our attention to the converse of Propositions 3 and 5: can one convert any idealized cipher which makes a DBL construction secure to one

which makes a PGV construction secure? We show strong evidence towards the impossibility of such a reduction. To this end, we restrict the class of reductions under the construction to *black-box* ones [RTV04]. Such a reduction is a pair of oracle Turing machines $(\mathsf{T}, \mathcal{R})$. Both machines have access to a blockcipher, T is a transformation which provides an idealized cipher, and $\mathcal{R}$ is a reduction which given oracle access to an algorithm $\mathcal{B}$ breaking the security of a PGV construction when instantiated with T^{E}, breaks the security of a DBL construction with respect to E (for random E). As it will become apparent from the proof of the theorem, the type of reductions that we actually rule out allow both the transformation and the reduction to depend on the blockcipher and hence, in the terminology of [RTV04], the class of reductions that we rule out lies somewhere in between fully-black-box and $\forall\exists$semi-black-box reductions. More concisely, this class is captured as an NBN reduction in the CAP taxonomy of Chapter 5. Jumping ahead, the meaning is that the *C*onstruction may make non-black-box use of primitive, and that the reduction makes black-box use of the *A*dversary resp. non-black-box use of the *P*rimitive.

We make two further simplifications on the structure of the reduction. First we assume that $\mathcal{R}$ queries its break oracle $\mathcal{B}$ once. We call this a *single-query reduction*. Second, we require the reduction to succeed with a constant probability whenever $\mathcal{B}$ is successful. Now, the intuition behind the impossibility of the existence of such a reduction follows that for lower bounds on the output size of hash combiners [Pie08]. The underlying idea is that the collision-resistance security of any of the DBL constructions is *beyond* that of the PGV constructions. More precisely, around $\Theta(2^n)$ queries are needed to break the collision resistance of any of the DBL constructions with noticeable probability, whereas this bound is only $\Theta(2^{n/2})$ for the PGV constructions. To derive a contradiction, we may simulate the break algorithm $\mathcal{B}$ for the reduction with only $\Theta(2^{n/2})$ queries, and the reduction will translate this collision efficiently to a DBL construction collision, which contradicts the $\Theta(2^n)$ collision-resistance bound.

We are now ready to state our irreducibility theorem. Since we are dealing with an impossibility result, for the sake of clarity of the presentation we present the theorem in asymptotic language.

Theorem 3. *There is no single-query fully-black-box ideal-cipher reduction from any of the PGV constructions to any of the DBL constructions for the collision-resistance and [everywhere] preimage-resistance games as long as the reduction is tight: when the number of queries, run times, and success probabilities are parametrized by a security parameter, the reduction is* $(\mathcal{O}(1), \mathcal{O}(1), \mathcal{O}(1), \mathcal{O}(1))$*-tight.*

Let us first recall the precise concrete security bounds for the DBL constructions. We set $N := 2^n$ and $\mathcal{E}$ to be the uniform distribution on $\mathsf{Block}(2n, n)$

throughout. The bounds for Hirose-DM are

$$\mathbf{Adv}^{\mathsf{coll}}_{\mathsf{HDM}_c,\mathcal{E}}(q) \le \frac{2q^2}{(N-2q)^2} + \frac{2q}{N-2q},$$

$$\mathbf{Adv}^{\mathsf{pre}}_{\mathsf{HDM}_c,\mathcal{E}}(q) \le \frac{8q}{N^2} + \frac{8q}{N(N-2)}, \qquad \mathbf{Adv}^{\mathsf{epre}}_{\mathsf{HDM},\mathcal{E}}(q) \le \frac{8q}{N^2} + \frac{8q}{N(N-2)},$$

where the collision-resistance bound holds for $2q < N$ and is from [FGL09b], and the [everywhere] preimage-resistance bounds are from [AFK+11] and are valid for any number of queries.

For Abreast-DM, when $q < N/6$, we have [LK11]

$$\mathbf{Adv}^{\mathsf{coll}}_{\mathsf{ADM},\mathcal{E}}(q) \le \frac{q}{(N-6q)} + \frac{18q^2}{(N-6q)^2},$$

$$\mathbf{Adv}^{\mathsf{pre}}_{\mathsf{ADM},\mathcal{E}}(q) \le \frac{6q}{(N-6q)^2}, \qquad \mathbf{Adv}^{\mathsf{epre}}_{\mathsf{ADM},\mathcal{E}}(q) \le \frac{6q}{(N-6q)^2}.$$

Finally, for Tandem-DM we have for any $1 \le \alpha \le 2q < N$

$$\mathbf{Adv}^{\mathsf{coll}}_{\mathsf{TDM},\mathcal{E}}(q) \le 2N\left(\frac{2eq}{\alpha(N-2q)}\right)^{\alpha} + \frac{4q\alpha}{N-2q} + \frac{4q}{N-2q};$$

for any $1 \le \alpha \le q < N$

$$\mathbf{Adv}^{\mathsf{pre}}_{\mathsf{TDM},\mathcal{E}}(q) \le \frac{16\alpha}{N} + \frac{8q}{N^2(N-2)} + 2\left(\frac{2eq}{\alpha N}\right)^{\alpha} + \frac{4q}{\alpha N} + \frac{q}{N^2(N-q)};$$

for any $1 \le \alpha \le q < N$

$$\mathbf{Adv}^{\mathsf{epre}}_{\mathsf{TDM},\mathcal{E}}(q) \le \frac{16\alpha}{N} + \frac{8q}{N^2(N-2)} + 2\left(\frac{2eq}{\alpha N}\right)^{\alpha} + \frac{4q}{\alpha N} + \frac{q}{N^2(N-q)} + \frac{1}{N};$$

where the collision-resistance bound is from [LSS11], and the [everywhere] preimage-resistance bound is taken from [AFK+11].

Proof. We only need to consider the reducibility of one of the PGV constructions as they reduce to each other via free transformations by Propositions 1 and 2.

Collision resistance. We start by treating the collision-resistance game. Let $\mathsf{DBL}^{\mathsf{E}}$ be a DBL construction with $(\mathsf{E}, \mathsf{E}^{-1})$ sampled from the ideal cipher $\mathcal{E}$. Suppose we have a fully-black-box reduction $(\mathsf{T}, \mathcal{R})$ where $\mathcal{R}$ succeeds in outputting a collision for $\mathsf{DBL}^{\mathsf{E}}$ with constant probability ϵ_{coll} whenever it is provided with a collision for $\mathsf{PGV}_1^{\mathsf{T}^{\mathsf{E}}}$ from $\mathcal{B}$. From $\mathcal{R}$ we construct an algorithm $\mathcal{A}$ which runs the reduction, simulates $\mathcal{B}$, and breaks the collision resistance of $\mathsf{DBL}^{\mathsf{E}}$ with a probability exceeding its best security bound. This leads to a contradiction if the number of queries that $\mathcal{A}$ makes is within the range for

which the bound applies. We show this is indeed the case as long as $\mathcal{R}$ does not place "too many" queries.

We derive $\mathcal{A}$ by letting it simulate a PGV collision-finder $\mathcal{B}$ for the reduction as follows. The reduction has access to blockcipher oracles $\mathsf{E}, \mathsf{E}^{-1}$ against which $\mathcal{A}$ also plays, but the reduction may nonetheless decide to provide T and $\mathcal{B}$ with simulated oracles $\tilde{\mathsf{E}}, \tilde{\mathsf{E}}^{-1}$. Hence, we consider an adversary $\mathcal{A}^{\mathsf{E},\mathsf{E}^{-1}}$ against $\mathsf{DBL}^{\mathsf{E}}$ which runs

$$\mathcal{R}^{\mathcal{B}^{\mathsf{T}^{\tilde{\mathsf{E}},\tilde{\mathsf{E}}^{-1}},\mathsf{T}^{-1^{\tilde{\mathsf{E}},\tilde{\mathsf{E}}^{-1}}}},\mathsf{E},\mathsf{E}^{-1}}$$

and answers $\mathcal{R}$'s queries to E and E^{-1} using its own oracles. The reduction's single query to the (black-box) adversarial interface $\mathcal{B}$ is answered as follows. $\mathcal{A}$ computes $q_{\mathcal{A}}$ values of PGV_1 (for a $q_{\mathcal{A}}$ to be determined later on) with respect to $\mathsf{T}^{\tilde{\mathsf{E}},\tilde{\mathsf{E}}^{-1}}$. To this end, $\mathcal{A}$ needs to run T and answer its blockcipher queries. Note that the reduction $\mathcal{R}$ may be *programming* the blockcipher and $\mathcal{A}$ cannot simply answer T's queries by forwarding them to its own oracles. Algorithm $\mathcal{A}$ handles these queries through $\mathcal{R}$. Figure 3.4 summarizes this setup schematically.

Assuming the reduction places at most q_{E} queries to E or E^{-1} for each blockcipher query of T, and that T places at most q_{T} queries to its oracles for each evaluation, we get that $\mathcal{A}$ makes a total of at most $q_{\mathsf{E}} \cdot q_{\mathsf{T}} \cdot q_{\mathcal{A}}$ queries to E or E^{-1} at this stage. Once the $q_{\mathcal{A}}$ values are computed, if $\mathcal{A}$ finds a collision, it returns it. Else it returns a pair of random distinct points. Algorithm $\mathcal{A}$ resumes $\mathcal{R}$ as before, and terminates by outputting whatever $\mathcal{R}$ outputs. Assuming that $\mathcal{R}$ places at most $q_{\mathcal{R}}$ queries to E or E^{-1} (in addition to those for handling T's queries), we have that algorithm $\mathcal{A}$ makes a total of at most $q_{\mathsf{Tot}} := q_{\mathcal{R}} + q_{\mathsf{E}} \cdot q_{\mathsf{T}} \cdot q_{\mathcal{A}}$ queries to E or E^{-1} during its run.

Using the results of Bellare and Kohno that the lower bound for the generic on the collision resistance of a compression function can only increase if the function is not "balanced" [BK04], we know that for *any* given blockcipher E', the lower bound on the success probability of the attack on the collision resistance of $\mathsf{PGV}_1^{\mathsf{E}'}$ (as given in Section 3.3) applies. Hence *independently of the specification of* T we have

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{coll}}_{\mathsf{DBL},\mathcal{E}}(q_{\mathsf{Tot}}) \geq & \mathbf{Adv}^{\mathsf{coll}}_{\mathsf{DBL},\mathcal{E}}(\mathcal{A}) \\
= & \Pr\left[\mathcal{A} \text{ finds a } \mathsf{DBL}^{\mathsf{E}} \text{ collision}\right] \\
\geq & \Pr\left[\mathcal{R} \text{ finds a } \mathsf{DBL}^{\mathsf{E}} \text{ coll.} \,\middle|\, \mathcal{A} \text{ finds a } \mathsf{PGV}_1^{\mathsf{T}^{\mathsf{E},\mathsf{E}^{-1}}} \text{ coll.}\right] \cdot \\
& \Pr\left[\mathcal{A} \text{ finds a } \mathsf{PGV}_1^{\mathsf{T}^{\mathsf{E},\mathsf{E}^{-1}}} \text{ coll.}\right] \\
\geq & \epsilon_{\mathsf{coll}} \cdot \frac{1}{8e} \frac{q_{\mathcal{A}}^2 + 1}{N}.
\end{aligned}$$

Let us now consider the above inequality for Hirose-DM. In order to simplify the analysis we use the simpler $6q/N$ upper bound for the collision-resistance

advantage when $q \leq N/4$. Setting $\epsilon := \epsilon_{\mathsf{coll}}/(6 \cdot 8e)$, we get

$$\epsilon \cdot \frac{q_{\mathcal{A}}^2 + 1}{N} \leq \frac{q_{\mathcal{R}} + q_{\mathsf{E}} \cdot q_{\mathsf{T}} \cdot q_{\mathcal{A}}}{N},$$

which implies

$$q_{\mathcal{A}} \leq \frac{1}{2\epsilon}\left(q_{\mathsf{E}} q_{\mathsf{T}} + \sqrt{q_{\mathsf{E}}^2 q_{\mathsf{T}}^2 + 4\epsilon q_{\mathcal{R}} - 4\epsilon^2}\right) \leq \frac{1}{\epsilon}\left(q_{\mathsf{E}} q_{\mathsf{T}} + \sqrt{\epsilon q_{\mathcal{R}}}\right).$$

We obtain the desired contradiction if $q_{\mathcal{A}}$ can be chosen so that it is larger than the upper bound given above while ensuring that the total number of queries falls within the range for which the collision-resistance bound holds, i.e., when $q_{\mathsf{Tot}} \leq N/4$. In order to show that these constraints can be met, we need to have that

$$\frac{1}{\epsilon}\left(q_{\mathsf{E}} q_{\mathsf{T}} + \sqrt{\epsilon q_{\mathcal{R}}}\right) + 1 \leq q_{\mathcal{A}} \leq \frac{N/4 - q_{\mathcal{R}}}{q_{\mathsf{E}} q_{\mathsf{T}}}.$$

This is the case if

$$q^4 + q^2\sqrt{2\epsilon q} + \epsilon q^2 + \epsilon q \leq \epsilon N/4,$$

where $q := \max\{q_{\mathcal{R}}, q_{\mathsf{E}}, q_{\mathsf{T}}\}$. Whenever $q \leq c \cdot \sqrt[4]{\epsilon_{\mathsf{coll}} N}$, for a constant $c \approx 6.75$, one can always pick a $q_{\mathcal{A}}$ such that it meets the above constraints. Hence reductions satisfying this inequality for q (e.g., those which are tight) are ruled out.

The collision-resistance irreducibility proofs for Abreast-DM and Tandem-DM are similar to that for Hirose-DM. The main difference is that we arrive at different constraints for $q_{\mathcal{A}}$.

For Abreast-DM, we may simplify the collision-resistance bound to $8q/N$ when $q \leq N/12$. Setting $\epsilon := \epsilon_{\mathsf{coll}}/(8 \cdot 8e)$ we get

$$\epsilon \cdot \frac{q_{\mathcal{A}}^2 + 1}{N} \leq \frac{q_{\mathcal{R}} + q_{\mathsf{E}} \cdot q_{\mathsf{T}} \cdot q_{\mathcal{A}}}{N}.$$

This inequality is identical to that derived for Hirose-DM (except that the constant ϵ has a different value), and the rest of the analysis follows that for Hirose-DM.

For Tandem-DM we set $\alpha = 3$. This ensures that the collision-resistance bound grows more slowly than q^2/N. (Note that this is not the case when $\alpha \leq 2$.) With this choice of α (and noting that $e < 3$) we obtain the simpler bound

$$\mathbf{Adv}^{\mathsf{coll}}_{\mathsf{TDM},\mathcal{E}}(q) \leq \frac{16Nq^3}{(N-2q)^3} + \frac{16q}{N-2q} \leq \frac{128q^3}{N^2} + \frac{32q}{N} \leq \frac{40q}{N},$$

where the penultimate and final inequalities holds for $q \leq N/4$ and $q \leq \sqrt{N}/4$ respectively. Setting $\epsilon := \epsilon_{\mathsf{coll}}/(40 \cdot 8e)$ we get

$$\epsilon \cdot \frac{q_{\mathcal{A}}^2 + 1}{N} \leq \frac{q_{\mathcal{R}} + q_{\mathsf{E}} \cdot q_{\mathsf{T}} \cdot q_{\mathcal{A}}}{N}.$$

Continuing with the analysis as in Hirose-DM we finally arrive at

$$q^4 + q^2\sqrt{2\epsilon q} + \epsilon q^2 + \epsilon q \leq \epsilon\sqrt{N}/4,$$

where $q := \max\{q_{\mathcal{R}}, q_{\mathsf{E}}, q_{\mathsf{T}}\}$ as before. Therefore reductions for which $q \leq c \cdot \sqrt[8]{\epsilon_{\mathsf{coll}}^2 N}$, for some constant c, are ruled out.

[Everywhere] preimage resistance. The intuition behind the proofs for the [everywhere] preimage-resistance games for HDM, ADM, and TDM are as in the collision-resistance games. The proof will utilize theorems analogous to that of Bellare and Kohno [BK04] for the [everywhere] preimage-resistance game. Algorithm $\mathcal{A}$ in the analysis is modified to output a random domain point if it does not find a preimage among its $q_{\mathcal{A}}$ queries. Therefore, the lower bound corresponding to the success probability of $\mathcal{A}$ against PGV_1 for the [everywhere] preimage-resistance game, independently of T, is $\epsilon_{\mathsf{pre}}(q_{\mathcal{A}} + 1)/(2N)$. We now treat each DBL compression function.

For Hirose-DM we use the simplified $32q/N^2$ bound for [everywhere] preimage resistance when $N \geq 3$. (This can be derived from the more precise bound given in Section 3.4.) Setting $\epsilon := \epsilon_{\mathsf{pre}}/(2 \cdot 32)$ we get

$$\epsilon \cdot \frac{q_{\mathcal{A}} + 1}{N} \leq \frac{q_{\mathcal{R}} + q_{\mathsf{E}} \cdot q_{\mathsf{T}} \cdot q_{\mathcal{A}}}{N^2}.$$

It is enough to consider this inequality for $q_{\mathcal{A}} = 0$. In this case we get that $q_{\mathcal{R}} \geq \epsilon N$, and since ϵ_{pre} (and hence ϵ) is a constant, the reduction must be placing a large number of queries, and cannot be tight. The analysis for the everywhere preimage-resistance game is identical.

For Abreast-DM we simplify the [everywhere] preimage-resistance bound to $24q/N^2$ for $q \leq N/12$. Setting $\epsilon := \epsilon_{\mathsf{pre}}/(2 \cdot 24)$ we get

$$\epsilon \cdot \frac{q_{\mathcal{A}} + 1}{N} \leq \frac{q_{\mathcal{R}} + q_{\mathsf{E}} \cdot q_{\mathsf{T}} \cdot q_{\mathcal{A}}}{N^2}.$$

Once again, for $q_{\mathcal{A}} = 0$ we must have that $q_{\mathcal{R}} \geq \epsilon N$, and the reduction cannot be tight. Everywhere preimage resistance is treated identically.

For Tandem-DM we treat the everywhere preimage-resistance game as the advantage bound for this game is higher than that for the preimage-resistance advantage by $1/N$. We set $\alpha = 2 + \sqrt{q}$ so that the advantage bound grows more slowly than q/N. (Note that a constant value for α is not sufficient to

ensure this condition.) We have

$$\begin{aligned}\mathbf{Adv}^{\mathsf{epre}}_{\mathsf{TDM},\mathcal{E}}(q) &\leq \frac{16(2+\sqrt{q})+1}{N} + \frac{4q}{N(2+\sqrt{q})} \\ &\quad + 2\left(\frac{6^2q^2}{N^2(2+\sqrt{q})^2}\right)\left(\frac{6q}{N(2+\sqrt{q})}\right)^{\sqrt{q}} \\ &\quad + \frac{q}{N^2(N-q)} + \frac{8q}{N^2(N-2)} \\ &\leq \frac{20\sqrt{q}}{N} + \frac{72q}{N^2} + \frac{2q}{N^3} + \frac{24q}{N^3} + \frac{33}{N} \quad \text{for} \quad q \leq N/2 \text{ and } N \geq 3 \\ &\leq \frac{151\sqrt{q}}{N} \quad \text{for} \quad q \geq 3.\end{aligned}$$

Setting $\epsilon := \epsilon_{\mathsf{pre}}/(2 \cdot 151)$ we finally arrive at

$$\epsilon \cdot \frac{q_{\mathcal{A}}+1}{N} \leq \frac{\sqrt{q_{\mathcal{R}} + q_{\mathsf{E}} \cdot q_{\mathsf{T}} \cdot q_{\mathcal{A}}}}{N}, \quad \text{which implies} \quad q_{\mathcal{A}} \leq \frac{1}{\epsilon^2}(q_{\mathsf{E}}q_{\mathsf{T}} + \epsilon\sqrt{q_{\mathcal{R}}}).$$

The rest of the analysis is similar to Hirose-DM: applying the bound on the total number of queries for which the above inequality holds we obtain

$$q^4 + \epsilon q^2\sqrt{q} + \epsilon^2 q^2 + \epsilon^2 q \leq \epsilon^2 N/2,$$

where $q := \max\{q_{\mathcal{R}}, q_{\mathsf{E}}, q_{\mathsf{T}}\}$ as before. As a result reductions for which $q \leq c \cdot \sqrt[4]{\epsilon^2_{\mathsf{pre}} N}$, for some constant c, are ruled out.

□

Finally, we identify two open questions in the vicinity of Theorem 3. First, it is conceivable that the techniques of [Pie08] can be leveraged to derive a more general theorem which rules out reductions that call the break oracle multiple times. Furthermore, one might also be able to extended the result to arbitrary games for two given constructions, as long as a lower bound on the success probability of an attack on the security of the first construction is noticeably higher than an upper bound on the security of the second.

3.5 Conclusions and Open Problems

We summarize our reducibility results in Figure 3.5 and refer to the caption for details. One important observation from these results is that we do not have one single "Y" column, i.e., a compression function which reduces to all of the other ones—or, equivalently, a compression function which is secure if any of the others is secure. This would be a clear winner in the sense that it is the safest choice for practice.

For the "n" entries of Table 3.5(b) we can show that there is a separation for a large class of potential transformation functions. More specifically, we show

	$\mathcal{G}_1$	$\mathcal{G}_2$	TDM	HDM_c	ADM
$\mathcal{G}_1$	Y 1	N 1	Y 3	Y 3	Y 3
$\mathcal{G}_2$	N 1	Y 1	–	–	–
TDM	N 2	–	Y	N 2	N 2
HDM_c	N 2	–	N 2	Y	N 2
ADM	N 2	–	N 2	N 2	Y

(a) Results for the identity transformation.

	$\mathcal{G}_1$	$\mathcal{G}_2$	TDM	HDM_c	ADM
$\mathcal{G}_1$	Y $\rightarrow$	Y 2	Y $\rightarrow$	Y $\rightarrow$	Y $\rightarrow$
$\mathcal{G}_2$	Y 2	Y $\rightarrow$	Y $\star$	Y $\star$	Y $\star$
TDM	N 3	N 3	Y	n 2	n 2
HDM_c	N 3	N 3	n 2	Y	n 2
ADM	N 3	N 3	n 2	n 2	Y

(b) Results for arbitrary transformations.

Figure 3.5: Summary of our reducibility results for collision resistance. A "Y" or "N" in a cell means that any cipher which makes the compression function corresponding to the row collision-resistant also makes the compression function corresponding to the column collision-resistant. A "–" in direct reductions indicates a syntax mismatch. The number below an entry indicates the theorem/proposition supporting the claim. An arrow "$\rightarrow$" means that the result is implied by the left table. Reductions on the diagonal of TDM, HDM_c, and ADM trivially follow by self-reductions. Note that for arbitrary transformations each cell might be using different transformations. The star symbol "$\star$" denotes reducibility by transitivity. An "n" is a separation for a restricted class of transformations; see Section 3.5.

that there is no surjective transformation T to reduce, say, ADM to HDM_{1^n}, as long as the transformation also preserves HDM-security "backwards." Here, surjectivity means that T^{E} varies over all possible blockciphers if E runs through all blockciphers, and backward security preservation means that $\mathcal{E}$ is secure for HDM if $\mathsf{T}^{\mathcal{E}}$ is. Transformations which are covered by this include, for example, those of the form $\mathsf{T}^{\mathsf{E}}_{\pi_1,\pi_2}(K_1|K_2,M) = \pi_2(\mathsf{E}(K_1|K_2,\pi_1(M)))$ for fixed involutions π_1,π_2 over $\{0,1\}^n$, or more generally, any transformation which is its own inverse (over $\mathsf{Block}(2n,n)$). An example of a surjective transformation which is not backward secure for PGV_1 is $\mathsf{T}^{\mathsf{E}}(K,M) = \mathsf{E}(K,M) \oplus K$, because it maps PGV_1 for T^{E} to PGV_2 for E, and we know that there are idealized ciphers making PGV_2 secure but PGV_1 insecure.

The argument is as follows. Assume that there exists such a T. Then for any blockcipher E which makes HDM secure, the blockcipher T^{E} makes ADM secure. However, we also know that there is a blockcipher $\mathsf{E}^\star$ such that $\mathsf{E}^\star$ gives rise to a collision-resistant $\mathsf{HDM}^{\mathsf{E}^\star}_{1^n}$ but renders $\mathsf{ADM}^{\mathsf{E}^\star}$ collision tractable (see Theorem 2). Now define E to be any blockcipher in the preimage of $\mathsf{E}^\star$ under T (such an E exists as T is surjective). The transformation now maps E to $\mathsf{E}^\star$, which means that it fails to provide security for ADM. Furthermore, E makes $\mathsf{HDM}^{\mathsf{E}}_{1^n}$ collision resistant by assumption about backward security.

This, however, contradicts the requirement of reducibility from ADM to HDM, because E makes HDM secure but T^{E} is insecure for ADM.

Open problems. Recall that we showed that one can transform a good blockcipher E (or rather distribution $\mathcal{E}$) for the PGV_1-group into a good one T^{E} for the PGV_2-group. We also presented a transformation in the opposite direction. Ideally, though, one would be interested in a *single* transformation T which, given $\mathcal{E}$ making a PGV construction secure, turns it into $\mathsf{T}^{\mathcal{E}}$ which *simultaneously* makes both the PGV_1-group and the PGV_2-group secure. Such a transformation would be of interest because incorporating it into the compression function would result in a construction that relies on a weaker assumption than either just PGV_1 or PGV_2. Consequently, it would provide a handle to *strengthen* existing schemes (in a provable way). Note that such a result would not contradict the separation of direct reducibility between the PGV_1-group and the PGV_2-group, because simultaneous security looks for a (transformed) cipher in the intersection of good (distributions over) blockciphers for both groups. This intersection is clearly non-empty because it contains the ideal cipher; the question to address here is how hard it is to hit a distribution when starting with the minimal security assumption that (a potentially non-ideal) $\mathcal{E}$ is good for at least one PGV construction. We remark our technique of separating the DBL constructions from PGV_1 does not seem to apply here, as the simultaneous security bound for PGV_1 and PGV_2 is $\Theta(q^2/2^n)$. However, surjective, backward-secure transformations are still ruled out according to the same argument as in the HDM vs. ADM case.

Another direction of research left open here is the existence of reductions among two compression functions for *different* games. For example, one might ask whether the collision resistance of one construction for a blockcipher gives preimage resistance in another (or perhaps the same) construction with the same cipher. In particular, using Simon's result [Sim98] one might be able to demonstrate the impossibility of reducing collision resistance to preimage resistance for any of the PGV constructions.

Finally, let us emphasize that all results in this work apply directly to compression functions. Needless to say, in practice compression functions are iterated in order to hash arbitrary lengths of data. This could extend the set of $\mathcal{E}$ that provide security, potentially changing the scope for transformations between constructions. We leave the question of the existence of reductions among iterated hash functions as an interesting open problem.

Chapter 4

Random-Oracle Reducibility

In this chapter, we apply our idealized-model-reduction approach to the case of the random-oracle model. We demonstrate how to use this technique to argue that one scheme is strictly better than another scheme even though both schemes rely on the random oracle to an unknown extent. Unlike in the previous chapter, we focus here more carefully on the security assumptions involved in either scheme.

After an informal general discussion in Section 4.1, we define several flavors of random-oracle reducibility in Section 4.2. We relate these reducibility notions in Section 4.3 and establish a hierarchy amongst them. Our primary application example appears in Section 4.4, where we reduce the twin ElGamal encryption scheme to the regular ElGamal encryption scheme, showing that the former scheme is indeed superior. In Section 4.5 we present further reducibility results concerning various signature schemes in the random-oracle model.

This work appeared at CRYPTO 2011 [BF11].

4.1 Introduction

Suppose you have a cryptographic scheme A which can be shown to be secure in the random-oracle model [BR93] under some assumption $\mathbb{A}$, say, the RSA assumption. Assume furthermore that someone presents to you a scheme B for the same purpose which is also secure in the random-oracle model, but now under the potentially weaker assumption $\mathbb{B}$ like factoring. Clearly, if it was not for the random oracle, and scheme B would also improve over A in other relevant aspects like efficiency, then scheme B should be preferred. Unfortunately, the random-oracle model introduces some undesirable uncertainty when simply following the strategy of picking the scheme with the weaker assumption.

Formally, proofs in the random-oracle model (ROM) all rely on equally-powerful random hash functions, but very often the *exact* requirements for the hash functions to conduct a security proof for a scheme remain unclear. This is all the more true since the random oracle in some schemes is uninstantiable in

the sense that no efficient hash function can securely replace the random oracle [CGH98]. For our example of schemes A and B above this means that scheme B may rely on a weaker assumption $\mathbb{B}$, but the actual requirements on the hash function may be much stronger than the ones for A. In the extreme, the hash function in scheme B may be uninstantiable, whereas the hash function for A may rely on a very mild cryptographic assumption like collision resistance (albeit no proof has been found for this so far).

A natural approach to overcome the problem would be to determine the exact requirements on the hash function and to show that scheme B also relies on weaker assumptions for the hash function than scheme A. However, pinning down these properties of random oracles is often tedious and does not yield the desired result, especially since one would also need to show that the properties are necessary. One example is the hash function properties for OAEP, where Boldyreva and Fischlin [BF05, BF06] and later Kiltz et al. [KOS10] gave necessary and, for much weaker security notions than IND-CCA, sufficient conditions on the hash function (in combination with further assumption about the underlying trapdoor permutation). None of these results, however, shows the desired kind of strong security. To complement these results, Kiltz and Pietrzak [KP09] claimed that for arbitrary trapdoor permutations the hash function in OAEP cannot be instantiated securely to derive IND-CCA security. The latter result is not known to be applicable to specific trapdoor permutations like RSA, though.

Yet another approach by Bellare, Hoang, and Keelveedhi [BHK13] suggests an additional abstraction layer called universal computational extractors (UCEs). Their idea is essentially to introduce standard-model assumptions that suffice to construct various schemes. In principal, one could use this technique to compare two schemes: whenever a single UCE-type assumption makes both schemes simultaneously secure, any hash function satisfying this assumption will also provide security for both schemes. However, it may still be that one of the two schemes actually only requires a milder assumption. Conversely, a UCE assumption that specifically targets one construction, the approach taken in newer versions of [BHK13], is very close to determining the exact requirements on the hash function—which one could have done in the first place. Arguably the most interesting case occurs when two UCE-type assumptions for two corresponding schemes form a strict subset relation *and* the assumptions are respectively both sufficient and necessary. There, one can make the desired assertion that one scheme requires less power than the other one with respect to the random oracle. It is also noteworthy that there is currently only one actual standard-model UCE *instantiation* (a variant of correlated-input hashing; [BM14]) of the applications put forward in [BHK13], albeit under very strong assumptions.

Random-oracle reducibility. The strategy we suggest here is based on the classical reductionist approach to relate cryptographic assumptions: show

that any hash function H, ranging from efficient instantiations to random oracles, which makes scheme A secure under assumptions $\mathbb{A}$ also makes scheme B secure under assumptions $\mathbb{B}$. We saw the usefulness of this approach earlier in Chapter 3 when we applied it to the ideal-cipher model. Since we compared very similar constructions there, were able to show direct reducibility in some cases. Technically, this seems to be too optimistic for hash functions in different schemes, because they often cannot be used unchanged but rely on different domains, ranges, etc. We thus allow again for a "structural" transformation T^{H} of H for scheme B, possibly depending on the specific hash function. There are three possibilities to relate the hash functions in the schemes:

Definition 5 (Random-oracle reducibility, informally)**.** *Let $\mathbb{A}$ and $\mathbb{B}$ be some sets of assumptions. A random oracle in scheme B strictly resp. strongly resp. weakly reduces to the random oracle in scheme A if for every hash function H there exists a transformation T such that*

(strictly)

$$A^{\mathsf{H}} \text{ secure under } \mathbb{A} \implies \text{scheme } B^{\mathsf{T}^{\mathsf{H}}} \text{ secure under } \mathbb{B},$$

(strongly)

$$A^{\mathsf{H}} \text{ secure under } \mathbb{A} \implies \begin{cases} \text{scheme } B^{\mathsf{T}^{\mathsf{H}}} \text{ secure under } \mathbb{A} \cup \mathbb{B} \\ \text{and } B^{\mathsf{T}^{\mathsf{H}'}} \text{ secure under } \mathbb{B} \text{ for some } \mathsf{H}', \end{cases}$$

(weakly)

$$A^{\mathsf{H}} \text{ secure under } \mathbb{A} \implies \text{scheme } B^{\mathsf{T}^{\mathsf{H}}} \text{ secure under } \mathbb{A} \cup \mathbb{B}.$$

Several details are hidden in this informal definition, of course, e.g., what a "secure" scheme is, which properties the transformation must satisfy, or how cryptographic assumptions of hash function instantiations are dealt with. We fill in these details in the formal definition and keep it informal for now. We note, however, that the formal definition covers *any* type of hash function, i.e., both oracle-based ones, as well as keyed hash functions with succinct descriptions, or mixtures thereof. In particular, the security of scheme B in the strong case may only be known for a random oracle H'.

In contrast to ideal-cipher reducibility in Chapter 3, we will potentially have to deal with different assumptions $\mathbb{A}$ and $\mathbb{B}$ on either cryptographic scheme. This was not needed for the ideal-cipher constructions since they are qualitatively "simpler" and comparatively lower-level constructions. As such, there are typically no further assumptions on these schemes, i.e., they correspond to the special case where $\mathbb{A} = \mathbb{B} = \emptyset$. One could, of course, consider Definition 5 for the ideal-cipher model as well, when dealing with cryptographic schemes in the ideal-cipher model that require further assumptions.

For identical assumptions $\mathbb{A} = \mathbb{B}$, or even if $\mathbb{A} \subseteq \mathbb{B}$, all three notions of Definition 5 coincide. The difference can be best explained for the case $\mathbb{B} \subsetneq \mathbb{A}$, i.e., that the assumptions $\mathbb{A}$ are strictly stronger than $\mathbb{B}$. The strict notion

can in this case be put informally as saying "scheme B is strictly superior to scheme A in regard of the assumptions, even for the hash function." The strong and presumably more accessible approach can be described as "scheme B is at least as good as scheme A in regard of the assumptions, but potentially superior." The weak case says that "scheme B is at least as good as scheme A." In terms of security assumptions it seems that the strict and strong versions are the interesting ones (hence the names); the weak version does not provide any potential improvement concerning the assumption. We note that in the strong case often a security proof for scheme B in the random-oracle model can be given without assuming security of A. We merely introduced the dependency via the prerequisite of the implication to make the notions comparable.

As a first sanity check, note how previous uninstantiability results relate to either kind of definition. If the hash-function security of B can be (weakly, strongly, or strictly) reduced to the hash-function security of A, and B turns out to be uninstantiable, then this also follows for scheme A (else T^{H} would be a valid instantiation for B). In this regard the reduction approach also allows to extend uninstantiability results without directly showing the ineffectiveness of efficient hash functions. Vice versa, any new result about secure instantiations of A would immediately transfer to B. Also, uninstantiability immediately implies that there are schemes A (allowing efficient instantiations) and B (being uninstantiable) such that the random oracle for B does not (even weakly) reduce to the one for scheme A. Indeed, we saw such *ir*reducibility results in Chapter 3 before, although they did not necessarily imply uninstantiability.

Example: hashed ElGamal encryption. To show that the strong approach is applicable and the definition non-trivial we discuss the case of hashed ElGamal encryption [ABR01] and its chosen-ciphertext security proof under the strong Diffie–Hellman (DH) assumption in the random-oracle model [CS03]. Here, the strong DH assumption says that computing DH keys is infeasible even if given (restricted) access to a decisional DH oracle. Cash, Kiltz, and Shoup [CKS09] present a variant which can be shown to be CCA secure under the (regular) DH assumption in the random-oracle model. This is a clear example of two schemes where the variant seems to improve over the original one in terms of assumption, but where this conclusion is technically not known to be sound because of the random-oracle model.

The original hashed ElGamal encryption scheme encrypts a message under public key $X = g^x$ as (Y, c), where $Y = g^y$ and $c = \mathsf{Enc}(k, m)$ for the hashed Diffie–Hellman key $k = \mathsf{H}(Y, X^y)$ and the symmetric encryption scheme Enc. The variant in [CKS09] instead computes two related ephemeral Diffie–Hellman keys from public keys $X_0 = g^{x_0}$ and $X_1 = g^{x_1}$, and derives a ciphertext (Y, c) for $Y = g^y$ and $c = \mathsf{Enc}(k, m)$ for $k = \mathsf{H}(Y, X_0^y, X_1^y)$. We show (for a slight derivate of the scheme in [CKS09]) that the random oracles can be strongly reduced to the one of hashed ElGamal. Ciphertexts in our variant are defined

as

$$(Y, c, k_1), \quad \text{where } Y = g^y,\ c = \mathsf{Enc}(k_0, m) \text{ for } k_0 = \mathsf{H}(Y, X_0^y),\ k_1 = \mathsf{H}(Y, X_1^y),$$

i.e., we split the hashing into two evaluations, one for each public-key part, and use the second key as a kind of confirmation that the first key is computed correctly. We can view this as the transformation

$$\mathsf{T}^{\mathsf{H}}(Y, Z_0, Z_1) = \mathsf{H}(Y, Z_0) | \mathsf{H}(Y, Z_1)$$

and where we use a special symmetric encryption scheme where the key part k_1 is output in clear.

We then prove that IND-CCA security for hashed ElGamal implies security of (our variant) of the twin DH scheme *for any hash function* H for the same assumptions that the hashed ElGamal is secure for. We also show that our variant is secure in the random-oracle model assuming only the assumptions given in [CKS09]. It follows that the random oracle in our scheme is strongly reducible to the one of hashed ElGamal.

Note that yet another hashed ElGamal scheme, related to the original scheme, has been shown to be uninstantiable [BBP04]. The scheme differs in two important aspects from our scheme, though. First, their hashed ElGamal encryption does not use randomness and is thus deterministic. Second, the security notion considered in [BBP04] is *IND-CCA preservation* which gives the adversary simultaneously access to the algorithms of the public-key scheme and the symmetric scheme involving secret keys. In contrast, we use the standard notion of IND-CCA security for the hybrid (public-key) scheme.

We note that the security reduction for our variant to the underlying primitives like the Diffie–Hellman problem for random oracle H' is looser than the one in [CKS09] in terms of concrete bounds, but since both proofs are in the random-oracle model, concrete bounds must be taken with a grain of salt anyway. At the same time, our scheme relates the random oracle to the one in the original scheme. Of course, concreteness of security bounds is another important aspect, besides efficiency, when considering random-oracle reducibility. In principle, it could be incorporated as an explicit requirement in the notion, as we did in Chapter 3. We relinquish to do so here, because both aspects, tightness and efficiency, depend to some extend on the individual willingness to pay for the additional security guarantees through the random-oracle reducibility.

Reductions for signature schemes. We give further examples of the applicability of random-oracle reducibility by considering common signature schemes like Guillou–Quisquater [GQ88] or PSS [BR96] and showing that the random oracle of a probabilistic version of FDH [Cor00] (full-domain hash) reduces to the random oracles in these schemes. However, note that FDH signatures are only known to be uninstantiable in special cases according to [DOP05, DHT11]

for plain hash evaluations over the message (i.e., no randomness). The first result only applies to random trapdoor permutations (i.e., the result is not known to apply to RSA), and the second more recent result holds when RSA is treated as a black-box group. Perhaps surprisingly, a recent result [HSW14] based on indistinguishability obfuscation actually shows instantiability of deterministic FDH signatures, where the hash function undergoes a non-black-box transformation. Any progress in terms of (un)instantiability of FDH signatures to our probabilistic case would immediately allow to conclude that the Guillou–Quisquater signature scheme and the PSS scheme are (un)instantiable. This would somehow extend the uninstantiability result of Goldwasser and Kalai [GK03] about general (and somewhat contrived) Fiat–Shamir schemes to the "more natural" species.

We discuss another random-oracle reduction of (probabilistic) BLS signatures [BLS04] to the Schnorr signature scheme [Sch91]. In this case, however, we need a non-standard assumption to make the reduction work. Namely, we need the knowledge-of-exponent assumption KEA1 [HT98, BP04] which roughly says that, when complementing a value X to a Diffie–Hellman tuple $(X, Y, \mathrm{DH}(X, Y))$, one must know the discrete logarithm y of Y. Additionally, we require that this assumption hold even if one can get additional Schnorr signatures under the key X. For the random oracles this means that, if our version of the BLS scheme is uninstantiable, then so is the Schnorr signature scheme, or the augmented KEA1 assumption is false.

Some words of caution. Just as reductions between number-theoretic assumptions merely relate problems like factoring and RSA, but do not touch the question if RSA is really hard, a reduction for random oracles does not mean that scheme B, in and of itself, is secure (under assumptions $\mathbb{B}$) or that the hash function can be securely instantiated. The reduction only says that scheme B can be made as secure as scheme A in regard of the hash function. Since we do not put any formal prerequisite about the security of scheme A, which may thus be insecure, the reduction could potentially be trivial.

However, as for relating number-theoretic assumptions, where the stronger assumption is usually accompanied by some hardness analysis, scheme A typically comes with some form of security guarantee. Often, this is at least a security proof in the random-oracle model, or sometimes for a relaxation thereof like non-programmable random oracles [Nie02, FLR+10], traceable random oracles [NYWO09], or leaky random oracles [YMO08]. The advantage of our approach is that it follows immediately that B can also be shown secure under the corresponding assumption about the hash function.

One caveat is that the transformed hash function T^{H}, unlike random oracles, obeys some structure, as the "split" evaluation in our ElGamal example. Hence, when instantiated with some efficient hash function h, scheme B could become insecure for the transformed hash function T^{h}, despite the reduction and a proof that T^{H} makes B secure for random oracle H. Noteworthy, at the same

time B could be secure when instantiated with h directly, instead of going through the transformation $\mathsf{T}!$ We observe, however, that this is an inherent limitation of the random-oracle model: it solely provides a heuristic which does not allow to conclude security under concrete instantiations. Our approach at least gives some confidence in the choice of the hash function in the sense that the security is at least as good as the one of another, hopefully well-examined scheme.

4.2 Defining Random-Oracle Reducibility

Hash functions. We consider families $\mathcal{H}$ of hash functions H where it is understood that H is (not necessarily efficiently) samplable from $\mathcal{H}$ according to a security parameter λ. It is thus also clear that a hash function H may have a restricted input or output length, depending on λ. We write $\mathsf{H} \leftarrow\!\!\$\ \mathcal{H}(1^\lambda)$ for the sampling. For example, to model a random oracle we let $\mathcal{H}(1^\lambda)$ be the family of all functions with the specified domain and range and the sampling picks a random function from this set. In the sequel we usually simply identify the hash function $\mathsf{H}(\cdot)$ with its description H itself. We assume that hash functions are deterministic in the sense that, once a hash function has been sampled, its behavior is fixed. A hash-function family may rely on some cryptographic assumptions $\mathbb{H}$; in case of random oracles no assumption $\mathbb{H}$ is necessary as the sampling of a random function already provides all desirable security properties.

Given a hash function H for a scheme A, we assume that each party or algorithm gets oracle access to H. Furthermore, the hash function H may include a public description part which is then also given to all parties and algorithms as additional input. This public part may be for example the full description of H, or only parts thereof, e.g., if H is a hybrid between a random oracle and a keyed hash function. A hash-function family is *efficient* if it follows the usual notion of an efficient keyed hash function, i.e., the sampling is efficient, a sampled function H is efficiently computable and entirely described through a public part.

Transformations. A hash function H used in a cryptographic scheme A may not be immediately applicable to another scheme B for the mere fact that the domain and range do not fit. We therefore "slot in" a transformation algorithm T, such that scheme B then uses the hash function T^{H} (with the semantic that any algorithm or party gets public descriptions of T^{H} as additional input). We write $\mathsf{T}^{\mathcal{H}}$ for the corresponding hash-function family (described by sampling $\mathsf{H} \leftarrow\!\!\$\ \mathcal{H}(1^\lambda)$ and evaluating T^{H}). Ideally, the transformation should only make structural modifications (like adapting the domain and range) and should be deterministic.

We required that transformations for ideal-cipher reducibility in Chapter 3 be stateless and we will require the same for random-oracle reducibility. In

addition to preventing lazy sampling (which is applicable to random oracles as well), this also prevents the following trivialization. Suppose scheme B is instantiable for some hash-function family $\mathcal{H}$. Construct the transformation T which ignores its oracle and instead initially samples a function $\mathsf{H}' \leftarrow\!\!\$\ \mathcal{H}$ and answers subsequent queries according to H'. Again, scheme B remains secure and reduces to any scheme A. Here, we used statefulness to remember the choice of the hash function in order to provide consistent answers.

One can nonetheless allow for rather general transformations, possibly even considering transformations which themselves rely on assumptions $\mathbb{T}$.

Security of schemes. We consider security of schemes to be defined via a general notion of games, like we did in Chapter 3. In this chapter, we use a slightly simplified asymptotic version of the previous definition. This will make the presentation more compact and allows us to focus on the new aspect of this chapter, namely the computational assumptions of the schemes we consider.

As we can subsume several games like the ones for blindness and unforgeability for blind signature schemes into a single game, with corresponding sub games for which the adversary initially decides to mount the attack against, we consider a single game G only, as before. In contrast to the previous definition, we will, however, deal only with one adversary and make an asymptotic statement. We let $\mathbf{Adv}^{\mathsf{G}}_{\mathcal{H}}(\mathcal{A})$ denote the *advantage* of adversary $\mathcal{A}$ playing game G where the game samples a hash function H from $\mathcal{H}$, i.e., the adversary's success probability of winning the game. This makes $\mathbf{Adv}^{\mathsf{G}}_{\mathcal{H}}$ an implicit part of G. Here, in decisional games the advantage usually denotes the adversary's success probability minus the trivial guessing probability of $1/2$, and in computational games the advantage is usually the adversary's probability of computing a solution.

We envision security assumptions $\mathbb{A}$ for a scheme A as a set of elementary properties such as unforgeability of an underlying MAC or number-theoretic assumptions like the hardness of factoring. We can then apply common set operations and relations to assumptions in a well-defined way, e.g., $\mathbb{A} \cup \mathbb{B}$ comprises all assumptions stated in $\mathbb{A}$ and $\mathbb{B}$, and $\mathbb{B} \subseteq \mathbb{A}$ means that assumptions in $\mathbb{B}$ hold if $\mathbb{A}$ holds. This approach is too applicable for the hash function assumptions $\mathbb{H}$ and possibly the transformation assumptions $\mathbb{T}$. We also assume that assumptions are "opt-in", i.e., need to be specified in the set, or else the assumption does not hold. Formally we can define this by considering a universe $\mathbb{U}$ of assumptions and say that any assumption in $\mathbb{U} \setminus \mathbb{A}$ is false.

Note that we keep the formal specifications of games and assumptions at a minimal level. This is possible as we later demand random-oracle reducibility with respect to specific games and assumptions. It is thus up to the reduction statement to consider "reasonable" games and assumptions. We only need very limited syntactical requirements here and can, for example, even allow conflicting assumptions in $\mathbb{A} \cup \mathbb{B}$ (in which case, however, the claims usually

become trivial).

Definition 6 (Game-based security)**.** *Let A denote a cryptographic scheme using a hash family $\mathcal{H}$ and G an associated security game. Scheme A is called* $(\mathsf{G}, \mathcal{H})$-secure under assumptions $\mathbb{A}$ *for hash family $\mathcal{H}$ relying on assumptions $\mathbb{H}$ if for any efficient adversary $\mathcal{A}$ we have that $\mathbf{Adv}^{\mathsf{G}}_{A,\mathcal{H}}(\mathcal{A})$ is negligible in the security parameter, where the probability is over all random choices of the game (including the choice of the hash function), the algorithms, and the adversary.*

As an example, consider the IND-CCA security game for an encryption scheme A (in the random-oracle model), in which the game G proceeds in stages where $\mathcal{A}$ in the first phase receives a public key (in case of an asymmetric scheme) and gets access to a decryption oracle plus the random oracle, then outputs a pair of equal-length messages m_0, m_1 to receive a *single* challenge ciphertext of m_b for secret random bit b, and finally continues asking decryption queries except for the challenge ciphertext. The adversary wins if correctly predicts b, and the advantage of the adversary is the probability for a correct prediction minus $1/2$. In the notation above an IND-CCA-secure encryption scheme relying on some cryptographic assumption $\mathbb{A}$ is $(\mathsf{G}, \mathcal{H})$-secure under $\mathbb{A}$ for random oracle $\mathcal{H}$.

Random-oracle reducibility. As explained in the introduction, we introduce weak, strong, and strict notions of random-oracle reducibility:

Definition 7 (Random-oracle reducibility)**.** *Let A be a cryptographic scheme with security game A and assumptions $\mathbb{A}$, and let B be a cryptographic scheme with game B and assumptions $\mathbb{B}$. Then the random oracle in scheme B (strictly resp. strongly resp. weakly)* reduces *to the random oracle in scheme A if for every hash-function family $\mathcal{H}$ relying on assumptions $\mathbb{H}$ there exists a stateless transformation T such that if scheme A is $(\mathsf{A}, \mathcal{H})$-secure under $\mathbb{A}$, ...*

(strict)
... then scheme B is $(\mathsf{B}, \mathsf{T}^{\mathcal{H}})$-secure under $\mathbb{B}$.

(strong)
... then scheme B is $(\mathsf{B}, \mathsf{T}^{\mathcal{H}})$-secure under $\mathbb{A} \cup \mathbb{B}$ and
scheme B is $(\mathsf{B}, \mathsf{T}^{\mathcal{H}'})$-secure under $\mathbb{B}$ for some $\mathcal{H}'$ relying on $\mathbb{H}'$.

(weak)
... then scheme B is $(\mathsf{B}, \mathsf{T}^{\mathcal{H}})$-secure under $\mathbb{A} \cup \mathbb{B}$.

We say that $(B, \mathsf{B}, \mathcal{H}, \mathbb{B})$ is (weakly or strongly or strictly) random-oracle reducible *to $(A, \mathsf{A}, \mathcal{H}, \mathbb{A})$. It is* polynomial-time (weakly or strongly or strictly) random-oracle reducible *if it is random-oracle reducible via (deterministic) stateless polynomial-time transformations T for any hash-function family $\mathcal{H}$.*

We occasionally simply say that B is random-oracle reducible (RO reducible) to A if the games and assumptions are clear from the context.

Some remarks about the definition and variations follow:

Figure 4.1: Relations between reducibility notions.

- The above does not rule out trivial examples where scheme B actually relies on stronger assumptions $\mathbb{B}$ than scheme A, e.g., if $\mathbb{A}$ is a subset of $\mathbb{B}$. As explained in the introduction, the most interesting examples seem to be the ones where assumptions $\mathbb{B}$ are weaker than $\mathbb{A}$ or at least incomparable. Occasionally, however, one may be interested in a scheme B which requires stronger assumptions $\mathbb{B}$ but which is more efficient (or has other desirable properties).
- For strong reducibility, the second condition of the implication includes a second hash function family $\mathcal{H}'$ that may rely on assumptions $\mathbb{H}'$. This can be thought of as a hint that scheme B is potentially better with respect to assumptions than A. In particular, if one is willing to "trade" the scheme's assumptions $\mathbb{B} \setminus \mathbb{A}$ for assumptions on the hash function $\mathbb{H}'$.
- We can devise stronger notions concerning the order of quantification for our reducibility notion. Above, the transformation can depend on the specific hash-function family $\mathcal{H}$, and thus possibly specific properties of $\mathcal{H}$. One could alternatively demand that the transformation needs to be universal in the sense that it works for any $\mathcal{H}$.
- The above definition assumes that transformation T does not rely on additional assumptions. More generally, one could specify assumptions $\mathbb{T}$ and say that scheme B is secure under assumptions $\mathbb{B}' = \mathbb{B} \cup \mathbb{T}$.
- According to our syntax, the adversary $\mathcal{B}$ in game G_B with the transformed random oracle would get access to T^{H}, but not H itself. This can be easily patched by letting the transformation T give direct access to H through a special query mode.

4.3 Basic Results

Relating the reducibility notions. We first show that strict reducibility implies strong reducibility which implies weak reducibility. Figure 4.1 depicts all relationships between the notions.

Proposition 6 (Strict ⇒ strong ⇒ weak reducibility)**.** *Let A be a cryptographic scheme with security game A and assumptions $\mathbb{A}$, and let B be a cryptographic schemes with game B and assumptions $\mathbb{B}$. If the random oracle in scheme B*

strictly reduces to the random oracle in scheme A, then it also strongly reduces to it. If it strongly reduces to it, then it also weakly reduces to it.

Proof. Consider first the implication from strict to strong. If A cannot be secure for *any* hash-function family then the claim is trivially true. Hence, assume that A is secure under $\mathbb{A}$ for a hash-function family $\mathcal{H}$ with assumption $\mathbb{H}$. Then it follows straightforwardly from the definition that B is also secure under $\mathbb{A} \cup \mathbb{B} \supseteq \mathbb{B}$ by the assumption about strict reducibility. Furthermore, the hash-function family $\mathcal{H}$ which makes A secure under $\mathbb{A}$ also makes B secure under $\mathbb{B}$, again by the strict reducibility.

The claim that any strong reducibility implies weak reducibility follows straightforwardly from the definition. □

We next discuss a scheme which supports a strong reduction, but not a strict one. Note that for $\mathbb{A} \subseteq \mathbb{B}$ this claim would be trivial because then the notions coincide. Instead, our separation example even holds for $\mathbb{B} \subsetneq \mathbb{A}$.

Proposition 7 (Strong $\not\Rightarrow$ strict reducibility)**.** *There exists schemes A, B for games A, B and assumptions $\mathbb{A}$, $\mathbb{B}$ such that $\mathbb{B} \subsetneq \mathbb{A}$, and the random oracle of B strongly reduces to the one of scheme A, but not strictly.*

Proof. Let scheme A run two copies of Lamport's one-time signature scheme [Lam79], one based on an alleged one-way function f, and the other one by using the given hash function (oracle). Verification checks if both signatures are valid. Let A be the standard unforgeability game for one-time signature schemes, and let $\mathbb{A}$ be the assumption that an underlying function f is really one way. Let B and B be the same scheme and game, but let $\mathbb{B}$ be the empty set.

Consider the hash-function family $\mathcal{H}$ which samples trivial functions $\mathsf{H} : \{0,1\}^* \to \{0\}$ only and where $\mathbb{H}$ is empty. Then scheme A is still unforgeable if f is one way, independently of the H part of the signature. In contrast, B would be insecure under $\mathbb{B}$ and for the trivial hash-function family, because, by assumption about the "minimalist" approach for the set $\mathbb{B}$, the function f is not one way then. Hence, the random oracle in B cannot be strictly reduced to the one in A.

Finally, note that for a hash-function family $\mathcal{H}'$ which is one way the signature scheme B becomes secure even under $\mathbb{B}$, because any forger would need to forge the one-time signature scheme for the hash function. At the same time, for any hash-function family scheme B is secure under $\mathbb{A} \cup \mathbb{B}$. These two properties show that the random oracle in B strongly reduces to the one in A. □

For the next separation we further need to exclude contrived examples where the hash function assumptions $\mathbb{H}$ "makes up" for assumptions in $\mathbb{A} \setminus \mathbb{B}$ to make scheme B secure. We say that $\mathbb{H}$ is *non-interfering* with $\mathbb{A}$ and $\mathbb{B}$ iff

$\mathbb{H} \cap (\mathbb{A} \setminus \mathbb{B}) = \emptyset$. In this case we say that the random oracle in scheme B reduces to the one in scheme A under non-interfering hash assumptions if reducibility holds for all hash function families $\mathcal{H}$ with non-interfering assumption $\mathbb{H}$.

Proposition 8 (Weak $\not\Rightarrow$ strong reducibility)**.** *There exists schemes A, B for games A, B and assumptions $\mathbb{A}$, $\mathbb{B}$ such that $\mathbb{B} \subsetneq \mathbb{A}$, and the random oracle of B weakly reduces to the one of scheme A, but not strongly for non-interfering hash functions.*

Proof. Consider again Lamport's one-time signature scheme as scheme A, relying on a one-way function f (whose one-wayness is postulated in $\mathbb{A}$). The scheme ignores the hash function. Let B be again the unforgeability game for one-time signature schemes. Let B the same scheme with the same security game, but let $\mathbb{B}$ be empty.

Any hash function makes both schemes secure under assumptions $\mathbb{A} \cup \mathbb{B}$ such that the (irrelevant) random oracle of B weakly reduces to the one of A. Since B cannot be secure assuming only $\mathbb{B}$, because the hash function cannot include the assumption about the one-wayness of f by the non-interference, the scheme cannot strongly reduce the random oracle. □

Uninstantiability implications. In this section we briefly show fundamental results about (un)instantiable random oracles. We define uninstantiability with respect to a very loose requirement on the assumptions, leaving it up to the reduction statement to consider only "standard" cryptographic assumptions in $\mathbb{A}$ and $\mathbb{B}$.

Definition 8 (Uninstantiability)**.** *Let A be $(\mathsf{A}, \mathcal{H})$-secure under assumptions $\mathbb{A}$ for random oracle $\mathcal{H}$. Then the random oracle is* uninstantiable *for A and $\mathbb{A}$ if for any efficient hash-function family $\mathcal{H}$ with assumption $\mathbb{H}$ the scheme A is not $(\mathsf{A}, \mathcal{H})$-secure under assumptions $\mathbb{A}$.*

Proposition 9 (B uninstantiable $\Rightarrow$ A uninstantiable)**.** *Assume that scheme B with game B and assumptions $\mathbb{B}$ is (strictly resp. strongly resp. weakly) polynomial-time RO-reducible to scheme A for A and (true) assumptions $\mathbb{A}$. If B is uninstantiable for B under $\mathbb{B}$ (for strict reductions) resp. $\mathbb{A} \cup \mathbb{B}$ (for strong or weak reduction), then so is A for A and assumptions $\mathbb{A}$.*

Proof. First consider strict reductions. Assume that there exists an instantiation for the hash function in scheme A using assumption $\mathbb{H}$, making it $(\mathsf{A}, \mathcal{H})$-secure under $\mathbb{A}$. Then the hash function described by sampling $\mathsf{H} \leftarrow\!\!\$\ \mathcal{H}$ and setting T^{H} yields an efficient and $(\mathsf{B}, \mathsf{T}^{\mathcal{H}})$-secure instantiation for scheme B under $\mathbb{B}$, because T is computable (deterministically) in polynomial time. This, however, contradicts the uninstantiability for B with respect to B and $\mathbb{B}$. For strong and weak reductions the claim follows accordingly for assumptions $\mathbb{A} \cup \mathbb{B}$. □

Given the uninstantiability notion we next note that there are schemes for which the random oracles are not (even weakly) reducible to each other:

Proposition 10 (Impossibility of reducibility)**.** *There exists schemes A, B for games A, B and (true) assumptions $\mathbb{A}$, $\mathbb{B}$ such that the random oracle of B does not support a weak or strong or strict polynomial-time reduction to the one of scheme A, even though B is secure in the random-oracle model.*

Proof. Basically, we use an instantiable scheme A and an uninstantiable version B of it, following uninstantiability ideas from Maurer et al. [MRH04], to derive the result.

Let A be again the Lamport one-time signature scheme based on a one-way function f, where A is the standard unforgeability game for one-time signature schemes, and assumption $\mathbb{A}$ testifies to the one-wayness of f. The scheme ignores the hash function. Let B be the slightly modified scheme A which inherits the same signing algorithm but where verification, given the hash function H, checks if the message m encodes a hash function; if so, it picks a random element (from a superpolynomial space) and checks that $\mathsf{H}(x) = m(x)$ and in this case accepts. Else it runs the regular verification algorithm. The game and assumption remain unchanged.

It is easy to see that for any efficient hash-function family $\mathcal{H}$ the original scheme A is secure, whereas scheme B can be easily broken by creating the description $m(\cdot) = \mathsf{T}^{\mathsf{H}}(\cdot)$ as a forgery. Furthermore, for random oracle $\mathcal{H}$ the scheme B is also secure, because for a random x the probability that a verification value x matches any of the at most polynomial hash-oracle queries of an attacker, which is necessary to have $m(x) = \mathsf{H}(x)$, is negligible. □

4.4 Example: Hashed ElGamal

In this section we show that the hash function in (a variant) the Twin Diffie–Hellman encryption scheme is RO reducible to the hash function in hashed ElGamal. We remark that we are not aware if the original twin DH scheme allows the same reduction.

Hashed ElGamal. We first review the classical hashed ElGamal encryption scheme as presented in [ABR01]. This scheme, denoted by $A = (\mathsf{KGen}_A, \mathsf{Enc}_A, \mathsf{Dec}_A)$ is based on the Diffie–Hellman problem and uses a hash function H and a symmetric cipher $(\mathsf{Enc}, \mathsf{Dec})$. Specifically:

Construction 1 (Hashed ElGamal encryption scheme)**.** *The hashed ElGamal encryption scheme $A = (\mathsf{KGen}_A, \mathsf{Enc}_A, \mathsf{Dec}_A)$ in the ROM is defined as follows:*

$\underline{\mathsf{KGen}_A(\lambda)\text{:}}$
pick $(\mathcal{G}, g, q)$
$x \leftarrow \mathbb{Z}_q$; $X \leftarrow g^x$
$\mathrm{sk} \leftarrow x$; $\mathrm{pk} \leftarrow (\mathcal{G}, g, q, X)$
return $(\mathrm{sk}, \mathrm{pk})$

$\underline{\mathsf{Enc}_A(\mathrm{pk}, m)\text{:}}$
$(\mathcal{G}, g, q, X) \leftarrow \mathrm{pk}$
$y \leftarrow \mathbb{Z}_q$; $Y \leftarrow g^y$
$Z \leftarrow X^y$; $k \leftarrow \mathsf{H}(Y, Z)$
$c \leftarrow \mathsf{Enc}_k(m)$
return (Y, c)

$\underline{\mathsf{Dec}_A(\mathrm{sk}, Y, c)\text{:}}$
$Z \leftarrow Y^{\mathrm{sk}}$
$k \leftarrow \mathsf{H}(Y, Z)$
$m \leftarrow \mathsf{Dec}_k(c)$
return m

Assuming that the symmetric cipher is secure against single-challenge chosen-ciphertext attacks and that the strong Diffie–Hellman assumption holds (where the adversary has access to a restricted DH decisional oracle), Cramer and Shoup prove in [CS03] that scheme A is secure against chosen-ciphertext attacks if H is a random oracle. The milder ordinary DH assumption is not known to be sufficient to prove CCA security, since the attacker obtains a decision oracle through the decryption oracle here, such that some information about the key may be leaked.

Twin DH scheme. Subsequently, Cash et al. [CKS09] introduce the so-called strong twin DH assumption which holds if and only if the regular DH assumption holds. Their corresponding DH problems are equally hard but the twin case includes access to a decision oracle. This enables a clean security proof for a variant of the hashed ElGamal scheme, because the decryption oracle is not more powerful than the decision oracle in the strong twin DH case. Thus, the twin ElGamal scheme allows for milder number-theoretic assumptions while preserving CCA security.

However, the random oracle in the twin ElGamal scheme is used slightly differently than in the original scheme: its domain is the set of group element triples, as opposed to tuples in the original scheme. While this is unproblematic in the ROM for hash functions $\mathsf{H} : \{0,1\}^* \to \{0,1\}^\lambda$ with arbitrary input length, the implications for other security properties for instantiations are less clear. For example, it may be that the twin Diffie–Hellman scheme demands stronger properties from the hash function. We show via our notion of RO reducibility that this is not the case, at least for our slight variation:

Construction 2 (Twin Diffie–Hellman encryption scheme)**.** *The twin DH encryption scheme* $B = (\mathsf{KGen}_B, \mathsf{Enc}_B, \mathsf{Dec}_B)$ *in the ROM is defined as follows:*

$\underline{\mathsf{KGen}_B(\lambda)\text{:}}$
pick $(\mathcal{G}, g, q)$
$x_0 \leftarrow \mathbb{Z}_q$; $X_0 \leftarrow g^{x_0}$
$x_1 \leftarrow \mathbb{Z}_q$; $X_1 \leftarrow g^{x_1}$
$\mathrm{sk} \leftarrow (x_0, x_1)$
$\mathrm{pk} \leftarrow (\mathcal{G}, g, q, X_0, X_1)$
return $(\mathrm{sk}, \mathrm{pk})$

$\underline{\mathsf{Enc}_B(\mathrm{pk}, m)\text{:}}$
$(\mathcal{G}, g, q, X_0, X_1) \leftarrow \mathrm{pk}$
$y \leftarrow \mathbb{Z}_q$; $Y \leftarrow g^y$
$Z_0 \leftarrow X_0^y$; $Z_1 \leftarrow X_1^y$
$k_0 | k_1 \leftarrow \mathsf{H}(Y, Z_0, Z_1)$
$c \leftarrow \mathsf{Enc}_{k_0}(m)$
return (Y, c, k_1)

$\underline{\mathsf{Dec}_B(\mathrm{sk}, Y, c, k_1')\text{:}}$
$(x_0, x_1) \leftarrow \mathrm{sk}$
$Z_0 \leftarrow Y^{x_0}$
$Z_1 \leftarrow Y^{x_1}$
$k_0 | k_1 \leftarrow \mathsf{H}(Y, Z_0, Z_1)$
$m \leftarrow \mathsf{Dec}_{k_0}(c)$
if $k_1' \neq k_1$
$m \leftarrow \bot$
return m

Towards showing random-oracle reducibility, we can view the transformation $\mathsf{T}^{\mathsf{H}} : \mathcal{G}^3 \to \{0,1\}^{2\lambda}$ of the hash function $\mathsf{H} : \mathcal{G}^2 \to \{0,1\}^{\lambda}$ as follows:

$$\mathsf{T}^{\mathsf{H}}(Y, Z_0, Z_1) = \mathsf{H}(Y, Z_0) | \mathsf{H}(Y, Z_1).$$

Splitting the actual encryption of the message into an encryption for one key half and where we output the other half in clear can then be seen as a special encryption scheme (with double-length keys). In this regard, our version of the twin Diffie–Hellman scheme reduces the random oracle to the hash function of the hashed ElGamal scheme.

RO REDUCIBILITY. We first show that our twin DH scheme *weakly* reduces the random oracle to the one of the hashed ElGamal scheme for IND-CCA security, i.e., assuming the strong DH assumption. We discuss afterward that the scheme is also secure in the random-oracle model assuming the regular DH assumption, implying that the reducibility is also strong:

Theorem 4. *Consider the hashed ElGamal encryption scheme for the IND-CCA security game and the assumptions $\mathbb{A}$ that the symmetric encryption scheme is IND-CCA secure and the strong DH assumption holds. Then the twin DH encryption scheme B with the IND-CCA security game and the assumptions $\mathbb{B}$ that the symmetric encryption scheme is IND-CCA secure and that the DH assumption holds, is strongly RO reducible to the hashed ElGamal encryption scheme via*

$$\mathsf{T}^{\mathsf{H}}(Y, Z_0, Z_1) = \mathsf{H}(Y, Z_0) | \mathsf{H}(Y, Z_1).$$

The proof follows from the following two propositions (11 and 12).

Proposition 11. *Under the assumptions as in Theorem 4 the twin DH encryption scheme is weakly RO reducible to the hashed ElGamal encryption scheme.*

Proof. Assume towards contradiction that there exists an algorithm $\mathcal{B}$ breaking the CCA-security of B. We then describe an adversary $\mathcal{A}$ that breaks the CCA-security of A. This adversary essentially simulates the "second key half" of the scheme by itself. Figure 4.2 summarizes its operation.

DESCRIPTION OF THE REDUCTION. To initialize, the simulation adversary $\mathcal{A}$ on input $(\mathcal{G}, g, q, X_0) = (\mathcal{G}, g, q, g^{x_0})$ chooses the other half of the secret key $x_1 \leftarrow\!\!\$\ \mathbb{Z}_q$ and calculates the corresponding public key $X_1 \leftarrow g^{x_1}$. Adversary $\mathcal{A}$ next runs adversary $\mathcal{B}$ with input $(\mathcal{G}, g, q, (X_0, X_1))$ and answers $\mathcal{B}$'s oracle queries as follows:

- First, $\mathcal{A}$ translates any hash query $\mathsf{H}(A, B, C)$ from $\mathcal{B}$ into two queries to $\mathcal{A}$'s own hash oracle. More precisely, $\mathcal{A}$ answers an (A, B, C) query with $(\mathsf{H}(A, B), \mathsf{H}(A, C)) = \mathsf{T}^{\mathsf{H}}(A, B, C)$.

Algorithm $\mathcal{A}^{\mathrm{H},\textsc{Dec},\textsc{Ch}}(\mathcal{G}, g, q, X_0)$:
$x_1 \leftarrow\!\!\$\ \mathbb{Z}_q$; $X_1 \leftarrow g^{x_1}$
$b \leftarrow \mathcal{B}^{\mathsf{T}^{\mathrm{H}},\textsc{SimDec},\textsc{SimCh}}(\mathcal{G}, g, q, (X_0, X_1))$
return b

Oracle $\textsc{SimDec}(Y, c, k_1)$:
if $k_1 \neq \mathrm{H}(Y, Y^{x_1})$
return $\perp$
return $\textsc{Dec}(Y, c)$

Oracle $\textsc{SimCh}(m_0, m_1)$:
$(Y, c) \leftarrow \textsc{Ch}(m_0, m_1)$
$k_1 \leftarrow \mathrm{H}(Y, Y^{x_1})$
return (Y, c, k_1)

Figure 4.2: The reduction algorithm of Proposition 11

- In order to answer $\mathcal{B}$'s challenge query (m_0, m_1), the adversary submits (m_0, m_1) to his own challenge oracle and parses the corresponding ciphertext answer as (Y, c). It remains to compute the extra value by re-using the randomness Y obtained from the oracle. Adversary $\mathcal{A}$ thus computes $k_1 = \mathsf{H}(Y, Y^{x_1}) = \mathsf{H}(Y, X_1^y)$ and finally returns the ciphertext (Y, c, k_1) to $\mathcal{B}$.

- On a decryption query (Y, c, k_1) of $\mathcal{B}$, adversary $\mathcal{A}$ first checks if (Y, c) corresponds to the value in the challenge ciphertext, or if $k_1 \neq \mathsf{H}(Y, Y^{x_1})$. If so, then $\mathcal{A}$ immediately returns $\perp$. Else $\mathcal{A}$ asks its own decryption oracle for the decryption m of (Y, c). To answer the query, it then returns m.

- Note also that we can grant $\mathcal{B}$ direct access to the H oracle. Adversary $\mathcal{A}$ would simply forward this query and hand back the answer.

When $\mathcal{B}$ eventually outputs a guess b then $\mathcal{A}$ outputs the same bit.

Analysis. The simulation is perfect in the following sense: $\mathcal{B}$ cannot submit a ciphertext (Y, c, k_1^*) to the decryption oracle (after receiving the challenge ciphertext (Y, c, k_1)) for $k_1^* \neq k_1$ which would decrypt correctly. Hence, $\mathcal{A}$ can reject such ciphertexts immediately and therefore only submits "cleaned" ciphertexts to its decryption oracle which have never appeared before. Hence, $\mathcal{A}$, too, represents a successful attacker on the hashed ElGamal scheme if $\mathcal{B}$ is one for the twin DH scheme. Moreover, the advantages of both algorithms in their corresponding IND-CCA game are identical. □

What about inconsistent hash values?. If one recalls the security proof of the hashed ElGamal scheme in the random-oracle model [CS03], then the reduction to the IND-CCA security of the symmetric encryption scheme checks

whether the (external) queries to the hash function correspond to the DH tuple in the challenge ciphertext. In this case the usage of a given encryption oracle with an independent key (which does not necessarily match the "correct" hash value) would be inconsistent. It is the strong DH assumption which implies that such inconsistencies are unlikely to be detected by the adversary against the symmetric encryption scheme. The problem of consistent hash queries, however, is not of our concern in the reduction above. We merely build an adversary $\mathcal{A}$ against the hashed ElGamal scheme out of an adversary $\mathcal{B}$ against the twin DH scheme. The problem, and especially in light of possible proofs without random oracles, has to be taken care of by the security proof for the hashed ElGamal scheme.

To complete the proof for a strong reduction we finally show that our version is secure in the random-oracle model:

Proposition 12. *The twin DH encryption scheme B with the assumptions $\mathbb{B}$ that the symmetric encryption scheme is IND-CCA secure and that the DH assumption holds, is IND-CCA secure in the random-oracle model.*

Proof. The proof is slightly more involved than the one in [CKS09], owned to the fact that the random oracle $\mathsf{H}(X, Z_0, Z_1)$ in [CKS09] ties together the twin DH tuples and that this property is required for the twin DH oracle. In contrast, in our scheme the pairs (X, Z_0) and (X, Z_1) are only loosely connected through H. We show that this loose connection can be made a strong one with two simulations of the adversary. By re-randomizing the instance for the second of the two runs, the reduction will be able to recover not only one, but both of the group elements for the twin DH solution.

In a first step we can "normalize" an adversary $\mathcal{A}$ against IND-CCA of our twin DH scheme. First, we may assume that $\mathcal{A}$ never makes a hash query twice. Second, we can assume that $\mathcal{A}$ never submits a tuple (Y_i, c_i, k_i) to the decryption oracle before receiving the challenge ciphertext (Y, c, k) where $Y_i = Y$. This decreases the adversary's success probability by a negligible amount D/q for the polynomial number D of $\mathcal{A}$'s decryption queries. (Recall that q is the exponential group order.) Third, we can assume that adversary $\mathcal{A}$ never submits a decryption request (Y_i, c_i, k_i) without having queried the hash function about $(Y_i, Y_i^{x_1})$ for $Y_i \neq Y$ before. The loss is at most $D \cdot 2^{-\lambda}$ for this. Fourth, we assume that the adversary never submits (Y_i, c_i, k_i) to the decryption oracle where $Y_i = Y$ but $k_i \neq k$; for such a query is never valid. Fifth, we assume that $X_0 \neq X_1$ which happens with probability $1 - 1/q$.

Taming hash queries. Consider a normalized adversary $\mathcal{A}$ against our twin DH scheme. We assume that $\mathcal{A}$ in addition to T^{H} also has direct access to the random oracle $\mathsf{H} : \mathcal{G}^2 \to \{0,1\}^*$. In fact, we assume from now on that all algorithms, including the adversary and the scheme's algorithms, never call T^{H}, but use H to simulate T^{H} with two queries. Define the following event HashQuery that, during the IND-CCA attack, $\mathcal{A}$ at some point asks a query

(Y, Z) to H such that Y appears in the challenge ciphertext, and $Z = Y^{x_0}$ or $Z = Y^{x_1}$ for the public-key entries $X_0 = g^{x_0}$ and $X_1 = g^{x_1}$.

We show that the probability $\epsilon(\lambda)$ of event HashQuery must be negligible. Assume toward contradiction that this was not the case. We then show how to break the the twin DH problem (and thus the DH problem) via algorithm $\mathcal{B}$. Algorithm $\mathcal{B}$ receives a group description $(\mathcal{G}, g, q)$ and values Y, X_0, X_1 as input. It can also query a twin DH oracle about values (g^a, B_0, B_1) which outputs 1 iff $B_0 = X_0^a$ and $B_1 = X_1^a$. The values X_0, X_1 serve as the public key presented to $\mathcal{A}$, and Y will be placed in the challenge ciphertext.

Algorithm $\mathcal{B}$ runs $\mathcal{A}$'s attack by using the input data as the public key, and simulating the random oracle and decryption queries as follows:

- $\mathcal{B}$ will maintain a list $\mathcal{L}$ of tuples of the form (A, B, k) or $(\mathtt{dh}, A, X_b, k)$ where the former type corresponds to direct hash queries of $\mathcal{A}$ and the latter type to implicit hash queries. Initially, $\mathcal{B}$ sets $\mathcal{L} := \{(\mathtt{dh}, Y, X_0, k_0), (\mathtt{dh}, Y, X_1, k_1)\}$ for random values k_0, k_1 for the hash values to compute the challenge ciphertext (note that Y is already known at the outset).

- Whenever $\mathcal{A}$ makes a hash query (A, B) algorithm $\mathcal{B}$ first searches for an entry (A, C, k) in $\mathcal{L}$ such that (A, B, C) or (A, C, B) forms a correct twin DH tuple (under X_0, X_1). Since $X_0 \neq X_1$ only one case can happen. If found, and there exists an entry $(\mathtt{dh}, A, X_0, k)$ in $\mathcal{L}$ for the case (A, B, C) resp. $(\mathtt{dh}, A, X_1, k)$ for the case (A, C, B), then replace this entry by (A, B, k) in $\mathcal{L}$. In any other case, pick k at random and store (A, B, k) in $\mathcal{L}$. Return k.

- If $\mathcal{A}$ makes a decryption request (Y_i, c_i, k_i) then check whether $Y_i = Y$ or not. In case $Y_i = Y$ look up the entry $(\mathtt{dh}, Y, X_0, k_0)$ in $\mathcal{L}$ and use k_0 to decrypt c_i. (Note that, by assumption, k_1 must be correct.) Suppose $Y_i \neq Y$. Then, since the adversary is normalized, there must be an entry (Y_i, Z_1, k_1) in $\mathcal{L}$ already, caused by a hash query, where $Z_1 = Y_i^{x_1}$. (There cannot exist another entry (Y_i, Z_1, k_1') for $k_1' \neq k_1$ as hash queries never repeat.) Given (Y_i, Z_1, k_1) check for an entry (Y_i, Z_0, k_0) such that (Y_i, Z_0, Z_1) forms a valid twin DH tuple for X_0, X_1. If such an entry exist then use k_0 to decrypt c_i. If no such entry exist, check for a tuple $(\mathtt{dh}, Y_i, X_0, k_0)$ in $\mathcal{L}$ and use k_0 to decrypt. Else, pick a new value k_0, store $(\mathtt{dh}, Y_i, X_0, k_0)$ in $\mathcal{L}$, and use k_0 to decrypt. Return the decrypted message.

To prepare the challenge ciphertext $\mathcal{B}$ uses the previously chosen values k_0, k_1 placed in $\mathcal{L}$, also picks one of the two messages m_0, m_1 at random, and returns $(Y, \mathsf{Enc}(k_0, m_b), k_1)$.

Once $\mathcal{A}$ finishes, algorithm $\mathcal{B}$ records all entries (A, B) in $\mathcal{L}$ with $A = Y$ and now reruns the above procedure, with the same group but for re-randomized data $Y' = Y^s$, $X_0' := X_a^{s_a}$, $X_1' := X_{1-a}^{s_{1-a}}$ for random $s, s_0, s_1 \leftarrow\!\!\$\ \mathbb{Z}_q^*$ and random

bit a. Every other random choice is based on fresh randomness. Any query (A, B, C) to the twin DH oracle in this second run is first transformed into $(A, B^{1/s_0}, C^{1/s_1})$ for $a = 0$ resp. $(A, C^{1/s_0}, B^{1/s_1})$ for $a = 1$. At the end, $\mathcal{B}$ transforms all pairs (A', B') in the list $\mathcal{L}$ of the second run by computing $((A')^{1/s}, (B')^{1/s_0})$ and $((A')^{1/s}, (B')^{1/s_1})$, effectively doubling the number of pairs. Sieve to keep only those with first element Y. Run on all combinations of the two (sieved) lists by the twin DH oracle to find a solution (Y, Z_0, Z_1) to the twin DH problem.

Analysis. The maintenance of the hash list $\mathcal{L}$ provides a more fine-grained implementation of how a random oracle would behave: since any decryption query for $Y_i \neq Y$ must already contain a corresponding entry $(Y_i, Y_i^{x_1}, k_1)$ by assumption, we can check via the twin DH oracle if we already have a matching entry (Y_i, Z_0, k_0). If not, we generate a fresh random string and store the implicit representation $(\mathtt{dh}, Y_i, X_0, k_0)$ in $\mathcal{L}$, and will later carefully check if a hash query for $Y_i^{x_0}$ is made (in which case we update the entry in $\mathcal{L}$ and re-use the value k_0).

As for $\mathcal{B}$'s success probability, we call a group $(\mathcal{G}, g, q)$ good if $\mathcal{A}$'s success probability conditioned on this group exceeds $\epsilon/2$. By an averaging argument a group is good with probability at least $\epsilon/2$. Hence, given such a good group, and the fact that $\mathcal{B}$ provides a perfect simulation, $\mathcal{B}$ obtains a valid entry (Y, Y^{x_0}) or (Y, Y^{x_1}) with probability at least $\epsilon/2$ in the first run. The same applies in the second run where the re-randomization is correctly undone for each twin DH oracle query. With probability $1/2$ algorithm $\mathcal{B}$ then obtains matching values (Y, Y^{x_0}) and (Y, Y^{x_1}) because the order bit a in the second run is information-theoretically hidden from $\mathcal{A}$. Overall, and neglecting the minor loss due to normalization of $\mathcal{A}$, algorithm $\mathcal{B}$ thus solves the twin DH problem with probability at least $\epsilon^3/16$. By assumption this is still non-negligible.

Final reduction to CCA security of symmetric scheme. Given that we can assume that $\mathcal{A}$ never asks the hash function about (Y, Y^{x_0}), it is now straightforward to show that a significant advantage in predicting the challenge bit must result in a security break of the underlying symmetric cipher. We merely sketch this step. We construct an adversary $\mathcal{B}$ from $\mathcal{A}$, where $\mathcal{B}$ plays an IND-CCA game against the symmetric cipher, and simulates $\mathcal{A}$'s attack with the help of the secret keys x_0, x_1 and using lazy sampling to simulate the random oracle (with the only exception that it "virtually puts" the key of the external encryption and decryption oracles as the hash value $\mathsf{H}(Y, Y^{x_0})$). Given the challenge ciphertext (Y, c, k) created by picking y at random, setting $Y = g^y$ and calling the encryption oracle about $\mathcal{A}$'s choice m_0, m_1 to receive c, any subsequent decryption query (Y, c', k) for $c' \neq c$ is answered by calling the external decryption oracle (which is admissible since $c' \neq c$). Any query for $Y' \neq Y$ can be answered by $\mathcal{B}$ itself. Algorithm $\mathcal{B}$ then uses $\mathcal{A}$'s final output to predict the challenge bit for the symmetric scheme. □

4.5 Reductions among Signature Schemes

In this section we briefly outline a few more applications of our notion. Specifically, we give three relations among signature schemes including a probabilistic version of FDH which we reduce to the Guillou–Quisquater (GQ) signature scheme [GQ88] and to the PSS signature scheme [BR96], respectively, and finally a reduction from Schnorr signatures [Sch91] to a (probabilistic version of) BLS signatures [BLS04].

GQ ⇒ pFDH. We first consider the RSA-based Guillou–Quisquater identification scheme and its derived signature scheme via the Fiat-Shamir heuristic [GQ88]. For public key $\mathrm{pk} = (X, N, e)$ and secret key x with $X = x^e \bmod N$ the signer computes a signature as (R, y) for random $R = r^e \bmod N$, and where $y = r^c x \bmod N$ for $c = \mathsf{H}(\mathrm{pk}, R, m)$. A probabilistic full-domain hash (FDH) RSA signature scheme with signatures of the form (R, σ) for $\sigma = (\mathsf{H}(\mathrm{pk}, R, m))^d \bmod N$ is (strictly) random oracle reducible to the aforementioned Guillou–Quisquater scheme via the transformation

$$\mathsf{T}^{\mathsf{H}}(\mathrm{pk}, R, m) = R^{\mathsf{H}(\mathrm{pk},R,m)} X \bmod N$$

for any type of forgery attack under the RSA assumption. The reason is that any Guillou–Quisquater signature for H can be seen as a FDH signature for $\mathsf{T}^{\mathsf{H}} : \{0,1\}^* \to \mathbb{Z}_N^*$, and any successful forgery for the FDH scheme for T^{H} is vice versa a valid forgery for the Guillou–Quisquater scheme.

PSS ⇒ pFDH. The reduction of another probabilistic version of FDH to the PSS signature scheme is similar to the GQ case. Consider FDH signatures $(\mathsf{T}^{\mathsf{H}}(r, m))^d \bmod N$ for the PSS-encoding $\mathsf{T}^{\mathsf{H}}(r, m) = \mathtt{str2int}(0|w|r^*|\mathsf{H}_2(w))$ for $w = \mathsf{H}_0(r, m)$ and $r^* \oplus r = \mathsf{H}_1(w)$. Here, $\mathsf{H}_0, \mathsf{H}_1, \mathsf{H}_2$ are hash functions derived from H as in the PSS scheme. Then any successful attack on FDH with hash function T^{H} easily yields a forgery against PSS with hash function H. Hence, PSS allows a strict random-oracle reduction to the probabilistic version of FDH under the RSA assumption for any type of forgery attack.

BLS ⇒ Schnorr. Consider a probabilistic version of the BLS signature scheme [BLS04], where signatures are of the form $\sigma = (R, \mathsf{H}(R, X, m)^x)$ for randomness R, message m, private key x and public key $X = g^x$. Verification is performed analogously to the original scheme via a pairing computation. We argue that the Schnorr signature scheme (recall that a signature there is of the form $\sigma = (c, r + cx \bmod q)$ for public key $X = g^x$, $R = g^r$, and $c = \mathsf{H}(R, m)$) is (strictly) random oracle reducible to the BLS version via the transformation $\mathsf{T}^{\mathsf{H}}(R, X, m) = RX^{\mathsf{H}(R,m)}$. This holds assuming the discrete logarithm assumption and under an augmented version of the KEA1 assumption [HT98, BP04] which states that, for any adversary $\mathcal{A}$ which for input a

description of the group, g, X, and with access to a Schnorr signing oracle under key X and a hash function oracle, outputs a pair (Y, Y^x), there exists an adversary $\mathcal{A}'$ which, on the same input and with access to the same oracles, outputs y with $X^y = Y^x$. The probability that $\mathcal{A}$ succeeds, but $\mathcal{A}'$ does not, must be negligible for all $\mathcal{A}$.

Suppose now that there exists some successful adversary $\mathcal{B}$ against our version of BLS. Construct adversary $\mathcal{A}$ against the Schnorr scheme as follows. Whenever $\mathcal{B}$ makes some query m, adversary $\mathcal{A}$ forwards this query to its own signing oracle. It uses the answer (c, y) to calculate $h = X^y$, computes $R = g^y X^{-c}$ (such that $\mathsf{H}(R, m) = c$) and finally answers $\mathcal{B}$'s query with (R, h). This simulates a correct signature since $\mathcal{B}$ expects R and $\mathsf{T}^{\mathsf{H}}(R, X, m)^x = (RX^{\mathsf{H}(R,m)})^x = (g^y)^x = X^y = h$. It remains to construct a Schnorr forgery from $\mathcal{B}$'s forgery, denoted by (m^*, R^*, Z^*). To this end we note that, under the augmented KEA1 assumption, for $\mathcal{A}$ (running $\mathcal{B}$ as a subroutine) outputting $Y^* = \mathsf{T}^{\mathsf{H}}(R^*, X, m^*)$ and $Z^* = (Y^*)^x$ for the valid forgery (m^*, R^*, Z^*), there must exist an adversary $\mathcal{A}'$ returning y^* with $Z^* = X^{y^*}$. This must be true with non-negligible probability, because $\mathcal{A}$ succeeds with non-negligible probability, and otherwise the augmented KEA1 assumption would be false. Hence, there exists an adversary which creates a valid forgery $(m^*, \mathsf{H}(R^*, m^*), y^*)$ for the Schnorr scheme with non-negligible probability.

Chapter 5

Notions of Cryptographic Reductions

In this final chapter of the thesis, we study the prevalent proof technique used in almost any result in cryptography. We propose an extensive framework that permits more fine-grained classifications of reductions and covers a wider range of reductions than the approach by Reingold, Vadhan, and Trevisan [RTV04].

We first discuss the necessary background in Section 5.1 and then introduce our new framework in Section 5.2. Section 5.3 puts the framework in action by classifying two well-known reductions. In Section 5.4, we examine the relations of the different notions within the framework before we add efficiency considerations to the picture (in Section 5.5). We then consider parametrized reductions in Section 5.6 that allow for shades of gray between two notions. Finally, Section 5.7 briefly illustrates how one can capture meta reductions in our framework.

This work was presented at ASIACRYPT 2013 [BBF13].

5.1 Introduction

A fundamental question in cryptography refers to the possibility of constructing one primitive from another one. For some important primitives like one-way functions, pseudorandom generators, pseudorandom functions, and signature schemes we know that one can be built from the other one [HILL99, GGM86, Rom90]. For other primitives, however, there are results separating primitives like key agreement or collision-resistant hash functions from one-way functions [IR89, Sim98].

Separations between cryptographic primitives usually refer to a special kind of reductions called *black-box* reductions. These reductions from a primitive $\mathcal{P}$ to another primitive $\mathcal{Q}$ treat the underlying primitive $\mathcal{Q}$ and/or the adversary as a black box. Reingold, Trevisan, and Vadhan [RTV04] ("RTV") suggested

a taxonomy for such reductions which can be divided roughly into three categories:

Fully-black-box reductions. A fully-black-box reduction $\mathcal{S}$ is an efficient algorithm that transforms any (even inefficient) adversary $\mathcal{A}$, breaking any instance G^f of primitive $\mathcal{P}$, into an algorithm $\mathcal{S}^{\mathcal{A},f}$ breaking the instance f of $\mathcal{Q}$. Here, the reduction treats both the adversary as well as the primitive as a black box, and G is the (black-box) construction out of f.

Semi-black-box reductions. In a semi-black-box reduction, for any instance G^f of $\mathcal{P}$, if an efficient adversary $\mathcal{A}^f$ breaks G^f, then there is an algorithm $\mathcal{S}^f$ breaking the instance f of $\mathcal{Q}$. Here, S^f can be tailor-made for $\mathcal{A}$ and f.

Weakly-black-box reductions. In a weakly-black-box reduction, for any instance G^f of $\mathcal{P}$, if an efficient adversary $\mathcal{A}$ (now without access to f) breaks G^f, then there is an algorithm $\mathcal{S}^f$ breaking the instance f of $\mathcal{Q}$.

RTV indicate that the notion of weakly-black-box reductions is close to free reductions (with no restrictions), such that separation results for this type of reduction are presumably hard to find. They discuss further notions like "$\forall\exists$ versions" of the above definitions, where the construction G does not make black-box use of f but may depend arbitrarily on f, and relativizing reductions where security of the primitives should hold relative to any oracle. We discuss these notions later in more detail.

Black-Box Separation Techniques

Known black-box separations usually obey the following two-oracle approach: to separate $\mathcal{P}$ from $\mathcal{Q}$, one oracle essentially makes any instance of $\mathcal{P}$ insecure, whereas the other oracle implements an instance of $\mathcal{Q}$. It follows that one cannot build (in a black-box way) $\mathcal{P}$ out of $\mathcal{Q}$. For example, Impagliazzo and Rudich [IR89] separate key agreement from one-way permutations by using a PSPACE-complete oracle to break any key agreement, and a random-permutation oracle to realize the one-way permutation. This type of separation rules out any reduction proof that remains true in the presence of an arbitrary oracle (i.e., it rules out relativizing reductions). In the key-agreement case, this also rules out semi-black-box-reductions via an embedding of the PSPACE-complete oracle into the black-box primitive [RTV04].

Later, Hsiao and Reyzin [HR04] consider simplified separations for fully-black-box reductions. Roughly speaking, they move the breaking oracle into the adversary such that the reduction can only access this oracle through the adversary (instead of directly, as in [IR89]). Because this makes separations often much more elegant, this technique has been applied successfully for many

other primitives, e.g., [DOP05, HHRS07, KP09, HH09, BCFW09, FLR+10, MP12, LOZ12, BH13].

Interestingly, there has been another type of separations based on so-called meta-reduction techniques, originally exploited by Boneh and Venkatenesan [BV98], but the general concept can be traced back even further to [GMR84]. Subsequently, these techniques are used in many other places [Cor02, PV06, HRS09, FS10, Pas11, GW11, DHT12, Seu12, FF13]. Such meta reductions take an alleged reduction from $\mathcal{P}$ to $\mathcal{Q}$ and show how to use such a reduction to break the primitive $\mathcal{P}$ directly, simulating the adversary for the reduction usually via rewinding techniques. It turns out that meta reductions are somewhat dual to the above notions for black-box reductions. They usually work against reductions which use the adversary only in a black-box way (for this adversary is merely simulated), whereas the reduction often receives the description of the primitive f. This notion then escapes the treatment in [RTV04].

An interesting side effect when the reduction is given the description of f is that then the separation technique still applies to concrete problems like RSA or discrete logarithms, and to constructions which use zero-knowledge proofs relative to f. Such zero-knowledge proofs often rely on Karp reductions of f to an NP-complete language and therefore on the description of f. That means, when restricted to black-box use, these constructions do not work in general, although some of them can still be rescued by augmenting the setup through a zero-knowledge oracle which allows to prove statements relative to f (see [BKSY11]). We also remark that in some cases, such as Barak's ingenious result about non-black-box zero-knowledge and related results [Bar01, BP12], the security relies on the code of the adversary instead, though.

Results in this Chapter

The purpose of this chapter is to complement the notions of fully-, semi-, and weakly-black-box reductions. We also introduce a more fine-grained view on the involved algorithms, such as the distinction between efficient and non-efficient adversaries, or the question in how far the framework can deal with the reduction having partial knowledge about the adversary. We also formalize meta reductions in the new framework and thus enable classification of this type of separation results. We then give a comprehensive picture of the relationship of all reduction types. Next we discuss these results in more detail.

As explained above, we extend the classification of black-box reductions to other types, like meta reductions relying on black-box access to the adversary but allowing to depend on the primitive's representation. This, interestingly, also affects the question of efficiency of the involved algorithms. That is, we believe that reductions for inefficient and efficient adversaries and primitives should in general not be resumed under a single paradigm, if efficiently computable primitives like one-way functions are concerned. For this class, classical separations techniques such as the embedding of the adversarially exploited

PSPACE-complete oracle into the primitive do not work anymore. Hence, in this case, one would need to additionally rely on a complexity assumption, such as for example in the work by Pass et al. [PTV11]. To testify the importance of the distinction between efficient and inefficient adversaries in black-box reductions, we show for example that black-box use of efficient adversaries is equivalent to non-black-box use, for constructions and reductions which are non–black box for the primitive. Another example where the non-black-box use of the primitive turned out to be crucial is in the work by Mahmoody and Pass [MP12], where they build non-interactive commitments from non-black-box one-way functions, whereas constructions out of black-box one-way functions provably fail.

Another issue we address is the question in how far information about the adversary available to the reduction may be considered as covered by black-box notions. Technically speaking, the running time of an efficient fully-black-box reduction must not depend on the adversary's running time, and thus, for example, not on the number of queries the adversary makes to the primitive. Since there is usually no universal, a priori bound for all adversaries on the number of queries, one would then need a non-standard cost model for the adversary's queries to the reduction. (For example, by discounting the time of any computation that is triggered by such a query—that would introduce other problems, though.) We overcome this dilemma by allowing the reduction's running time (or other parameters) to depend on adversarial parameters, such as the number of queries the adversary makes when attacking primitive $\mathcal{P}$. We call this a *parameter-dependent* reduction.

We can go even one step further and give the reduction the adversarial parameters as input. This is necessary, for example, to allow the reduction to depend on the adversary's success probability, but otherwise treating the adversary as a black box. A well-known example of such an "almost" fully-black-box reduction is the security proof of the Goldreich–Levin hardcore predicate [GL89], attributed to Rackoff in [Gol04]. This reduction depends on the adversary's success probability for a majority decision, but does not rely on any specifics of the adversary nor the function to be inverted itself. We call such reductions *parameter aware*.

We note that it is up to the designer of the reduction or separation to precisely specify the parameters. Such parametrized black-box reductions potentially allow authors to counteract the idea behind black-box reductions by placing the adversary's code in the parameters and thus making the reduction fully depend on the adversary again (via a universal Turing machine). But we believe that such trivial cases can be easily detected *if the dependency is signalized clearly*, just as a trivial reduction of a cryptographic protocol to its own security. So far, however, literature seems to be often less explicit on which parameters the reduction is based upon, and if the reduction should really count as black box. Stating reductions clearly as parametrized black-box reductions should make this more prominent.

In summary, we thus provide a more comprehensive and fine-grained view on both black-box constructions and separations, allowing to identify and relate separations more clearly. In our view, two important results are that we can place relativizing reductions between non-black-box constructions for inefficient and for efficient adversaries, and that for efficient adversaries the question of the reduction having black-box access to the adversary, or allowing full dependency on the adversary, is irrelevant. This holds as long as the construction and reduction itself make non-black-box use of the primitive. From a technical point of view, one of the interesting results is that any reduction from the indistinguishability of hardcore bits to one-wayness, such as in the Goldreich–Levin case [GL89], must depend on the adversary's success probability (and thus needs to be parametrized).

5.2 Notions of Reducibility

Since we augment the basic notions of the original RTV framework in various directions, we find it useful to use a different terminology for the reduction types. Instead of referring the original prefixes fully, semi, weakly, and their $\forall\exists$ variants, we use a more descriptive three-character "CAP" notation with words from the language $\{\mathrm{B},\mathrm{N}\}^3$. 'B' in the first position (the C-position) refers to the fact that the Construction is black box, in the second A-position that the Adversary is treated as a black box by the reduction, and in the third P-position the Primitive is treated as a black box by the reduction. Accordingly, an entry 'N' stands for a non-black-box use. From each combination of constraints, we then derive the order of quantification to obtain the actual definitions.

This derivation is essentially a topological ordering for a given set of partial order constraints. If, for example, the construction uses the primitive as a black box, then this translates into the constraint "there exists a construction $G\ldots$ for all primitives f," or, more succinctly, the ordering relation $\exists G \prec \forall f$. Combining all three resulting relations of the CAP notation into an topological ordering then gives the desired definition that respects all constraints simultaneously.

Hence, a fully-black-box reduction in the RTV framework corresponds to a BBB reduction in our notation, and a $\forall\exists$-fully-black-box reduction is an NBB reduction in our sense. The CAP notation will later turn out to be handy when showing implications from an XYZ reduction to an $\widehat{X}\widehat{Y}\widehat{Z}$ reduction, whenever $\widehat{X}\widehat{Y}\widehat{Z}$ is pointwise at most as large as XYZ (with N being smaller than B). It also allows to see immediately that the RTV framework only covers a fraction of all 8 possibilities for the CAP choices (although the NNB type is actually not meaningful, as we discuss later), and that we fill in the missing types BBN, as often ruled out by meta reductions, and the BNB type where the primitive but not the adversary is treated as a black-box.

Extending the RTV framework in another dimension, we differentiate

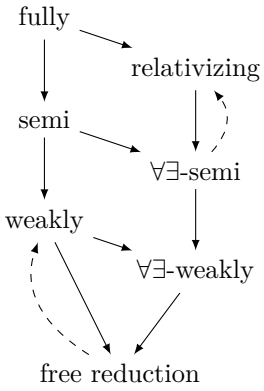


Figure 5.1: Notions of reductions and their relations in the original RTV framework. Dashed arrows indicate equivalence for a restricted class of reductions.

further based on the (in)efficiency of the primitives and adversaries. We append the suffix 'a' to denote an efficiency requirement on the adversary, i.e., a BBBa reduction only works for all probabilistic polynomial-time (PPT) adversaries $\mathcal{A}$, while a BBB reduction is a fully-black-box reduction that transforms *any* adversary $\mathcal{A}$ into an adversary against another primitive. Likewise, we use 'p' to indicate that we restrict primitives to those which are efficiently computable; the suffix 'ap' naturally combines both restrictions.

Overview

At the top of the RTV hierarchy there are fully-black-box reductions—or, BBB reductions in our CAP terminology. These BBB reductions from a primitive $\mathcal{P}$ to a primitive $\mathcal{Q}$ are a pair $(G, \mathcal{S})$ consisting of a construction G and a reduction algorithm $\mathcal{S}$. Both treat the primitive in a black-box way and the reduction treats the adversary in a black-box way. So, for *all* adversaries $\mathcal{A}$ and *all* instantiations f of the primitive $\mathcal{Q}$, we have that, if the adversary $\mathcal{A}^f$ breaks G^f, then the reduction $\mathcal{S}^{\mathcal{A},f}$ with black-box access to the adversary $\mathcal{A}$ and f breaks the implementation f. As a consequence, the existence of primitive $\mathcal{Q}$ implies the existence of the primitive $\mathcal{P}$.

The RTV framework discusses several variants and relaxations of fully-black-box reductions, called semi, weakly, and relativizing reductions (see Figure 5.1). For semi-black-box reductions (aka. BNN reductions) $\mathcal{S}$ can depend on both, the description of the adversary $\mathcal{A}$ and on the instantiation f, and only the construction is black box. For weakly-black-box reductions (which are also of the BNN type), the adversary is additionally restricted to be efficient and does not get access oracle to the primitive (but may depend on it). Lastly, there is a relativizing reduction between the primitives $\mathcal{P}$ and $\mathcal{Q}$, if for all oracles, the primitive $\mathcal{P}$ exists relative to an oracle whenever $\mathcal{Q}$ exists relative to this oracle.

We augment the RTV framework by new classes which represent, among others, reductions that are ruled out by certain meta reductions. That is, we first introduce the notion of BBN reductions where $\mathcal{S}$ has to work for all

(black-box) adversaries, but may depend on the code of f. The other case, where $\mathcal{S}$ is universal for all black-box f but may depend on $\mathcal{A}$, is called a BNB reduction. In both cases the initial 'B' indicates that the construction still makes black-box calls to the primitive. We remark that semi-black-box and weakly-black-box reductions are of the same BNN type in our notation as they only differ in regard to the adversary's access to f. As pointed out in [RTV04] weakly-black-box reductions are close to free reductions, and black-box separations are presumably only possible at the semi level or above. In a sense, our CAP model only captures these levels above, and other types like free or relativizing (or weakly) reductions are special. For the sake of completeness, we symbolically denote (but do not define) weakly reductions wXNN and remark that they essentially correspond to the weakly type of RTV. Note that weakly-black-box reductions are called mildly-black-box in some versions of [RTV04].

The RTV framework also considers the type of construction (black box vs. non–black box) and uses the prefix $\forall\exists$ to indicate that construction G does not need to be universal for all f but can, instead, depend on the description of f. In our CAP terminology this "flips" the initial 'B' to an 'N'. By this, we get 8 combinations, of which 7 are reasonable. The notion of NNB reduction is not meaningful, because we are restricted by the following dependencies: the construction may depend on the primitive, the reduction may depend on the adversary, and the reduction should be universal for the primitive. Thus, there is only one way to order the quantifiers ($\forall\mathcal{A}\exists\mathcal{S}\forall f\exists G$) which does not seem to be a reasonable notion of security, because the construction can now depend on the adversary (and if it does not, we are in the other cases).

We furthermore note that the notion of an NBB reduction is debatable, because it relies on a universal reduction which works for arbitrary constructions. That is, the order of quantifiers is $\exists\mathcal{S}\forall f\exists G\forall\mathcal{A}$. But since there may indeed be such reductions, say, a trivial reduction from a primitive to itself, we do not exclude this type of reduction here.

Definitions of Reductions

We next provide full definitions of BBB (a.k.a. fully-black-box) reductions, BNB and BBN reductions; one can derive the remaining definitions similarly.

A primitive $\mathcal{Q} = (\mathcal{F}_\mathcal{Q}, \mathcal{R}_\mathcal{Q})$ is represented as a set $\mathcal{F}_\mathcal{Q}$ of random variables, corresponding to the set of implementations, and a relation $\mathcal{R}_\mathcal{Q}$ that describes the security of the primitive as tuples of random variables, i.e., a random variable $\mathcal{A}$ is said to break an instantiation $f \in \mathcal{F}_\mathcal{Q}$, if and only if $(f, \mathcal{A}) \in \mathcal{R}_\mathcal{Q}$. Following [RTV04], we say that a primitive exists if there is a polynomial-time computable instantiation $f \in \mathcal{F}_\mathcal{Q}$ such that no polynomial-time random variable breaks the primitive. Indeed, [RTV04] demand that primitive sets $\mathcal{F}_\mathcal{Q}$ are non-empty, and we impose the same restriction.

For efficient primitives or adversaries we stipulate that the random variable is efficiently computable in the underlying machine model which, unless mentioned differently, is assumed to be Turing machines; the results remain valid for other computational models like circuit families. Considering security as a general relation allows to cover various (if not all) notions of security: games such as CMA-UNF for unforgeability of signature schemes, simulation-based notions such as implementing a UC commitment functionality, and even less common notions such as distributional one-way functions. In Section 5.3 we define as examples the DDH assumption (cast as a primitive) and the indistinguishability of the ElGamal encryption scheme. We also review the reduction from the ElGamal encryption to the DDH assumption and identify its type according to our terminology. Note that a "black-boxness" consideration in this particular setting is indeed meaningful, because the DDH assumption can hold in a variety of group distributions and the concrete procedures that sample from these group distributions can be abstracted away. We also discuss another example of weak one-way functions (and the construction of strong one-way functions [Yao82]) to highlight that the type of reduction hinges on the exact formulation of the underlying primitive: the construction and the reduction is then either of the NBN type or of the BBB kind.

We stress that the distinction between the *mathematical object* describing the adversary as a random variable, and its *implementation* through, say, a Turing machine is important here; else one can find counter examples to implications among black-box reduction types proven in [RTV04]. The problem is, roughly, that the relation may simply be secure because it syntactically excludes all oracle Turing machines $\mathcal{A}^f$. We note that Reingold et al. [RTV04] indeed define the relations for adversarial *machines*. Technically, only interpreting such adversaries as abstract objects sustains the implications in [RTV04], but a more relaxed interpretation in the above sense is not too far fetched. However, for sake of convenience, we too often refer to $\mathcal{A}^f$ by the machine implementing it, even when considering the mathematical random process for relations $\mathcal{R}_{\mathcal{Q}}$. In this case it is understood that we actually mean the abstract random variable instead. The same holds for the constructions of the form G^f and the first component of the security relations. An alternative approach would be to rely on machines, but to formally introduce semantical relations. These relations roughly require that, for any algorithm $\mathcal{A}$ in $\mathcal{R}_{\mathcal{Q}}$, any oracle machine $\mathcal{A}^f$ with the same output behavior is also in $\mathcal{R}_{\mathcal{Q}}$.

We now turn to the actual definitions. Many (but not all) reductions in cryptography fall into the class of so-called fully-black-box reductions, a very restrictive notion, where the reduction algorithm is only provided with black-box access to the primitive and the adversary. Throughout the chapter, if there is a XYZ reduction from primitive $\mathcal{P}$ to a primitive $\mathcal{Q}$, we notate this as $\mathcal{P} \hookrightarrow \mathcal{Q}$ XYZ reduction. Note that the correctness requirement is the same for all definitions. Therefore, the shorthand notation towards the end of each definition covers the security requirement only.

Definition 9 ($\mathcal{P} \hookrightarrow \mathcal{Q}$ BBB or fully-black-box reduction)**.** *There exists a* fully-black-box (or BBB) reduction *from a primitive $\mathcal{P} = (\mathcal{F}_\mathcal{P}, \mathcal{R}_\mathcal{P})$ to a primitive $\mathcal{Q} = (\mathcal{F}_\mathcal{Q}, \mathcal{R}_\mathcal{Q})$ if there exist probabilistic polynomial-time oracle algorithms G and $\mathcal{S}$ such that:*

Correctness. *For every $f \in \mathcal{F}_\mathcal{Q}$, it holds that $G^f \in \mathcal{F}_\mathcal{P}$.*

Security. *For every implementation $f \in \mathcal{F}_\mathcal{Q}$ and every machine $\mathcal{A}$, if $(G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P}$, then $(f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}$, i.e.,*

$$\exists \mathsf{PPT} G \ \exists \mathsf{PPT} \mathcal{S} \ \forall f \in \mathcal{F}_\mathcal{Q} \ \forall \mathcal{A} \ ((G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}).$$

Definition 10 ($\mathcal{P} \hookrightarrow \mathcal{Q}$ BNB reduction)**.** *There exists a* BNB reduction *from a primitive $\mathcal{P} = (\mathcal{F}_\mathcal{P}, \mathcal{R}_\mathcal{P})$ to a primitive $\mathcal{Q} = (\mathcal{F}_\mathcal{Q}, \mathcal{R}_\mathcal{Q})$ if there exists a probabilistic polynomial-time oracle machine G such that:*

Correctness. *For every $f \in \mathcal{F}_\mathcal{Q}$, it holds that $G^f \in \mathcal{F}_\mathcal{P}$.*

Security. *For every machine $\mathcal{A}$, there is a probabilistic polynomial-time oracle algorithm $\mathcal{S}$ such that: for every implementation $f \in \mathcal{F}_\mathcal{Q}$, if $(G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P}$, then $(f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}$, i.e.,*

$$\exists \mathsf{PPT} G \ \forall \mathcal{A} \ \exists \mathsf{PPT} \mathcal{S} \ \forall f \in \mathcal{F}_\mathcal{Q} \ ((G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}).$$

Definition 11 ($\mathcal{P} \hookrightarrow \mathcal{Q}$ BBN reduction)**.** *There exists a* BBN reduction *from a primitive $\mathcal{P} = (\mathcal{F}_\mathcal{P}, \mathcal{R}_\mathcal{P})$ to a primitive $\mathcal{Q} = (\mathcal{F}_\mathcal{Q}, \mathcal{R}_\mathcal{Q})$ if there exists a probabilistic polynomial-time oracle machine G such that:*

Correctness. *For every $f \in \mathcal{F}_\mathcal{Q}$, it holds that $G^f \in \mathcal{F}_\mathcal{P}$.*

Security. *For every implementation $f \in \mathcal{F}_\mathcal{Q}$, there is a probabilistic polynomial-time oracle algorithm $\mathcal{S}$ such that for every machine $\mathcal{A}$, if $(G^f, \mathcal{A}) \in \mathcal{R}_\mathcal{P}$, then $(f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}$, i.e.,*

$$\exists \mathsf{PPT} G \ \forall f \in \mathcal{F}_\mathcal{Q} \ \exists \mathsf{PPT} \mathcal{S} \ \forall \mathcal{A} \ ((G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}).$$

Note that we always grant $\mathcal{S}$ black-box access to f and $\mathcal{A}$, as they may not be efficiently computable so that the probabilistic polynomial-time reduction algorithm $\mathcal{S}$ cannot efficiently simulate them, even if it knows the code of f, respectively, of $\mathcal{A}$. For a compact summary of the quantification for all definitions, see Figure 5.2; the written-out definitions omitted above follow from the table.

Name	Quantification			
BBB	$\exists \mathsf{PPT} G$	$\exists \mathsf{PPT} \mathcal{S}$	$\forall f \in \mathcal{F}_{\mathcal{Q}}$	$\forall \mathcal{A}$
BNB	$\exists \mathsf{PPT} G$	$\forall \mathcal{A}$	$\exists \mathsf{PPT} \mathcal{S}$	$\forall f \in \mathcal{F}_{\mathcal{Q}}$
BBN	$\exists \mathsf{PPT} G$	$\forall f \in \mathcal{F}_{\mathcal{Q}}$	$\exists \mathsf{PPT} \mathcal{S}$	$\forall \mathcal{A}$
BNN	$\exists \mathsf{PPT} G$	$\forall f \in \mathcal{F}_{\mathcal{Q}}$	$\forall \mathcal{A}$	$\exists \mathsf{PPT} \mathcal{S}$
NBB	$\exists \mathsf{PPT} \mathcal{S}$	$\forall f \in \mathcal{F}_{\mathcal{Q}}$	$\exists \mathsf{PPT} G$	$\forall \mathcal{A}$
NBN	$\forall f \in \mathcal{F}_{\mathcal{Q}}$	$\exists \mathsf{PPT} G$	$\exists \mathsf{PPT} \mathcal{S}$	$\forall \mathcal{A}$
NNN	$\forall f \in \mathcal{F}_{\mathcal{Q}}$	$\exists \mathsf{PPT} G$	$\forall \mathcal{A}$	$\exists \mathsf{PPT} \mathcal{S}$

Figure 5.2: The quantification of the security statement follows immediately from the CAP notation.

Efficient versus Inefficient Algorithms

Reductions usually run the original adversary as a subroutine. However, in many cases, the reduction does not use the code of the original adversary, but instead only transforms the adversary's inputs and outputs. Thus, one might consider the reduction algorithm as having black-box access to the adversary only. An efficient reduction can then also be given black-box access to an inefficient adversary, and, maybe surprisingly, most reductions even work for inefficient adversaries. Imagine, for example, the case that one extracts a forgery against a signature scheme from a successful intrusion attack against an authenticated channel. Then, the extraction usually still works for inefficient adversaries. On the other hand, (unconditional) impossibility results often require the reduction algorithm to be able to deal with inefficient adversaries.

When designing a fine-grained framework for notions of reducibility, one thus needs to decide whether one considers efficient or inefficient adversaries. Reingold et al. [RTV04] defined their most restrictive notion of reductions, the fully-black-box reductions, for inefficient adversaries. In contrast, their notion of a semi-black-box reduction treats only efficient adversaries thus making it easier to find such a reduction. Surprisingly, even for such a weak notion, they were able to give impossibility results. The reason is that they used inefficient primitives, which allow to embed arbitrary oracles so that they could make use of two-oracle separation techniques. Hence, the efficiency question does not only apply to adversaries, but also to the primitives (and, consequently, to the combination of both). We postpone the treatment of the case of primitives for now and refer the reader to Section 5.5.

We now define the efficient adversary analogues of the notions of reduction introduced earlier. Note that we still give the reduction $\mathcal{S}$ oracle access to the adversary $\mathcal{A}$ in *all* notions, even though the latter can be dropped for all cases where $\mathcal{S}$ depends on $\mathcal{A}$ in a non-black-box way. In these cases, a probabilistic polynomial-time reduction $\mathcal{S}$ can simulate the now likewise efficient adversarial

algorithm $\mathcal{A}$. For consistency, though, we keep the $\mathcal{A}$ oracles in the definitions. To distinguish the two cases of efficient and inefficient adversaries, denote by BBBa reduction a reduction only dealing with efficient adversaries.

Definition 12 ($\mathcal{P} \hookrightarrow \mathcal{Q}$ BBBa reduction for efficient adversaries)**.** *There exists a BBBa reduction from a primitive $\mathcal{P} = (\mathcal{F}_\mathcal{P}, \mathcal{R}_\mathcal{P})$ to a primitive $\mathcal{Q} = (\mathcal{F}_\mathcal{Q}, \mathcal{R}_\mathcal{Q})$ if there exist probabilistic polynomial-time oracle machines G and $\mathcal{S}$ such that:*

Correctness. For every $f \in \mathcal{F}_\mathcal{Q}$, it holds that $G^f \in \mathcal{F}_\mathcal{P}$.

Security. For every implementation $f \in \mathcal{F}_\mathcal{Q}$ and every probabilistic polynomial-time machine $\mathcal{A}$, if $(G^f, \mathcal{A}) \in \mathcal{R}_\mathcal{P}$, then $(f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}$, i.e.,

$$\exists \mathsf{PPT} G \ \exists \mathsf{PPT} \mathcal{S} \ \forall f \in \mathcal{F}_\mathcal{Q} \ \forall \mathsf{PPT} \mathcal{A} \ ((G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}).$$

We omit the enumeration of the other XYZa definitions. Again, the definitions for the remaining types of reductions are readily derived with the help of Figure 5.2.

5.3 Warm Up: Working with CAP

In order to get accustomed to our notation, we now model two well-understood cryptographic tasks and their reductions within our framework.

ElGamal encryption based on the DDH Assumption

We first consider the reduction from the indistinguishability of ElGamal encryption to the DDH problem. It is well known [TY98] that the DDH assumption, basically stating that (g, g^a, g^b, g^{ab}) is indistinguishable from (g, g^a, g^b, g^c), is equivalent to the indistinguishability of the ElGamal encryption, with ciphertexts $(g^r, \mathrm{pk}^r \cdot m)$.

The primitives. One way to capture the DDH assumption as a primitive $\mathcal{Q} = (\mathcal{F}_\mathcal{Q}, \mathcal{R}_\mathcal{Q})$ according to our terminology is to let the set $\mathcal{F}_\mathcal{Q}$ consist of random variables f that output a random group instance whose size is determined by the security parameter input. The relation $\mathcal{R}_\mathcal{Q}$, on input a pair $(f, \mathcal{A})$ of a instance and an adversary, generates such a group with generator g through f, picks a random bit d and random elements a, b, c in the range of the group's order. It then runs the adversary $\mathcal{A}$ on a DDH tuple (g, g^a, g^b, g^{ab}) if $d = 0$, or on a random tuple (g, g^a, g^b, g^c) in case of $d = 1$. The adversary $\mathcal{A}$ is in the relation if it can predict d with non-negligible advantage over $\frac{1}{2}$.

Note that the above is just *one* way to capture the DDH assumption and that there may be others. The choice may also influence the type of reduction we obtain, underlining once more the importance of specifying the primitives clearly. Our choice here matches the idea of the one-way function case, where the functional part provides the core functionality of the primitive,

i.e., allowing the evaluation of the primitive, and the relation part defines its security property. More generally, we can model any falsifiable hardness assumptions (in the sense of [GW11]) analogously, letting the relation take on the role of the challenger in the (possibly interactive) security game with the adversary.

The second primitive, namely $\mathcal{P} = (\mathcal{F}_\mathcal{P}, \mathcal{R}_\mathcal{P})$, represents any IND-CPA secure encryption scheme. Here, the set $\mathcal{F}_\mathcal{P}$ contains all triples of algorithms $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ satisfying the correctness property of a public key encryption scheme. Accordingly, we define $\mathcal{R}_\mathcal{P}$ by saying that $(G, \mathcal{S}) \in \mathcal{R}_\mathcal{P}$ if and only if $\mathcal{S}$ wins the IND-CPA distinguishing game for G with non-negligible probability.

The (c)onstruction. We can construct $\mathcal{P}$ from $\mathcal{Q}$ in the obvious way. That is, we may specify an ElGamal construction G^f that uses black-box access to $f \in \mathcal{F}_\mathcal{Q}$ (as specified above) to obtain a description of a random group. Using this description, the construction performs operations on the group in order to implement the algorithms of the ElGamal scheme. The construction is black box with respect to the "DDH primitive," matching the intuition that for any group (distribution), we obtain an encryption scheme whose security is directly related to the hardness of the DDH assumption in the underlying group (distribution).

The reduction – (a)dversary. Let us briefly recall the interaction between the reduction $\mathcal{S}$ and the adversary $\mathcal{A}$ in order to see that $\mathcal{S}$ uses $\mathcal{A}$ only in a black-box way. The reduction obtains a group description and a challenge triple (g^a, g^b, g^c) as input from the DDH game. Then, it runs the adversary oracle on the public key g^a and embeds the DDH challenge into the challenge ciphertext during the simulation of the IND-CPA game, i.e., it calculates $C \leftarrow (g^b, g^c \cdot m_d)$ for a randomly chosen bit d and returns C to the adversary. Finally, the reduction outputs 1 if and only if the adversary's output d' matches d. Hence, the reduction only uses the adversary as an oracle.

The reduction – (p)rimitive. Here, the same discussion as for the construction applies—the primitive is treated as a black box; the reduction merely performs group operations on the group that is generated by the primitive oracle. This matches the intuition that the reduction works for an arbitrary group (distribution).

In conclusion we hence have a BBB reduction in this case.

Amplification of One-Way Functions

As a second example, let us consider the primitive $\mathcal{Q} = (\mathcal{F}_\mathcal{Q}, \mathcal{R}_\mathcal{Q})$ describing weak one-way functions, i.e., functions f for which there exists a function $\epsilon(n)$ bounded away non-negligibly from 1, such that for any PPT adversary $\mathcal{A}$ we have

$$\Pr\left[\mathcal{A}(1^n, f(x)) \to x' \in f^{-1}(x)\right] \leq \epsilon(n) + \mathsf{negl}(n).$$

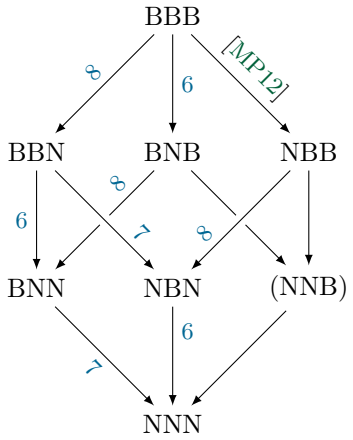


Figure 5.3: The relations among the basic CAP notions form a cube. Arrows indicate an implication (via Theorem 5). All implications are strict; the number/citation on the arrow indicates the theorem supporting the strictness claim. The NNB notion is shown for completeness only.

The relation $\mathcal{R}_{\mathcal{Q}}$ associates to each $f \in \mathcal{F}_{\mathcal{Q}}$ a function ϵ as above, and contains a pair $(f, \mathcal{A})$ if $\mathcal{A}$'s inversion probability exceeds $\epsilon(n)$ non-negligibly.

The concatenation construction which evaluates f for $\Theta(n/\epsilon(n))$ independent inputs yields a (strong) one-way function [Yao82]. Note that in the construction, the number of function evaluations depends on the specific parameter ϵ of the weak one-way function. Since the reduction only makes black-box use of the adversary but also relies on ϵ, the transformation is thus an NBN reduction in our terminology.

However, if we change the viewpoint slightly, and define the primitive $\mathcal{Q}_\epsilon = (\mathcal{F}_{\mathcal{Q}_\epsilon}, \mathcal{R}_{\mathcal{Q}_\epsilon})$ to contain all functions for some global bound ϵ, i.e., such that $\mathcal{R}_{\mathcal{Q}_\epsilon}$ consists of all pairs $(f, \mathcal{A})$ with $f \in \mathcal{F}_{\mathcal{Q}_\epsilon}$ and where $\mathcal{A}$'s inversion probability is non-negligibly larger than $\epsilon(n)$, then we obtain a BBB reduction for the concatenation construction (and its reduction) based on the same global ϵ. This shows that the type of reduction critically depends on the definition of the primitive.

5.4 Relations Amongst the Definitions

Naturally, the next question is how the definitions are related. We first note that a number of implications among the reductions is immediately clear by simply shifting quantifiers, that is, if we have a for-all quantifier, then there is certainly an existential version of the reduction in question. The next proposition states this formally, we omit the proof because it is only syntactical.

Theorem 5. *Let XYZ and $\widehat{X}\widehat{Y}\widehat{Z}$ be two types of CAP reductions such that $\widehat{X}\widehat{Y}\widehat{Z} \leq XYZ$ point-wise (where $N \leq B$) and let $\mathcal{P}$ and $\mathcal{Q}$ be two primitives. If there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ XYZ reduction, then there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ $\widehat{X}\widehat{Y}\widehat{Z}$ reduction. Also, if there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ $XYZa$ reduction, then there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ $\widehat{X}\widehat{Y}\widehat{Z}a$ reduction.*

In the remainder of this section, we prove via means of counterexamples that for all notions for inefficient adversaries, all the above implications are, indeed,

strict; see Figure 5.3. These separations are split into a number of interesting observations. For example, we prove that the Goldreich–Levin hardcore bit reduction [GL89] has to depend on the success probability of the adversary (Theorem 6). Moreover, we show that the construction of one-way functions out of weak one-way functions ([Yao82, GIL+90]) needs to depend on the weakness parameter of the weak one-way function (Theorem 7). RTV conjecture that there is no inherent restriction in treating the adversary as a black box. Jumping ahead, Theorem 9 in the next section partially confirms this conjecture, for efficient adversaries. For notions of inefficient adversaries, however, non-black-box use of the adversary is a promising approach to overcome existing impossibility results, as Theorem 8 shows. Namely, we prove that for inefficient adversaries, BNB reductions do not imply BBB reductions, and BNN reductions do not imply BBN reductions. For both separations, we will consider a reduction that has to depend on the adversary in a non-black-box way, namely the Goldreich–Levin hardcore bit reduction [GL89].

Theorem 6. *There are primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is an $\mathcal{P} \hookrightarrow \mathcal{Q}$ BNB reduction, but no BBB reduction. For the same two primitives, there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ BNN reduction, but no BBN reduction, as well as a $\mathcal{P} \hookrightarrow \mathcal{Q}$ NNN reduction, but no NBN reduction.*

Proof. We will prove that there are two primitives such that there is an BNB reduction and thus, by Theorem 5 also a BNN and a NNN reduction. We will then prove that for the same two primitives, there is no NBN reduction and therefore (again according to Theorem 5) neither a BBN, nor a BBB reduction. The common element here is, of course, that for a reduction between the primitives, the reduction needs to depend on the adversary in a non-black-box way. We will see that the Goldreich–Levin hardcore bit construction has this property.

We define primitives $\mathcal{Q}$ and $\mathcal{P}$ both as random oracles for length-doubling functions with different interfaces/security games/security relations. In short, $\mathcal{Q}$ is the game for one-wayness of a random oracle, and $\mathcal{P}$ is the game for the prediction of the Goldreich–Levin hardcore bit. For security parameter λ, the input length to the functions is λ. To break $\mathcal{Q}$, the adversary is given $f(x)$ for a random x and may query the random oracle. The adversary is successful if it determines x' with $f(x') = f(x)$ with non-negligible probability. To break $\mathcal{P}$, an adversary is given $f(x)$ for a random x and a random string r of length $|x|$. That is, G^f is the primitive which samples x, r and outputs $f(x)$ and r. The adversary wins if it can determine the inner product $\langle x, r\rangle$ over $\mathbb{Z}_2$ with a non-negligible advantage over $\frac{1}{2}$.

The Goldreich–Levin reduction [GL89] now proves that if there is a successful adversary $\mathcal{A}$ against $\mathcal{P}$ then we can invert the one-way function. As the reduction highly depends on the success probability of the adversary $\mathcal{A}$ but is black box with respect to the implementation of the one-way function, and

as, likewise, the hardcore-bit construction is black box, the Goldreich–Levin reduction is a BNB reduction.

If we can prove that the dependence of the reduction on the adversary's success probability is necessary, then we know that there is no NBN reduction from $\mathcal{P}$ to $\mathcal{Q}$. However, as our primitives are random oracles, this follows by a simple information-theoretic argument, namely, if the reduction asks the adversary, say, p times while the adversary's success probability is less than, say, $\frac{\lambda}{4p}$, then, on average, the reduction can extract at most $p \cdot \frac{\lambda}{4p} = \frac{\lambda}{4}$ bits of information from the adversary. Using Hoeffding's bound, we obtain that the adversary gives a correct response at most $\frac{\lambda}{2}$ times, which is not enough to compute a preimage of length λ. We now state this argument formally.

Let $p(\lambda) \geq \lambda^2$ be an upper bound on the running time of the reduction $\mathcal{S}$, then $\mathcal{S}$ queries $\mathcal{A}$ at most $p(\lambda)$ times. We will now construct an (inefficient) adversary $\mathcal{A}$ that has non-negligible winning advantage against $\mathcal{P}$, and yet, $\mathcal{S}$ only breaks the one-wayness of $\mathcal{Q}$ with negligible probability when given access to $\mathcal{A}$. Let $\epsilon(\lambda) := \frac{\lambda}{4p(\lambda)}$. On input $(f(x), r)$, the adversary $\mathcal{A}$ returns a random bit with probability $1 - \epsilon$. With probability ϵ, the adversary computes the smallest x' such that $f(x) = f(x')$ and returns $\langle x', r \rangle$. Note that with overwhelming probability, for a random x, there is no second preimage for $f(x)$ under f, as it is a length-doubling random function: recall that f is a random function, hence the probability for a second preimage for *any* image is $1 - (1 - 2^{-2n})^{2^n} \leq 2^{-n}$ (using Bernoulli's inequality). Thus, the adversary $\mathcal{A}$'s success probability is negligibly close to $\frac{1}{2} + \epsilon$, and the adversary therefore breaks primitive $\mathcal{P}$.

We now prove that with overwhelming probability, the adversary does not decide to return the correct answer more than $\frac{\lambda}{2}$ times. Towards this goal, consider the Chernoff–Hoeffding bound for independent Bernoulli random variables X_i that all have mean μ, and let $0 \leq \delta \leq 1$ be a parameter,

$$\Pr\left[\sum_{i=1}^{n} X_i > (1+\delta)n\mu\right] \leq e^{-\frac{n\mu\delta^2}{3}}.$$

We set $X_i = 0$, if the adversary decides to return a random bit on the ith query and $X_i = 1$, if the adversary decides to return the correct answer. Then, μ equals $\epsilon(\lambda) = \frac{\lambda}{4p(\lambda)}$. We set $n := p(\lambda)$ as the upper bound on the number of queries made by the reduction. Set $\delta := 1$, then the probability that the adversary decides to return the correct answer more than $\frac{\lambda}{2}$ times is negligible, i.e.:

$$\Pr\left[\sum_{i=1}^{p(\lambda)} X_i > \frac{\lambda}{2}\right] = \Pr\left[\sum_{i=1}^{p(\lambda)} X_i > (1+1)p(\lambda)\tfrac{\lambda}{4p(\lambda)}\right] \leq e^{-p(\lambda)\frac{\lambda}{4p(\lambda)}\frac{1^2}{3}} = e^{-\frac{\lambda}{12}}.$$

We conclude that the probability that the adversary decides to return a correct answer (and not a random reply) more than $\frac{\lambda}{2}$ times when being invoked $p(n)$

times, is negligible. As a thought experiment, we can thus replace the (stateless) adversary $\mathcal{A}$ by a stateful adversary $\mathcal{A}'$ which draws a set of $\frac{\lambda}{2}$ random indices between 1 and $p(n)$ in the beginning. Then, whenever the index is in this set, the adversary computes the smallest x' such that $f(x) = f(x')$ and returns $\langle x', r\rangle$. Else, the adversary returns a random response. Moreover, we can have $\mathcal{A}'$ to return some additional information indicating the reply's correctness, namely 0, if it outputs a random bit, and 1, if it returns the actual bit $\langle x', r\rangle$. This adversary is actually "more helpful" to the reduction than the original one, because the reduction knows which bits are correct and which bits are random. In a next step, since the other answers of $\mathcal{A}'$ are all random bits, we can replace $\mathcal{A}'$ by the adversary $\mathcal{A}''$ who can only be queried $\frac{\lambda}{2}$ times and who always returns $\langle x', r\rangle$ on input $(f(x), r)$, where x' is the smallest element such that $f(x) = f(x')$. (Note that any query on some input $(f(\tilde{x}), r)$ where $\tilde{x} \neq x$ is not helpful to the reduction since f is a random function.)

It remains to prove that no efficient reduction can invert f on a random input when being allowed $\frac{\lambda}{2}$ queries to $\mathcal{A}''$. Towards this goal, we consider the random oracle via lazy sampling.

Before making a query to $\mathcal{A}''$, the reduction's probability of finding a preimage of its input $y = f(x)$ in a single query is roughly $2^{-|y|} + 2^{-|x|} = 2^{-2\lambda} + 2^{-\lambda}$, i.e., the probability that y is sampled as the answer plus the probability, that the reduction queries the real preimage x. After learning the inner product of x with some value r, the preimage space for x is divided into two halves, so that the reduction's success probability increases to $2^{-2\lambda} + 2 \cdot 2^{-\lambda}$ per query, which is still negligible. Repeating this process $\frac{\lambda}{2}$ times yields a success probability of roughly $2^{-2\lambda} + 2^{\lambda/2} \cdot 2^{-\lambda}$ which is still negligible. □

For the following theorem, we interpret the results of Lin, Trevisan, and Wee [LTW05] and Yao [Yao82] as an instance of our framework. Namely, Yao [Yao82] shows how to construct strong one-way functions out of weak one-way functions via an NBN reduction, while Lin, Trevisan, and Wee [LTW05] show that any such construction has to depend on the weakness parameter of the weak one-way function. In other words, one cannot have any BYZ reduction between these two primitives.

Theorem 7. *There exists primitives $\mathcal{P}$ and $\mathcal{Q}$ such that for all $YZ \in \{BN, BNa, NN, NNa\}$, there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ NYZ reduction but there is no $\mathcal{P} \hookrightarrow \mathcal{Q}$ BYZ reduction.*

Proof. For completeness, we now review the construction by Yao [Yao82] and also give a simple impossibility in the spirit of Lin, Trevisan and Wee [LTW05] to explain why one cannot build strong one-way functions out of weak one-way functions via a BYZ reduction.

Recall that a one-way function is a function that is hard to invert on a random input, i.e., for all efficient adversaries $\mathcal{A}$ we have

$$\Pr\left[\mathcal{A}(1^n, f(x)) \to x' \in f^{-1}(x)\right] \leq \mathsf{negl}(n).$$

A *weak* one-way function is a function that is one-way on a certain fraction on its input domain. In other words, a weak one-way function is secure if there is a function $\epsilon(n)$ bounded away non-negligibly from 1, such that for any efficient adversary $\mathcal{A}$ the inverting probability is essentially at most $\epsilon(n)$. Formally,

$$\Pr\left[\mathcal{A}(1^n, f(x)) \to x' \in f^{-1}(x)\right] \leq \epsilon(n) + \mathsf{negl}(n).$$

Yao [Yao82] proved that any weak one-way function can be transformed into a one-way function via concatenation, i.e., if f is a weak one-way function with parameter $\epsilon(n)$, then for $k := n \cdot \left\lceil \frac{1}{\epsilon(n)} \right\rceil$, one has that

$$G^f(x_1|...|x_k) := f(x_1)|...|f(x_k), \text{ where } |x_i| = n,$$

is a one-way function. The reason is, that with overwhelming probability, for a random $x = x_1|...|x_k$ at least one of the x_i lies in the hard ϵ-fraction of the weak one-way function f. This is a non-black-box construction, as we use the parameter ϵ to construct G. Note that, depending on f and G, the adversary $\mathcal{A}$ now expects inputs of a certain format and thus, Yao's reduction is only black-box with respect to the adversary, but not with respect to the function f. Thus, it is an NBN reduction. By Theorem 5, it is also an NNN reduction, and due to Theorem 10, these reductions also work when restricted to efficient adversaries.

We now give some intuition why the dependence on ϵ is necessary. Consider towards contradiction Yao's concatenation operator construction, where the number of queries k that the construction makes to f does not depend on the parameter $\epsilon(n)$ which determines the degree of one-wayness of the weak one-way function f. Assume that the concatenation construction is the above construction for some value $k = n \cdot n^c$ with $c > 1$ (the proof holds in particular for any k for which $n \cdot n^c$ is an upper bound). Then, set $\epsilon(n) := n^{-2c}$ and let f be a length-preserving function that behaves like a random oracle on an $\epsilon(n)$ fraction of its inputs, and let f return the all-zero string otherwise. Formally, let R be a length-preserving random oracle, and define

$$f(x_i) := \begin{cases} R(x_i) & \text{if } \langle x_i \rangle < \epsilon(|x_i|)2^{|x_i|}; \\ 0^{|x_i|} & \text{otherwise.} \end{cases}$$

Here, $\langle x_i \rangle$ denotes the value of x_i when x_i is interpreted as a natural number. Let $\mathcal{A}$ be the adversary that on input $(1^n, y)$ returns a random value $x \leftarrow_\$ \{0,1\}^{|y|}$. We prove that $\mathcal{A}$ has a noticeable winning probability against G^f. Towards this goal, we show that with noticeable probability over x, the

construction returns the all-zero string, i.e., $G^f(x) = 0^{|x|}$. If the challenge value is the all-zero string and if the adversary picks a preimage of an all-zero string, then the adversary is successful. As we will show, both events happen with noticeable probability and as both events are independent (the adversary ignores its input challenge), the overall success probability of the adversary is noticeable, too.

Recall that $k = n \cdot n^c$. For a random $x = x_1|...|x_k$, we have that the probability that $G^f(x) = 0^{|x|}$, i.e., that for all i, it holds that $\langle x_i \rangle \geq \epsilon(|x_i|)2^{|x_i|}$ is lower bounded by

$$\begin{aligned}
\Pr_{x_i}\left[G^f(x) = 0^{|x|}\right] &\geq (1-\epsilon(n))^k \\
&\geq (1-n^{-2c})^{n \cdot n^c} \\
&\geq 1 - n^{-2c} \cdot n \cdot n^c \qquad (5.1) \\
&\geq 1 - n^{-(c-1)} \qquad (5.2) \\
&\geq \tfrac{7}{8},
\end{aligned}$$

where (5.1) follows from the Bernoulli inequality, stating that $(1+z)^r \geq 1+rz$ for $r \geq 0$ and $z \geq -1$ and (5.2) holds for sufficiently large security parameters n. We see that the probability that $G^f(x) = 0^{|x|}$ for a random x is greater than $\frac{7}{8}$, and so is the probability that the adversary's challenge is the all-zero string. Likewise, the probability that $\mathcal{A}$ returns a preimage of an all-zero string is greater than $\frac{7}{8}$. As both events are independent, the probability that both events happen is greater than $\frac{7}{8} \cdot \frac{7}{8}$ and thus, $\mathcal{A}$ has a noticeable success probability against G^f. This concludes that any construction using $k = n \cdot n^c$ (independent of ϵ) must be insecure.

For general constructions, we refer the reader to Lin, Trevisan, and Wee [LTW05]. Note that they do not only show that the dependence on ϵ is necessary, they also prove quantitative lower bounds on the number of queries that the construction needs to make. □

Theorem 8. *For $X \in \{N, B\}$, there exist primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ XBN reduction, a $\mathcal{P} \hookrightarrow \mathcal{Q}$ XBNa reduction, a BNN reduction and BNNa reduction but such that there is no $\mathcal{P} \hookrightarrow \mathcal{Q}$ XBB/XBBa/BNB/BNBa reduction.*

Proof. As in previous proofs, we will show two separations via a single separation, namely, we define two primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ BBN reduction (and thus an $\mathcal{P} \hookrightarrow \mathcal{Q}$ NBN reduction via Theorem 5, as well as an NBNa reduction and a BBNa reduction), but no $\mathcal{P} \hookrightarrow \mathcal{Q}$ NBBa reduction (and thus no BBBa/BBB/NBBa reduction). We will then show the case BNB/BNBa versus BBN/BBNa separately.

For the primitive $\mathcal{P}$, we consider a trivial primitive, namely the constant zero function, denoted f_z. The pair $(f_z, \mathcal{A})$ is in $\mathcal{R}_\mathcal{P}$ for all adversaries $\mathcal{A}$. For

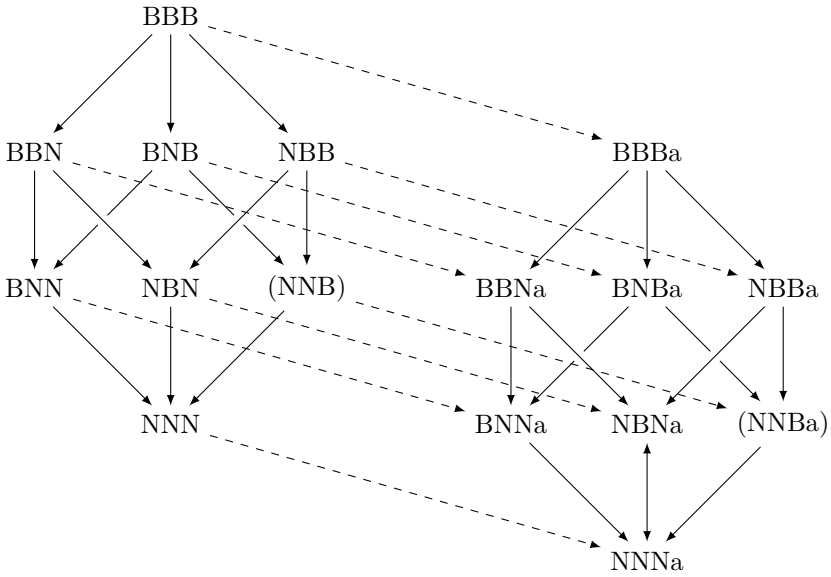


Figure 5.4: Introducing efficient adversaries unfolds another dimension with similar relations. The dashed arrows designate (strict) implications.

the primitive $\mathcal{Q}$, we consider the two random variables f_0 and f_1 that take as input a string. The random variable f_b returns 1, if the first bit of the input string is b, and 0, else. We define $(f_b, \mathcal{A}) \in \mathcal{R}_{\mathcal{Q}}$ if and only if $\mathcal{A}$ on input $\perp$ queries $b|x$ to the primitive with probability 1 for $b \in \{0,1\}$ and $x \in \{0,1\}^*$. In the BBNa reduction, the reduction $\mathcal{S}$ may depend on the primitive $\mathcal{Q}$ and ignores the given adversary $\mathcal{A}$. If the primitive $\mathcal{Q}$ is instantiated by f_0, then the reduction $\mathcal{S}$ constantly returns 0. If the primitive $\mathcal{Q}$ is instantiated by f_1, then the reduction $\mathcal{S}$ constantly returns 1. Thus, there is a BBN reduction. In contrast, there is no NBBa reduction, as we can consider the adversary $\mathcal{A}$ that does nothing (and is still successful by definition) and then, $\mathcal{S} = \mathcal{S}^{\mathcal{A}}$ cannot return both, 0 and 1 with probability 1. Thus, $\mathcal{S}$ fails for at least one of the two primitives $f_0, f_1 \in \mathcal{F}_{\mathcal{Q}}$, which concludes the proof.

By inspection, the given reduction is also a BNN/BNNa reduction, and the impossibility considerations also apply to BNN/BNNa reductions. □

5.5 Adding Efficiency

Once we consider efficient adversaries, the picture of Figure 5.3 grows by one dimension; Figure 5.4 illustrates the result. Interestingly, some of the implications of Theorem 5 are not strict when one is concerned with reductions for efficient adversaries. Maybe surprisingly, NNNa reductions and NBNa reductions are, indeed, equivalent. Note that this means that knowledge of the code of the adversary does not lend additional power to the reduction:

Theorem 9 (Equivalence of NNNa and NBNa)**.** *For all primitives $\mathcal{P}$ and $\mathcal{Q}$, there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ NBNa reduction if and only if there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ NNNa reduction.*

Proof. Using straightforward logical deductions like in Theorem 5, it follows that NBNa-reductions imply NNNa reductions. For the converse direction, assume that we have two primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ NNNa reduction. We now have to show that there also is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ NBNa reduction, that is, we have to give a reduction algorithm $\mathcal{S}$ that depends on f in a non-black-box way, and yet $\mathcal{S}$ depends on $\mathcal{A}$ only in a black-box way. We proceed by case distinction over f.

Case I: Suppose $f \in \mathcal{F}_{\mathcal{Q}}$ such that for all constructions G, the primitive G^f is a secure implementation of $\mathcal{P}$, i.e., for all polynomial-time adversaries $\mathcal{A}$ it holds that $(G^f, \mathcal{A}^f) \notin \mathcal{R}_{\mathcal{P}}$. Then, proving the existence of a reduction satisfying the implication $(G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{Q}}$ is trivial, as the premise of the implication is never satisfied.

Case II: For any $f \in \mathcal{F}_{\mathcal{Q}}$ outside the class described in Case I, we know that there exists a PPT construction G such that for all $\mathcal{A}$ there is a reduction algorithm $\mathcal{S}$ that satisfies $(G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{Q}}$, and such an efficient $\mathcal{A}$ with $(G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}}$ exists. For any such f, we now fix a unique adversary $\mathcal{A}_f$, say, by taking the random variable $\mathcal{A}_f$ with the shortest description according to a particular encoding, such that it satisfies $(G^f, \mathcal{A}_f^f) \in \mathcal{R}_{\mathcal{P}}$. For such an $\mathcal{A}_f$ let $\mathcal{S}$ be a probabilistic polynomial-time reduction making black-box use of $\mathcal{A}_f$ such that $(f, \mathcal{S}^{\mathcal{A}_f,f}) \in \mathcal{R}_{\mathcal{Q}}$. Consider the oracle algorithm $\mathcal{S}_f^f$ that has the same behavior as $\mathcal{S}^{\mathcal{A}_f,f}$, but it incorporates $\mathcal{A}_f$ and only has an oracle for f. The algorithm $\mathcal{S}_f^f$

- only depends on f,
- satisfies $(\mathcal{S}_f^f, f) \in \mathcal{R}_{\mathcal{Q}}$, and
- is implementable in probabilistic polynomial time, as $\mathcal{S}$ and $\mathcal{A}_f$ are both polynomial time algorithms.

Thus, regardless of construction G, we showed that for all f there is an efficient reduction $\mathcal{S}$ such that $(\mathcal{S}^{\mathcal{A},f}, f) \in \mathcal{R}_{\mathcal{Q}}$, namely by choosing $\mathcal{S}^f = \mathcal{S}_f^f$. Thus, we also know that for all f, there is a reduction $\mathcal{S}$ such that for all $\mathcal{A}$, if $(\mathcal{A}, G^f) \in \mathcal{R}_P$ then $(\mathcal{S}^{\mathcal{A},f}, f) \in \mathcal{R}_{\mathcal{Q}}$. If now, we add an adversary oracle $\mathcal{A}$ that is used by $\mathcal{S}$ (here, we require that the relation be machine independent), we also obtain that $(\mathcal{S}^{\mathcal{A},f}, f) \in \mathcal{R}_{\mathcal{Q}}$. And thus, there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ NBNa reduction. □

We now show that, while a reduction for inefficient adversaries always implies a reduction for efficient adversaries of the same type, the converse is not true in general.

Theorem 10. *For each $XYZ \in \{BBB, BNB, BBN, NBB, BNN, NBN, NNN\}$, there are primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ $XYZa$ reduction, but no $\mathcal{P} \hookrightarrow \mathcal{Q}$ XYZ reduction.*

Proof. For the primitive $\mathcal{P}$ we consider a trivial primitive, namely the constant all-zero function, denoted f_z. Let $\mathcal{L}$ be an EXPTIME-complete problem. The pair $(f_z, \mathcal{A})$ is in the relation $\mathcal{R}_\mathcal{P}$ if and only if the adversary $\mathcal{A}$ is a deterministic function that decides $\mathcal{L}$. Let $\mathcal{F}_\mathcal{Q}$ also consist of the set that only contains the all-zero function f_z. The relation $\mathcal{R}_\mathcal{Q}$ is empty. Observe that, for efficient adversaries, the primitive $\mathcal{P}$ is secure because EXPTIME strictly contains the complexity class P [HS65]. Thus, there is a trivial reduction since the premise of the implication

$$(G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}$$

is never satisfied for any efficient adversary $\mathcal{A}$. Hence, for all $XYZ \in \{\text{BBB}, \text{BNB}, \text{BBN}, \text{NBB}, \text{BNN}, \text{NBN}, \text{NNN}\}$, there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ XYZa reduction. In contrast, inefficient adversaries can break the primitive $\mathcal{P}$, while, as $\mathcal{R}_\mathcal{Q}$ is empty, no reduction $\mathcal{S}$ can break $\mathcal{R}_\mathcal{Q}$, even oracle $\mathcal{A}$. Thus, for all $XYZ \in \{\text{BBB}, \text{BNB}, \text{BBN}, \text{NBB}, \text{BNN}, \text{NBN}, \text{NNN}\}$, there is no $\mathcal{P} \hookrightarrow \mathcal{Q}$ XYZ reduction. □

Relativizing Reductions

In complexity theory as in cryptography, most reductions relativize in the presence of oracles. A reduction relativizes, if it remains true in the presence of arbitrary oracles. More concretely, if a (secure instantiation of the) primitive $\mathcal{P}$ can be built from a (secure instantiation of the) primitive $\mathcal{Q}$, then the construction still works, if additionally, all parties get access to, say, a random oracle. We say that there is a *relativizing reduction* from $\mathcal{P}$ to $\mathcal{Q}$, if for all oracles Π, the primitive $\mathcal{P}$ exists relative to Π, whenever $\mathcal{Q}$ exists relative to Π. Often, separation results rule out such reductions.

Definition 13 (Relativizing reduction)**.** *There exists a* relativizing reduction *from a primitive $\mathcal{P}$ to a primitive $\mathcal{Q}$, if for all oracles Π, the primitive $\mathcal{P}$ exists relative to Π whenever $\mathcal{Q}$ exists relative to Π. A primitive $\mathcal{P}$ is said to exist relative to Π if there is an $f \in \mathcal{F}_\mathcal{P}$ which has an efficient implementation when having access to the oracle Π such that there is no probabilistic polynomial-time algorithm $\mathcal{A}$ with $(f, \mathcal{A}^{\Pi,f}) \in \mathcal{R}_\mathcal{P}$.*

We remark that, since we define security relations over random variables and not their implementations, it is understood that the implementation of f may actually depend on Π, too. According to Reingold et al. [RTV04], relativizing reductions are a relatively restrictive notion of reducibility that they place between BBB reductions and NNNa reductions. Jumping ahead, we note this is due their treatment of (in-)efficient adversaries: they require

BBB reductions to also work for inefficient adversaries $\mathcal{A}$, and so do we. In contrast, for NNNa reductions, Reingold et al. allow the reduction algorithm to fail for inefficient adversaries $\mathcal{A}$. As we can show, *all* notions of reducibility for inefficient adversaries, including NNN reductions, imply relativizing reductions, i.e., we can place relativizing reductions between NNN and NNNa reductions showing that, in fact, the notion is very liberal compared to notions of reductions that treat inefficient adversaries. In contrast, for efficient adversaries, relativizing reductions imply NNNa and (the equivalent) NBNa reductions and are incomparable to all stronger notions that treat efficient adversaries.

We now prove that relativizing reductions are implied by NNN reductions for inefficient adversaries. The proof is inspired by Reingold et al. [RTV04] who show that BBB reductions imply relativizing reductions.

Theorem 11. *If there is a $P \hookrightarrow Q$ NNN reduction, then there is a relativizing reduction from $\mathcal{P}$ to $\mathcal{Q}$.*

Proof. Assume there is an NNN reduction between two primitives $\mathcal{P}$ and $\mathcal{Q}$ and assume towards contradiction that there is an oracle Π such that $\mathcal{Q}$ exists relative to this oracle, but $\mathcal{P}$ does not. Let $f \in \mathcal{F}_{\mathcal{Q}}$ be an instantiation of $\mathcal{Q}$ that is efficiently computable by an algorithm that has oracle access to Π and such that f is secure against all efficient oracle machines $\mathcal{S}$, i.e., for all probabilistic polynomial-time machines $\mathcal{S}$, one has $(f, \mathcal{S}^{\Pi,f}) \notin \mathcal{R}_{\mathcal{Q}}$. By assumption of a $\mathcal{P} \hookrightarrow \mathcal{Q}$ NNN reduction, there exists a PPT oracle algorithm G for f, such that for all (possibly unbounded) adversaries $\mathcal{A}$ there is a PPT reduction algorithm $\mathcal{S}$ such that $(G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}}$ implies $(f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{Q}}$. Now, G^f is efficiently computable relative to the oracle Π, because G is PPT and f is efficiently computable relative to Π. Since $\mathcal{P}$ does not exist relative to Π, there is an efficient adversary $\mathcal{A}$ such that $(G^f, \mathcal{A}^{\Pi}) \in \mathcal{R}_{\mathcal{P}}$, i.e., by considering that the relations are defined over random variables, setting $\mathcal{A}' := \mathcal{A}^{\Pi}$ one also has $(G^f, \mathcal{A}'^f) \in \mathcal{R}_{\mathcal{P}}$. Thus, the NNN reduction gives an efficient reduction $\mathcal{S}$ such that $(f, \mathcal{S}^{\mathcal{A}',f}) \in \mathcal{R}_{\mathcal{Q}}$. As $\mathcal{S}$ is PPT and as f and $\mathcal{A}'$ are efficiently computable relative to oracle Π, one has that $\mathcal{S}^{\mathcal{A}',f}$ is efficiently computable relative to Π. Thus, f is not "$\mathcal{Q}$ secure" against all efficient oracle machines with oracle access to Π, yielding a contradiction. □

This proves that for inefficient adversaries, relativizing reductions are implied by NNN reductions, the most liberal notion of reductions for inefficient adversaries. Conversely, for efficient adversaries, relativizing reductions imply NNNa and NBNa reductions, but they are not implied by any of the stronger notions. We adapt the proof due to Reingold et al. for the following theorem.

Theorem 12. *If there is a relativizing reduction from $\mathcal{P}$ to $\mathcal{Q}$, then there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ NNNa reduction, and a $\mathcal{P} \hookrightarrow \mathcal{Q}$ NBNa reduction.*

Proof. Suppose we have a relativizing reduction from $\mathcal{P}$ to $\mathcal{Q}$ and consider any $f \in \mathcal{F}_Q$. Then, one of the following two cases holds. Either there is a PPT

algorithm $\mathcal{S}$ such that $(f, \mathcal{S}^f) \in \mathcal{R}_\mathcal{Q}$, i.e., breaking $\mathcal{Q}$. Define a machine G that computes an arbitrary (possibly insecure) implementation of $\mathcal{P}$. Then, we have an NNNa/NBNa reduction using G and $\mathcal{S}$. In the other case, such $\mathcal{S}$ does not exist and we know that $\mathcal{Q}$ exists relative to the oracle computing f. Hence, $\mathcal{P}$ exists relative to f as well via the relativizing reduction, implying that no efficient algorithm $\mathcal{A}^f$ breaks $\mathcal{P}$. Thus, the security premises of NNNa and NBNa reductions are not satisfied and therefore they exist. □

Theorem 13. *For $XYZ \in \{BBB, NBB, BBN, BNB, BNN, NBN, NNN\}$, there are primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ XYZa reduction for efficient adversaries, but no relativizing reduction.*

Proof. We show that BBBa reductions do not imply relativizing reductions; as BBBa reductions imply the "lower level" reductions, the other cases follow. We use the same approach as for Theorem 10.

Let $\mathcal{Q}$ be the primitive that contains the constant all-zero function f_z. We define the relation $\mathcal{R}_\mathcal{P}$ such that $\mathcal{P}$ is trivially secure against all *efficient* adversaries, namely, let $\mathcal{L}$ be an EXPTIME-complete language, then $(f_\text{z}, \mathcal{A})$ is in $\mathcal{R}_\mathcal{P}$ if $\mathcal{A}$ is a deterministic function and decides $\mathcal{L}$. As the complexity class P is strictly contained in EXPTIME, no efficient adversary can break $\mathcal{P}$. Let $\mathcal{Q}$ also be the primitive that contains the constant all-zero function f_z, but with a different relation, namely $\mathcal{R}_\mathcal{Q}$ is empty. In particular, no adversary can break $\mathcal{Q}$. Hence, there is a trivial $\mathcal{P} \hookrightarrow \mathcal{Q}$ BBBa reduction, because the premise of the implication

$$(G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}$$

is never satisfied for efficient adversaries and the implication is thus trivially true. In contrast, there is no relativizing reduction between the two primitives. That is, assume, we add an oracle that decides the EXPTIME-complete language $\mathcal{L}$, then relative to this oracle, there are suddenly efficient adversaries that break $\mathcal{P}$. However, as $\mathcal{R}_\mathcal{Q}$ is still empty, there cannot be a reduction $\mathcal{S}$ in this oracle world, giving us a contradiction. □

Reingold et al. note that BNNa reductions for efficient adversaries and relativizing reductions are often equivalent. In particular, they prove that if a primitive $\mathcal{Q}$ allows any oracle Π to be embedded into it, then a $\mathcal{P} \hookrightarrow \mathcal{Q}$ BNNa reduction implies a $\mathcal{P} \hookrightarrow \mathcal{Q}$ relativizing reduction. However, *efficient* primitives $\mathcal{Q}$ such as one-way functions (as opposed to random oracles, for example), are not known to satisfy this property. We discuss this issue in more detail in the following section about efficient primitives.

Efficient Primitives versus Inefficient Primitives

A reduction for *efficient* primitives is a reduction that only works if $f \in \mathcal{F}_\mathcal{Q}$ is efficiently implementable, i.e., in probabilistic polynomial time. If we make this distinction then, according to Figure 5.3, we unfold yet another dimension

(analogously to the case of efficient adversaries). As we discuss below, our results for non-efficient primitives hold in this "parallel universe" of efficient primitives as well, and between the two universes there are straightforward implications and separations (as in the case of efficient and inefficient adversaries).

Technically, one derives the efficient primitive version XYZp of an XYZ reduction by replacing all universal quantifiers over primitives f in $\mathcal{F}_\mathcal{Q}$ by universal quantifiers that are restricted to efficiently implementable f in $\mathcal{F}_\mathcal{Q}$. More concretely, we replace $\forall f \in \mathcal{F}_\mathcal{Q}$ by the term $\forall \mathsf{PPT} f \in \mathcal{F}_\mathcal{Q}$. For example, the notion of a BBBp reduction then reads as follows:

Definition 14 ($\mathcal{P} \hookrightarrow \mathcal{Q}$ BBBp or fully-black-box reduction for efficient primitives)**.** *There exists a* fully-black-box (*or* BBBp) reduction for efficient primitives *from* $\mathcal{P} = (\mathcal{F}_\mathcal{P}, \mathcal{R}_\mathcal{P})$ *to* $\mathcal{Q} = (\mathcal{F}_\mathcal{Q}, \mathcal{R}_\mathcal{Q})$ *if there exist probabilistic polynomial-time oracle algorithms* G *and* $\mathcal{S}$ *such that:*

Correctness. *For every polynomial-time computable function* $f \in \mathcal{F}_\mathcal{Q}$*, it holds that* $G^f \in \mathcal{F}_\mathcal{P}$*.*

Security. *For every polynomial-time computable function* $f \in \mathcal{F}_\mathcal{Q}$ *and every machine* $\mathcal{A}$*, if* $(G^f, \mathcal{A}) \in \mathcal{R}_\mathcal{P}$*, then* $(f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}$*, i.e.,*

$$\exists \mathsf{PPT} G \;\; \exists \mathsf{PPT} \mathcal{S} \;\; \forall \mathsf{PPT} f \in \mathcal{F}_\mathcal{Q} \;\; \forall \mathcal{A} \;\; ((G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}).$$

In the same manner, for any XYZ reduction, we can define the corresponding XYZp reduction. Similarly, one can transform all reduction types XYZa for efficient adversaries into reduction types XYZap for efficient adversaries *and efficient primitives*. For example, the notion of a BBBap reduction is as follows:

Definition 15 ($\mathcal{P} \hookrightarrow \mathcal{Q}$ BBBap Reduction)**.** *There exists a* fully-black-box (*or* BBBap) reduction for efficient adversaries and efficient primitives *from* $\mathcal{P} = (\mathcal{F}_\mathcal{P}, \mathcal{R}_\mathcal{P})$ *to* $\mathcal{Q} = (\mathcal{F}_\mathcal{Q}, \mathcal{R}_\mathcal{Q})$ *if there exist probabilistic polynomial-time oracle algorithms* G *and* $\mathcal{S}$ *such that:*

Correctness. *For every polynomial-time computable function* $f \in \mathcal{F}_\mathcal{Q}$*, it holds that* $G^f \in \mathcal{F}_\mathcal{P}$*.*

Security. *For every polynomial-time computable function* $f \in \mathcal{F}_\mathcal{Q}$ *and every probabilistic polynomial-time machine* $\mathcal{A}$*, if* $(G^f, \mathcal{A}) \in \mathcal{R}_\mathcal{P}$*, then* $(f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}$*, i.e.,*

$$\exists \mathsf{PPT} G \;\; \exists \mathsf{PPT} \mathcal{S} \;\; \forall \mathsf{PPT} f \in \mathcal{F}_\mathcal{Q} \;\; \forall \mathsf{PPT} \mathcal{A} \;\; ((G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}).$$

We will now review the separations proved so far in light of this new dimension. Basically, all relations that hold for XYZ reductions and XYZa reductions, also hold for XYZp and XYZap reductions, except for the relation to relativizing reductions as we will see below in Theorem 15. Firstly, the

proof of Theorem 9, showing equivalence of black-box access and non-black-box access to the efficient adversary, works for all classes of primitives and in particular for efficiently implementable ones. We conclude that there is an NNNap reduction from $\mathcal{P}$ to $\mathcal{Q}$ if and only if there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ NNNap reduction. For Theorem 10, separating reductions for the cases of efficient resp. inefficient adversaries, we observe that the primitive there, the constant all-zero function, is efficiently implementable. The proof hence also shows that for all $XYZ \in \{\mathrm{BBB}, \mathrm{BNB}, \mathrm{BBN}, \mathrm{NBB}, \mathrm{BNN}, \mathrm{NBN}, \mathrm{NNN}\}$, there are primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ XYZap-reduction, but no $\mathcal{P} \hookrightarrow \mathcal{Q}$ XYZp reduction. Note that this observation neither separates XYZ reductions from XYZp reductions, nor does it separate XYZa reductions from XYZap-reduction. These two classes of separations will be taken care of by Theorem 14.

Similarly to the constant all-zero-function case, all results that rely on random oracles carry through, as random oracles are efficiently computable. That is, Theorem 6, that uses a special class of weak one-way functions implemented as one-way oracles, still holds and shows that there are primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is an $\mathcal{P} \hookrightarrow \mathcal{Q}$ BNBp reduction, but no BBBp reduction. The same theorem establishes that for the same two primitives, there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ BNNp reduction, but no BBNp reduction, and there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ NNNp reduction, but no NBNp reduction. Moreover, as Theorem 7 also relies on random oracles only, we conclude that there are primitives $\mathcal{P}$ and $\mathcal{Q}$ such that for all $YZ \in \{\mathrm{BB}, \mathrm{BBa}, \mathrm{NN}, \mathrm{NNa}\}$, there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ NYZp-reduction, but there is neither a $\mathcal{P} \hookrightarrow \mathcal{Q}$ BYZp-reduction, nor a $\mathcal{P} \hookrightarrow \mathcal{Q}$ BBNp reduction, nor a $\mathcal{P} \hookrightarrow \mathcal{Q}$ BBNap reduction. Finally, Theorem 8 only uses the efficiently implementable constant all-zero function, and thus, the proof of Theorem 8 also establishes that for $X \in \{\mathrm{N}, \mathrm{B}\}$ there exist primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ XBNp reduction, a $\mathcal{P} \hookrightarrow \mathcal{Q}$ XBNap reduction, a BNNp reduction and BNNap reduction but such that there is no $\mathcal{P} \hookrightarrow \mathcal{Q}$ XBBp/XBBap/BNBp/BNBap reduction.

We now prove an analogue to Theorem 10, to separate reductions for arbitrary reductions from reductions for efficient primitives.

Theorem 14. *For each $XYZ \in \{BBB, BNB, BBN, NBB, BNN, NBN, NNN\}$, if* BPP $\neq$ EXPTIME*, then there are primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ $XYZp$ reduction, a $\mathcal{P} \hookrightarrow \mathcal{Q}$ $XYZap$ reduction, but neither a $\mathcal{P} \hookrightarrow \mathcal{Q}$ $XYZa$ reduction, nor a $\mathcal{P} \hookrightarrow \mathcal{Q}$ XYZ reduction.*

Proof. In the proof of Theorem 10 we made the primitive $\mathcal{Q}$ unbreakable. Thus, a reduction can only exist if one of the universal quantifiers $\forall \mathcal{A}$ or $\forall f$ quantifies over the empty set. We can use the same technique here, but this time we swap the role of the adversary and the role of the function f. Let $\mathcal{L}$ be an EXPTIME-complete language, and let f be the characteristic function for $\mathcal{L}$. Now, let $\mathcal{Q} = (\mathcal{F}_\mathcal{Q}, \mathcal{R}_\mathcal{Q})$ be defined through the singleton function set $\mathcal{F}_\mathcal{Q} = \{f\}$ and the empty relation $\mathcal{R}_\mathcal{Q} = \emptyset$. Let $\mathcal{P}$ be the primitive where $\mathcal{F}_\mathcal{P}$

only contains the constant all-zero function, denoted by f_z, and where all PPT adversaries $\mathcal{A}$ break this function, i.e., $\mathcal{R}_\mathcal{P} := \{(f_z, \mathcal{A}) : \mathcal{A} \text{ is PPT}\}$. Then, let G be the construction that ignores its oracle and constantly returns 0. Thus, for any $f \in \mathcal{F}_\mathcal{Q}$ the construction G^f implements the all-zero function.

Now, any universal PPT reduction algorithm $\mathcal{S}$ implements a $\mathcal{P} \hookrightarrow \mathcal{Q}$ XYZp reduction, a $\mathcal{P} \hookrightarrow \mathcal{Q}$ XYZap reduction, as the quantifier $\forall \text{PPT} f \in \mathcal{F}_\mathcal{Q}$ quantifies over the empty set, since BPP $\neq$ EXPTIME. On the other hand, no pair of a construction G and a reduction algorithm $\mathcal{S}$ (depending on f and $\mathcal{A}$) will implement a $\mathcal{P} \hookrightarrow \mathcal{Q}$ XYZa reduction or a $\mathcal{P} \hookrightarrow \mathcal{Q}$ XYZ reduction. This is because the premise of the implication

$$(G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_\mathcal{Q}$$

can be easily satisfied, as G^f implements the all-zero function and a PPT adversary $\mathcal{A}$ breaks it. On the other hand, the conclusion is impossible to achieve since $\mathcal{R}_\mathcal{Q}$ is empty.

As mentioned before, the original RTV paper required that $\mathcal{F}_\mathcal{P}$ contain at least one efficiently computable primitive. Thus, we have to slightly adapt our proof to work also in their setting. To do so, we add the constant all-zero function f_z to $\mathcal{F}_\mathcal{Q}$, i.e., $\mathcal{F}_\mathcal{Q} := \{f, f_z\}$ and define $\mathcal{R}_\mathcal{Q} := \mathcal{R}_\mathcal{P}$, i.e., for f, there is still no adversary that breaks f. Then, the same analysis applies. □

Note that the proof of Theorem 10 also shows the stronger statement that there are primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ XYZp reduction, a $\mathcal{P} \hookrightarrow \mathcal{Q}$ XYZ reduction, but neither a $\mathcal{P} \hookrightarrow \mathcal{Q}$ XYZa reduction, nor a $\mathcal{P} \hookrightarrow \mathcal{Q}$ XYZap reduction. This is, because all considered primitives in the proof of Theorem 10 are efficiently computable, namely the constant all-zero function.

As in the case of efficient adversaries, XYZp reductions are not strong enough to imply relativizing reductions.

Theorem 15. *There exists primitives $\mathcal{P}$ and $\mathcal{Q}$ such that there is an $\mathcal{P} \hookrightarrow \mathcal{Q}$ XYZp reduction, but no relativizing reduction.*

Proof. Consider the primitives $\mathcal{P}$ and $\mathcal{Q}$ designed in the previous proof. We saw that there is an $\mathcal{P} \hookrightarrow \mathcal{Q}$ XYZp reduction. We will show that there is no relativizing reduction from $\mathcal{P}$ to $\mathcal{Q}$ by proving that, relative to an oracle Π that implements the characteristic function of an EXPTIME-complete problem, the primitive $\mathcal{Q}$ exists, while the primitive $\mathcal{P}$ does not. First note that f is efficiently computable relative to an EXPTIME-complete oracle. Moreover, by definition of $\mathcal{R}_\mathcal{Q}$, there is no adversary $\mathcal{A}$ such that $(f, \mathcal{A})$ is in $\mathcal{R}_\mathcal{Q}$ and thus, f is efficiently computable relative to the EXPTIME-complete oracle and cannot be broken even by an adversary that is given access to the oracle. We showed that f exists relative to Π. On the other hand, $\mathcal{Q}$ does not exist relative to any oracle as the constant all-zero adversary $\mathcal{A}$ always breaks all implementations in $\mathcal{P}_\mathcal{Q}$, as $\mathcal{R}_\mathcal{P} = \{(f_z, \mathcal{A}) : \mathcal{A} \text{ is PPT}\}$. □

Theorem 12 shows that relativizing reductions imply NNNa reductions and thus, they also imply NNNap reductions. However, it is not clear whether they equally imply NNNp reductions. Theorem 12 considers a (possibly inefficient) implementation f of $\mathcal{F}_{\mathcal{Q}}$ as an oracle and argues that, relative to this oracle, the primitive $\mathcal{F}_{\mathcal{P}}$ exists, i.e., there is an efficient oracle algorithm G such that G^f implements $\mathcal{P}$ and cannot be broken by an adversary that has access to f. When switching the roles of the function f and the adversary $\mathcal{A}$, then this argument does not carry over, as the construction G does not get access to the adversary $\mathcal{A}$.

Note that the separation in Theorem 15 tells us that the use of efficient primitives is a possible way to bypass the important class of oracle separations with inefficient oracles. Nevertheless, it might also be interesting to explore the converse direction, i.e., whether relativizing reductions imply any type of XYZp reduction or not.

5.6 Parametrized Black-Box Reductions

Many reductions in cryptography commonly classified as "black box" technically do not fall in this class, as a black-box reduction algorithm must not have any information about the adversary beyond the input/output behavior, except for the sole guarantee that it breaks security with non-negligible probability. Strictly speaking, this excludes information such as running time, number of queries, or the actual success probability of a given adversary. This prompts the question of what the "natural" notion of a black-box reduction should be. Not surprisingly, the answer is a matter of taste, just like the question whether fully black box or semi black box is the "right" notion of a black-box reduction. As in the case of different notions of black-box reductions, we can nonetheless give a technically profound, and yet easy-to-use notion of *parametrized* black-box reductions (of any type). Before going into the details we first consider some motivating examples of dependencies on parameters of the adversary.

Parameter-Aware and Parameter-Dependent Reductions

Let us reduce unforgeability of a MAC scheme to its own unforgeability, i.e., the reduction algorithm $\mathcal{S}$ merely relays queries and answers between the unforgeability game and the adversary $\mathcal{A}$. Although the reduction algorithm is trivial, its running time depends on the adversary's behavior. Namely, the running time of the reduction is polynomial in the security parameter and the number of queries placed by the adversary. Hence the running time of $\mathcal{S}$ actually depends on $\mathcal{A}$, while the code of the strictly polynomial-time algorithm $\mathcal{S}$ should be universal for all $\mathcal{A}$, thus allowing only an a priori limited number of interactions with the adversary.

Another example is the well-known Goldreich–Levin hardcore-bit reduction [GL89], in the version attributed to Rackoff [Gol04]. Recall that the reduction algorithm receives some input $f(x)$ and has access to an adversary that predicts a hardcore bit with some non-trivial advantage $\epsilon(n)$. The reduction then uses amplification techniques by asking the adversary on many different input strings and thereby yields a preimage of $f(x)$ with non-negligible probability. As the amplification step heavily depends on $\epsilon(n)$, the reduction is not universal anymore; it changes with different values of $\epsilon(n)$. Moreover, the running time of the reduction depends on $1/\epsilon(n)$ or, more precisely, on some polynomial $p(n)$ with $1/p(n) > 1/\epsilon(n)$. Other than that, the reduction treats both the adversary and the primitive as black boxes.

The MAC example above shows that we sometimes want to allow the reduction, especially its running time, to depend on adversarial parameters such as its number of queries. In the second example the reduction needs (one of) the parameters as explicit input. We call the latter (black-box) reductions *parameter aware*, and the former *parameter dependent*. In fact, we make parameter-aware reductions strictly stronger by also making the reduction's running time depend on the input parameters.

The difference between the two notions is roughly that, in the parameter-aware case the reduction receives some auxiliary information about the adversary which may not be even known by the adversary itself (like the success probability), akin to non-uniform advice. In the parameter-dependent case the reduction only has sufficient time to run the adversary without violating prematurely fixed bounds on the running time. As another example one may consider knowledge extractors in proofs of knowledge [BG92] which run in expected polynomial time related to the prover's success probability to convince the verifier. This knowledge extractor can be oblivious about the actual success bound, and yet the running time depends on it. Remarkably, in order to prune the expected polynomial time by standard techniques to make the algorithm run in strict polynomial time, one needs to know the success probability and obtains a parameter-aware algorithm.

To simplify, we only define the two cases for BBB reductions and refrain from distinguishing between different parameters (for inputs resp. for running time dependency). We formalize this by having a function par mapping adversaries $\mathcal{A}$ to a function $\mathsf{par}_{\mathcal{A}}$ which, in turn, maps the security parameter to the desired parameters like the number of queries (to simplify, we include again the security parameter in these parameters). As such, when saying below that the reduction runs in polynomial time in $\mathsf{par}_{\mathcal{A}}$ it should be understood as saying that the running time of the reduction is polynomial in the output length of $\mathsf{par}_{\mathcal{A}}$ for the security parameter. This usually assumes some unitary encoding of the parameters. In the parameter-aware case we simply give the reduction the transformed input $\mathsf{par}_{\mathcal{A}}$ instead and write $\mathcal{S}(\mathsf{par}_{\mathcal{A}})$.

Definition 16 (Parameter-aware and parameter-dependent BBB reduction)**.**

There exists a parameter-aware BBB reduction *from a primitive* $\mathcal{P} = (\mathcal{F}_\mathcal{P}, \mathcal{R}_\mathcal{P})$ *to a primitive* $\mathcal{Q} = (\mathcal{F}_\mathcal{Q}, \mathcal{R}_\mathcal{Q})$ *with respect to* par*, if there exist probabilistic polynomial-time oracle machines* G *and* $\mathcal{S}$ *such that:*

Correctness. *For every* $f \in \mathcal{F}_\mathcal{Q}$*, it holds that* $G^f \in \mathcal{F}_\mathcal{P}$*.*

Security. *For every implementation* $f \in \mathcal{F}_\mathcal{Q}$ *and every machine* $\mathcal{A}$*, if* $(G^f, \mathcal{A}) \in \mathcal{R}_\mathcal{P}$*, then we have* $(f, \mathcal{S}^{\mathcal{A},f}(\mathsf{par}_\mathcal{A}) \in \mathcal{R}_\mathcal{Q})$*, i.e.,*

$$\exists \mathsf{PPT} G \;\; \exists \mathsf{PPT} \mathcal{S} \;\; \forall f \in \mathcal{F}_\mathcal{Q} \;\; \forall \mathcal{A} \;\; \Big((G^f, \mathcal{A}^f) \in \mathcal{R}_\mathcal{P} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}(\mathsf{par}_\mathcal{A})) \in \mathcal{R}_\mathcal{Q} \Big),$$

where algorithm $\mathcal{S}$ *runs in polynomial-time in its input and* $\mathsf{par}_\mathcal{A}$*. There is a* parameter-dependent BBB reduction *if the above holds if input* $\mathsf{par}_\mathcal{A}$ *is not given to* $\mathcal{S}$ *(but the running time may still depend on* $\mathsf{par}_\mathcal{A}$*).*

Although we state parametrized reductions in a rather general way, various standard choices for par are conceivable. In the light of the examples above, reasonable choices could be parameter functions that map descriptions of adversaries and the security parameter to the number of queries they make (par_q), to (the inverse of) their success probability (par_ϵ), or to their running time (par_t). This usually requires a refinement of the formalization of primitives to some form of games, e.g., to be able to specify the number of queries of the adversary. Consequently, this allows us to capture many known "black-box" reductions in the literature as BBB reductions with explicit parameters. At the same time, however, we get a very strict notion of a black-box reduction by letting $\mathsf{par}_\mathcal{A} = \bot$. In fact, this recovers Definition 9 and corresponds to our view on black-box reductions so far. We note that our definition leaves open whether the adversary actually knows a description of the function $\mathsf{par}_\mathcal{A}$ itself or not.

Finally, let us stress that we could also "parametrize" the other black-box objects, i.e., give the construction some hint about the black-box primitive such as its computation time, or hand the reduction further information about the primitive's parameters. We refrain from doing this formally as it is straightforward from the definition above and since, unlike in the case of adversarial parameters, we are not aware of "natural" examples for such cases.

Relationships

We note that parametrized black-box reductions and separations rely critically on the specific parameters. In particular, some of our separations consider reductions that are required to depend on, say, the success probability of the adversary, as in the case of the Goldreich–Levin hardcore bit (see Theorem 6). This separation does not carry over to the parametrized case. In contrast, separations for efficient/inefficient adversaries as well as the theorems on relativized reductions still apply.

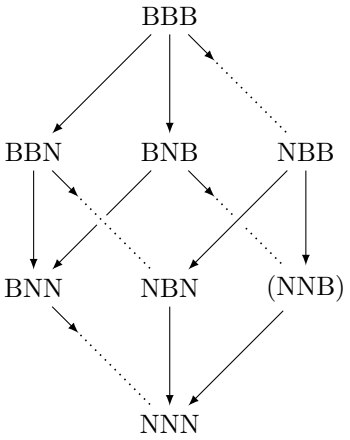


Figure 5.5: Parametrization makes shades of gray accessible (in the case of BYZ reductions). Parametrized counterparts of each type partly descend towards the corresponding NYZ reduction with full dependency on the construction.

More pictorially, one can imagine parametrized black-box reductions in light of Figure 5.3 as descending from the BYZ plane for black-box constructions towards the NYZ plane, where the construction can completely depend on the primitive—see Figure 5.5. The parameters and the distinction between awareness and dependency determines how far one descends. Analogously, parametrization for BBB reductions means that one descends from the top node BBB to BNB (also in the case of efficient adversaries). As such, it is clear that implications along edge paths remain valid, e.g., a parametrized NBN reduction still implies a NNN reduction.

The case of NBB reductions, however, shows that parametrization cannot fully bridge the gap to NNB reductions. As explained before, the latter type with quantification $\forall\mathcal{A}\exists\mathcal{S}\forall f\exists G$ does not seem to be meaningful, because the construction G would now depend on the adversary $\mathcal{A}$. Parametrization of NBB reductions (with quantification $\exists\mathcal{S}\forall f\exists G\forall\mathcal{A}$) still makes sense, though, because the dependency of $\mathcal{S}$ on the adversary is only through the running time or the input. Put differently, the parametrization allows for the "admissible non-black-boxness" for the NBB type of reduction.

Parameter Awareness and Parameter Dependency

Concerning the relationship of the two types of parametrized reductions we note that parameter awareness is not more powerful than parameter-dependent reductions, in the case that one can compute $\mathsf{par}_{\mathcal{A}}$ in time depending on the parameters. For sake of concreteness, we discuss this for the number of adversarial queries, showing that the two notions are equivalent in this case, except for the cases where the reduction cannot depend on the construction. This equivalence, of course, only makes sense for security relations in which there is a game between the adversary and the primitive in which the adversary actually poses queries. We call a relation $\mathcal{R}_{\mathcal{P}}$ *challenger based* if there is an efficient algorithm $C^{\mathcal{A},G^f}$ and a relation $\mathcal{R}^C_{\mathcal{P}}$ such that $(G^f,\mathcal{A})\in\mathcal{R}_{\mathcal{P}}$ iff

$(G^f, C^{\mathcal{A},G^f}) \in \mathcal{R}^C_{\mathcal{P}}$. In this case we denote by par_q the number of oracle calls of C to $\mathcal{A}$ (as a function of the security parameter). Note that this number now includes the first invocation of the adversary, but can essentially be thought of the number of queries $\mathcal{A}$ poses to the security game. We also assume that C simply makes additional queries if the adversary stops early, such that the number of oracle calls remains identical for all runs.

Proposition 13. *Let $\mathcal{P} = (\mathcal{F}_\mathcal{P}, \mathcal{R}_\mathcal{P})$ be a primitive with a challenger-based relation $\mathcal{R}_\mathcal{P}$. For any parameter-aware $\mathcal{P} \hookrightarrow \mathcal{Q}$ XYZ reduction (of type $XYZ \notin \{NNB, NBB\}$) with respect to par_q there is also parameter-dependent $\mathcal{P} \hookrightarrow \mathcal{Q}$ XYZ reduction of the same type XYZ. (This holds also for XYZa reductions.)*

Proof. Consider a parameter-aware reduction $\mathcal{S}$ of some admissible type, and let the C be the algorithm implementing the guaranteed challenger game for $\mathcal{R}_\mathcal{P}$. Then we build a parameter-dependent reduction $\mathcal{S}'$ as follows. Reduction $\mathcal{S}'$ first runs C with the adversary oracle once against G^f, simulating oracle G^f for C with the help of (possibly black-box access to) f. It counts the number of queries par_q the challenger (resp. the adversary) makes. Note that $\mathcal{S}'$ only needs $\mathcal{O}(\mathsf{par}_\mathsf{q})$ steps for simulating the black-box adversary oracle, even if given $\mathcal{A}$ as a black box. Also, $\mathcal{S}'$ requires for the simulation to be able to depend on the running time of G^f, which it indeed does for the admissible reduction types. Finally, once $\mathcal{S}'$ has received par_q it can simply invoke $\mathcal{S}$ for additional input par_q. □

It is now conceivable that, if one cannot compute $\mathsf{par}_\mathcal{A}$ (given that one can run in time depending on $\mathsf{par}_\mathcal{A}$), then parameter-aware reductions should be more powerful. This, however, presumes some notion of unpredictability which, in turn, stipulates some form of verifiability of correct outputs. As mentioned before, even the adversary itself may not be able to verify such outputs, e.g., think of its own success probability. This adds an additional layer of dependency of parameters, which is beyond our scope here.

5.7 Meta Reductions

In this last section, we define meta reductions within our augmented CAP framework and prove that if there exists a meta reduction from a $\mathcal{P} \hookrightarrow \mathcal{Q}$ reduction to the primitive $\mathcal{Q}$, then there is no reduction from $\mathcal{P}$ to $\mathcal{Q}$, provided that $\mathcal{Q}$ exists. More generally, if there exists a meta reduction from a $\mathcal{P} \hookrightarrow \mathcal{Q}$ reduction to a primitive $\mathcal{N}$, then there is no reduction from $\mathcal{P}$ to $\mathcal{Q}$, provided that $\mathcal{N}$ exists, i.e., there is an efficient implementation of $\mathcal{N}$ such that no efficient adversary breaks it. We now rephrase meta reductions for all notions introduced in the previous section. Below we usually put the statement in terms of reductions and meta reduction of the same CAP type. It is clear that a meta reduction of the XYZ type, ruling out reductions of the XYZ type,

also exclude all higher-level (i.e., "more black-box") reductions of type $\widehat{X}\widehat{Y}\widehat{Z} \geq XYZ$. Such higher-level reductions imply a reduction of the type XYZ and would thus contradict the impossibility result.

Meta Reductions for BYZ Reductions

Definition 17 (($\mathcal{P} \hookrightarrow \mathcal{Q}$ BBB) $\hookrightarrow \mathcal{N}$ fully-black-box meta reduction)**.** *For primitives $\mathcal{P}$, $\mathcal{Q}$ and $\mathcal{N}$, a probabilistic polynomial-time algorithm $\mathcal{M}$ is a* ($\mathcal{P} \hookrightarrow \mathcal{Q}$ BBB) $\hookrightarrow \mathcal{N}$ meta reduction *from a $\mathcal{P} \hookrightarrow \mathcal{Q}$ BBB reduction to $\mathcal{N}$, if the following holds for all $g \in \mathcal{F}_{\mathcal{N}}$:*

Reduction implies Insecurity. *If $\mathcal{P}$ BBB reduces to $\mathcal{Q}$ via a construction G and a reduction algorithm $\mathcal{S}$, then there is a* PPT *$\mathcal{M}$ such that one has $(g, \mathcal{M}^g) \in \mathcal{R}_{\mathcal{N}}$.*

$$\forall g \in \mathcal{F}_{\mathcal{N}}\ \forall \mathsf{PPT} G\ \forall \mathsf{PPT} \mathcal{S}\ \exists f \in \mathcal{F}_{\mathcal{Q}}\ \exists \mathcal{A}\ \exists \mathsf{PPT} \mathcal{M}$$

$$\Big[((G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{Q}}) \tag{5.3}$$

$$\Rightarrow (g, \mathcal{M}^g) \in \mathcal{R}_N\Big] \tag{5.4}$$

To construct a ($\mathcal{P} \hookrightarrow \mathcal{Q}$ BBB) $\hookrightarrow \mathcal{Q}$ meta reduction, one usually instantiates f via g and picks an (possibly inefficient) adversary $\mathcal{A}$ that breaks G^f. The efficient reduction algorithm $\mathcal{S}$ will turn $\mathcal{A}$ into a successful adversary against $f = g$. Thus, the meta reduction $\mathcal{M}$ aims at simulating $\mathcal{A}$ *efficiently* for $\mathcal{S}$. For this purpose, the meta reductions rewinds the reduction, e.g., to extract a signature from the reduction that simulates a signing oracle—the extracted signature can then be presented as a genuine fresh signature to a rewound version of the reduction $\mathcal{S}$. (We hide the details under the rug; indeed, the actual analysis is usually more complicated, see [FS10] for an example.) Consider the order of quantifiers in the above definition: the meta reduction may use non-black-box information about g and $\mathcal{S}$ such as the running time of $\mathcal{S}$ or its success probability. This definition is as liberal as possible on the meta reduction while preserving its ultimate goal: if a reduction $(G, \mathcal{S})$ exists and a meta reduction, then clearly, the primitive $\mathcal{Q}$ cannot exist.

Theorem 16. *If $\mathcal{N}$ exists and if there is a ($\mathcal{P} \hookrightarrow \mathcal{Q}$ BBB) $\hookrightarrow \mathcal{N}$ (a.k.a. fully-black-box) meta reduction, then there is no $\mathcal{P} \hookrightarrow \mathcal{Q}$ BBB reduction.*

Corollary 1. *If there is a ($\mathcal{P} \hookrightarrow \mathcal{Q}$ BBB) $\hookrightarrow \mathcal{Q}$ (a.k.a. fully-black-box) meta reduction, then a secure instantiation of $\mathcal{P}$ cannot be based on the existence of $\mathcal{Q}$ via a $\mathcal{P} \hookrightarrow \mathcal{Q}$ BBB reduction.*

Proof of Theorem 16. Assume that there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ reduction and a ($\mathcal{P} \hookrightarrow \mathcal{Q}$ BBB) $\hookrightarrow \mathcal{N}$ meta reduction. To derive a contradiction, we show that $\mathcal{N}$ cannot exist. Let $g \in \mathcal{F}_{\mathcal{N}}$ be arbitrary but fixed. As there is a $\mathcal{P} \hookrightarrow \mathcal{Q}$ BBB reduction,

let $(G, \mathcal{S})$ be a pair of a probabilistic polynomial-time construction G and a probabilistic polynomial-time reduction $\mathcal{S}$ such that for all f and $\mathcal{A}$, if $(G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}}$, then $(f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{Q}}$. Moreover, for $(G, \mathcal{S})$, let $\mathcal{M}$ be the meta reduction together with the corresponding f and $\mathcal{A}$ granted by the meta reduction property. As the condition $(G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{P}} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{Q}}$ is satisfied for all f and $\mathcal{A}$, we have that the meta reduction $\mathcal{M}$ breaks g, formally $(g, \mathcal{M}^g) \in \mathcal{R}_{\mathcal{N}}$. Note that $\mathcal{M}^g$ is efficiently computable if and only if g is efficiently computable. As the analysis holds for all $g \in \mathcal{F}_{\mathcal{N}}$, we derive that all efficiently computable instantiations of $\mathcal{N}$ can be broken by an efficient adversary. □

Note that all introduced notions for meta reductions easily translate into meta reductions for efficient adversaries by only quantifying over efficient $\mathcal{A}$. The remaining types of meta reductions follow analogously, we omit an explicit presentation here.

Examples of Meta Reductions

Meta reductions have been used for the first time (albeit not explicitly under this name) in the work of Boneh and Venkatesan [BV98] to study the relation between breaking RSA and factoring. Their result says that there is no (straight-line respectively algebraic) reduction from breaking RSA to factoring since such a reduction would immediately yield an efficient algorithm for factoring. Since they consider concrete problem instantiations, neither of the primitives is black box. In order to look at this result in terms of meta reductions, it is instructive to view the RSA oracle as the adversary and the reduction as a generic straight-line program-evaluation machine for the actual reduction's output that handles embedded RSA oracle calls within the program by forwarding them to the adversary. This makes the adversarial access black box and results in a NBN type meta reduction.

Bresson et al. [BMV08] discuss separations amongst so-called one-more problems where an adversary may query an oracle for solutions on n instances but needs to provide eventually $n + 1$ instance/solution pairs in order to be successful. The results indicate that solving such problems on n instances does not reduce to the case of solving the same problem on $n - 1$ instances (using fresh randomness). Again, the adversary and the primitive for the n-instance problem is treated in a black-box manner by the reduction. One may argue that the construction is black box as well since the problem can be constructed for an arbitrary number of instances solely by given access to oracles for generating and verifying one instance. Hence, this is an example for a BBB meta reduction. We note that all the reductions in this work come with certain restrictions though and *meta* reductions appear both as black-box and non-black-box—in the case of algebraic reductions—flavors.

The work of Haitner et al. [HRS09] uses meta reductions to show that witness hiding of certain proof systems cannot be based on either a specific hardness assumption or, separately, on any implementation of a primitive. These two variants precisely reflect the difference how the primitive is treated within the reduction and the construction. The latter case indicates a BBB meta reduction. For specific assumptions, the reduction may depend on the primitive and the authors call this a *weakly*-black-box reduction which shall not be confused with the weakly terminology of [RTV04]. In our framework this type of reduction classifies as a NBN meta reduction.

Fischlin and Schröder [FS10] prove the impossibility of basing blind signatures on a non-interactive standard assumption using a meta reduction. Here, the construction may be non black box, the adversary is treated as a black box, but the reduction is not restricted to black-box access to the primitive. This classifies as a NBN meta reduction.

Finally, the work by Pass [Pas11] presents a powerful framework to show that a certain type of argument system cannot be based on certain standard assumptions. By restating several interesting constructions as an argument system, it follows that these constructions cannot be based on standard assumptions either. More specifically, these constructions include the Schnorr identification scheme, the adaptive selective decommitment problem, one-more inversion assumptions, and unique blind signatures (generalizing the aforementioned result). Again, the underlying technique of this framework is a meta reduction. These results hold whenever the adversary in the reduction is treated as a black box but allows arbitrary constructions, which, in particular, may be non–black box. Since the reduction may depend on the standard assumptions as well, this type of meta reduction is considered a NBN meta reduction in our terminology.

Chapter 6

Conclusions

In this thesis, we addressed two issues which are strongly related to cryptographic reductions.

First, we provided a novel tool allowing us to compare cryptographic constructions that are provably secure in idealized models such as the random-oracle model. We defined two flavors of this tool (Chapter 4 and Chapter 3, respectively); one focusing on concrete security and the other one explicitly dealing with the underlying assumptions of the constructions. We applied this tool to the blockcipher-based compression functions due to Preneel, Govaerts, and Vandewalle [PGV93] and more advanced designs, as well as to two closely-related ElGamal-type encryption schemes. Our results showed that the compression functions can be categorized into two equivalent groups and the ElGamal-type encryption schemes are essentially interchangeable with respect to the requirements on the hash function they rely on.

In Chapter 5, we attacked the lack of both an adequate classification of cryptographic reductions and a concise language to communicate these classifications. We devised a framework that captures a broad range of cryptographic reductions and mapped out their relationships. In comparison with a previous approach by Reingold, Trevisan, and Vadhan [RTV04], we were able to include an overlooked type of reduction that is typically ruled out by so-called meta reductions. Moreover, we distinguished more rigorously between efficient and inefficient adversaries/primitives and could thus provide a more comprehensive view on the landscape of cryptographic reductions. Lastly, our framework exhibits a clean modular structure and that allows for easy extensibility.

References

[ABR01] Michel Abdalla, Mihir Bellare, and Phillip Rogaway. The oracle Diffie-Hellman assumptions and an analysis of DHIES. In David Naccache, editor, *Topics in Cryptology – CT-RSA 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 143–158, San Francisco, CA, USA, April 8–12, 2001. Springer, Berlin, Germany.

[ANPS07] Elena Andreeva, Gregory Neven, Bart Preneel, and Thomas Shrimpton. Seven-property-preserving iterated hashing: ROX. In Kaoru Kurosawa, editor, *Advances in Cryptology – ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 130–146, Kuching, Malaysia, December 2–6, 2007. Springer, Berlin, Germany.

[AFK+11] Frederik Armknecht, Ewan Fleischmann, Matthias Krause, Jooyoung Lee, Martijn Stam, and John P. Steinberger. The preimage security of double-block-length compression functions. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 233–251, Seoul, South Korea, December 4–8, 2011. Springer, Berlin, Germany.

[BBF13] Paul Baecher, Christina Brzuska, and Marc Fischlin. Notions of black-box reductions, revisited. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 296–315, Bengalore, India, December 1–5, 2013. Springer, Berlin, Germany.

[BFFS13] Paul Baecher, Pooya Farshim, Marc Fischlin, and Martijn Stam. Ideal-cipher (ir)reducibility for blockcipher-based hash functions. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes*

in Computer Science, pages 426–443, Athens, Greece, May 26–30, 2013. Springer, Berlin, Germany.

[BF11] Paul Baecher and Marc Fischlin. Random oracle reducibility. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 21–38, Santa Barbara, CA, USA, August 14–18, 2011. Springer, Berlin, Germany.

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *42nd Annual Symposium on Foundations of Computer Science*, pages 106–115, Las Vegas, Nevada, USA, October 14–17, 2001. IEEE Computer Society Press.

[BH13] Kfir Barhum and Thomas Holenstein. A cookbook for black-box separations and a recipe for UOWHFs. In Amit Sahai, editor, *TCC 2013: 10th Theory of Cryptography Conference*, volume 7785 of *Lecture Notes in Computer Science*, pages 662–679, Tokyo, Japan, March 3–6, 2013. Springer, Berlin, Germany.

[BBP04] Mihir Bellare, Alexandra Boldyreva, and Adriana Palacio. An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 171–188, Interlaken, Switzerland, May 2–6, 2004. Springer, Berlin, Germany.

[BG92] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In Ernest F. Brickell, editor, *Advances in Cryptology – CRYPTO'92*, volume 740 of *Lecture Notes in Computer Science*, pages 390–420, Santa Barbara, CA, USA, August 16–20, 1992. Springer, Berlin, Germany.

[BHK13] Mihir Bellare, Viet Tung Hoang, and Sriram Keelveedhi. Instantiating random oracles via UCEs. Cryptology ePrint Archive, Report 2013/424, 2013. http://eprint.iacr.org/2013/424.

[BK04] Mihir Bellare and Tadayoshi Kohno. Hash function balance and its impact on birthday attacks. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 401–418, Interlaken, Switzerland, May 2–6, 2004. Springer, Berlin, Germany.

[BP04] Mihir Bellare and Adriana Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 273–289,

Santa Barbara, CA, USA, August 15–19, 2004. Springer, Berlin, Germany.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press.

[BR94] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In Alfredo De Santis, editor, *Advances in Cryptology – EUROCRYPT'94*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111, Perugia, Italy, May 9–12, 1994. Springer, Berlin, Germany.

[BR96] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In Ueli M. Maurer, editor, *Advances in Cryptology – EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416, Saragossa, Spain, May 12–16, 1996. Springer, Berlin, Germany.

[BR06] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Berlin, Germany.

[BK09] Alex Biryukov and Dmitry Khovratovich. Related-key cryptanalysis of the full AES-192 and AES-256. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 1–18, Tokyo, Japan, December 6–10, 2009. Springer, Berlin, Germany.

[BKN09] Alex Biryukov, Dmitry Khovratovich, and Ivica Nikolic. Distinguisher and related-key attack on the full AES-256. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 231–249, Santa Barbara, CA, USA, August 16–20, 2009. Springer, Berlin, Germany.

[BP12] Nir Bitansky and Omer Paneth. From the impossibility of obfuscation to a new non-black-box simulation technique. In *53rd Annual Symposium on Foundations of Computer Science*, pages 223–232, New Brunswick, NJ, USA, October 20–23, 2012. IEEE Computer Society Press.

[BRS02] John Black, Phillip Rogaway, and Thomas Shrimpton. Black-box analysis of the block-cipher-based hash-function constructions from PGV. In Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 320–335, Santa Barbara, CA, USA, August 18–22, 2002. Springer, Berlin, Germany.

[BRSS10] John Black, Phillip Rogaway, Thomas Shrimpton, and Martijn Stam. An analysis of the blockcipher-based hash functions from PGV. *Journal of Cryptology*, 23(4):519–545, October 2010.

[BCFW09] Alexandra Boldyreva, David Cash, Marc Fischlin, and Bogdan Warinschi. Foundations of non-malleable hash and one-way functions. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 524–541, Tokyo, Japan, December 6–10, 2009. Springer, Berlin, Germany.

[BF05] Alexandra Boldyreva and Marc Fischlin. Analysis of random oracle instantiation scenarios for OAEP and other practical schemes. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 412–429, Santa Barbara, CA, USA, August 14–18, 2005. Springer, Berlin, Germany.

[BF06] Alexandra Boldyreva and Marc Fischlin. On the security of OAEP. In Xuejia Lai and Kefei Chen, editors, *Advances in Cryptology – ASIACRYPT 2006*, volume 4284 of *Lecture Notes in Computer Science*, pages 210–225, Shanghai, China, December 3–7, 2006. Springer, Berlin, Germany.

[BLS04] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. *Journal of Cryptology*, 17(4):297–319, September 2004.

[BV98] Dan Boneh and Ramarathnam Venkatesan. Breaking RSA may not be equivalent to factoring. In Kaisa Nyberg, editor, *Advances in Cryptology – EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 59–71, Espoo, Finland, May 31 – June 4, 1998. Springer, Berlin, Germany.

[BKSY11] Zvika Brakerski, Jonathan Katz, Gil Segev, and Arkady Yerukhimovich. Limits on the power of zero-knowledge proofs in cryptographic constructions. In Yuval Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 559–578, Providence, RI, USA, March 28–30, 2011. Springer, Berlin, Germany.

[BMV08] Emmanuel Bresson, Jean Monnerat, and Damien Vergnaud. Separation results on the "one-more" computational problems. In Tal Malkin, editor, *Topics in Cryptology – CT-RSA 2008*, volume 4964 of *Lecture Notes in Computer Science*, pages 71–87, San Francisco, CA, USA, April 7–11, 2008. Springer, Berlin, Germany.

[BM14] Christina Brzuska and Arno Mittelbach. Using indistinguishability obfuscation via UCEs. Cryptology ePrint Archive, Report 2014/381, 2014. http://eprint.iacr.org/2014/381.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th Annual ACM Symposium on Theory of Computing*, pages 209–218, Dallas, Texas, USA, May 23–26, 1998. ACM Press.

[CKS09] David Cash, Eike Kiltz, and Victor Shoup. The twin Diffie-Hellman problem and applications. *Journal of Cryptology*, 22(4):470–504, October 2009.

[Cor00] Jean-Sébastien Coron. On the exact security of full domain hash. In Mihir Bellare, editor, *Advances in Cryptology – CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 229–235, Santa Barbara, CA, USA, August 20–24, 2000. Springer, Berlin, Germany.

[Cor02] Jean-Sébastien Coron. Optimal security proofs for PSS and other signature schemes. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 272–287, Amsterdam, The Netherlands, April 28 – May 2, 2002. Springer, Berlin, Germany.

[CDMP05] Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. Merkle-Damgård revisited: How to construct a hash function. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 430–448, Santa Barbara, CA, USA, August 14–18, 2005. Springer, Berlin, Germany.

[CS03] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.

[DHT11] Yevgeniy Dodis, Iftach Haitner, and Aris Tentes. On the instantiability of hash-and-sign RSA signatures. Cryptology ePrint Archive, Report 2011/087, 2011. http://eprint.iacr.org/2011/087.

[DHT12] Yevgeniy Dodis, Iftach Haitner, and Aris Tentes. On the instantiability of hash-and-sign RSA signatures. In Ronald Cramer, editor, *TCC 2012: 9th Theory of Cryptography Conference*, volume 7194 of *Lecture Notes in Computer Science*, pages 112–132, Taormina, Sicily, Italy, March 19–21, 2012. Springer, Berlin, Germany.

[DOP05] Yevgeniy Dodis, Roberto Oliveira, and Krzysztof Pietrzak. On the generic insecurity of the full domain hash. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 449–466, Santa Barbara, CA, USA, August 14–18, 2005. Springer, Berlin, Germany.

[DRS09] Yevgeniy Dodis, Thomas Ristenpart, and Thomas Shrimpton. Salvaging Merkle-Damgård for practical applications. In Antoine Joux, editor, *Advances in Cryptology – EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 371–388, Cologne, Germany, April 26–30, 2009. Springer, Berlin, Germany.

[FLS+08] Niels Ferguson, Stefan Lucks, Bruce Schneier, Doug Whiting, Mihir Bellare, Tadayoshi Kohno, Jon Callas, and Jesse Walker. The skein hash function family, 2008.

[FF13] Marc Fischlin and Nils Fleischhacker. Limitations of the meta-reduction technique: The case of schnorr signatures. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 444–460, Athens, Greece, May 26–30, 2013. Springer, Berlin, Germany.

[FLR+10] Marc Fischlin, Anja Lehmann, Thomas Ristenpart, Thomas Shrimpton, Martijn Stam, and Stefano Tessaro. Random oracles with(out) programmability. In Masayuki Abe, editor, *Advances in Cryptology – ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 303–320, Singapore, December 5–9, 2010. Springer, Berlin, Germany.

[FS10] Marc Fischlin and Dominique Schröder. On the impossibility of three-move blind signature schemes. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 197–215, French Riviera, May 30 – June 3, 2010. Springer, Berlin, Germany.

[FGL09a] Ewan Fleischmann, Michael Gorski, and Stefan Lucks. On the security of tandem-DM. In Orr Dunkelman, editor, *Fast Software Encryption – FSE 2009*, volume 5665 of *Lecture Notes in Computer Science*, pages 84–103, Leuven, Belgium, February 22–25, 2009. Springer, Berlin, Germany.

[FGL09b] Ewan Fleischmann, Michael Gorski, and Stefan Lucks. Security of cyclic double block length hash functions. In Matthew G. Parker, editor, *12th IMA International Conference on Cryptography and Coding*, volume 5921 of *Lecture Notes in Computer Science*, pages 153–175, Cirencester, UK, December 15–17, 2009. Springer, Berlin, Germany.

[GKM+11] Praveen Gauravaram, Lars R. Knudsen, Krystian Matusiewicz, Florian Mendel, Christian Rechberger, Martin Schläfffer, and Soren S. Thomsen. Grøstl — a SHA-3 candidate, 2011.

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd Annual ACM Symposium on Theory of Computing*, pages 99–108, San Jose, California, USA, June 6–8, 2011. ACM Press.

[Gol04] Oded Goldreich. *Foundations of Cryptography: Basic Applications*, volume 2. Cambridge University Press, Cambridge, UK, 2004.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33:792–807, 1986.

[GIL+90] Oded Goldreich, Russell Impagliazzo, Leonid A. Levin, Ramarathnam Venkatesan, and David Zuckerman. Security preserving amplification of hardness. In *31st Annual Symposium on Foundations of Computer Science*, pages 318–326, St. Louis, Missouri, October 22–24, 1990. IEEE Computer Society Press.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *21st Annual ACM Symposium on Theory of Computing*, pages 25–32, Seattle, Washington, USA, May 15–17, 1989. ACM Press.

[GK03] Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the Fiat-Shamir paradigm. In *44th Annual Symposium on Foundations of Computer Science*, pages 102–115, Cambridge, Massachusetts, USA, October 11–14, 2003. IEEE Computer Society Press.

[GMR84] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A "paradoxical" solution to the signature problem (extended abstract). In *25th Annual Symposium on Foundations of Computer Science*, pages 441–448, Singer Island, Florida, October 24–26, 1984. IEEE Computer Society Press.

[GQ88] Louis C. Guillou and Jean-Jacques Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both trasmission and memory. In C. G. Günther, editor, *Advances in Cryptology – EUROCRYPT'88*, volume 330 of *Lecture Notes in Computer Science*, pages 123–128, Davos, Switzerland, May 25–27, 1988. Springer, Berlin, Germany.

[HT98] Satoshi Hada and Toshiaki Tanaka. On the existence of 3-round zero-knowledge protocols. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 408–423, Santa Barbara, CA, USA, August 23–27, 1998. Springer, Berlin, Germany.

[HHRS07] Iftach Haitner, Jonathan J. Hoch, Omer Reingold, and Gil Segev. Finding collisions in interactive protocols - a tight lower bound on the round complexity of statistically-hiding commitments. In *48th Annual Symposium on Foundations of Computer Science*, pages 669–679, Providence, USA, October 20–23, 2007. IEEE Computer Society Press.

[HH09] Iftach Haitner and Thomas Holenstein. On the (im)possibility of key dependent encryption. In Omer Reingold, editor, *TCC 2009: 6th Theory of Cryptography Conference*, volume 5444 of *Lecture Notes in Computer Science*, pages 202–219. Springer, Berlin, Germany, March 15–17, 2009.

[HRS09] Iftach Haitner, Alon Rosen, and Ronen Shaltiel. On the (im)possibility of Arthur-Merlin witness hiding protocols. In Omer Reingold, editor, *TCC 2009: 6th Theory of Cryptography Conference*, volume 5444 of *Lecture Notes in Computer Science*, pages 220–237. Springer, Berlin, Germany, March 15–17, 2009.

[HKN+05] Danny Harnik, Joe Kilian, Moni Naor, Omer Reingold, and Alon Rosen. On robust combiners for oblivious transfer and other primitives. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 96–113, Aarhus, Denmark, May 22–26, 2005. Springer, Berlin, Germany.

[HS65] Juris Hartmanis and Richard Edwin Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117:285–306, 1965.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[Her05] Amir Herzberg. On tolerant cryptographic constructions. In Alfred Menezes, editor, *Topics in Cryptology – CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 172–190, San Francisco, CA, USA, February 14–18, 2005. Springer, Berlin, Germany.

[Hir04] Shoichi Hirose. Provably secure double-block-length hash functions in a black-box model. In Choonsik Park and Seongtaek Chee, editors, *ICISC 04: 7th International Conference on Information Security and Cryptology*, volume 3506 of *Lecture Notes in Computer Science*, pages 330–342, Seoul, Korea, December 2–3, 2004. Springer, Berlin, Germany.

[Hir06] Shoichi Hirose. Some plausible constructions of double-block-length hash functions. In Matthew J. B. Robshaw, editor, *Fast Software Encryption – FSE 2006*, volume 4047 of *Lecture Notes in Computer Science*, pages 210–225, Graz, Austria, March 15–17, 2006. Springer, Berlin, Germany.

[HSW14] Susan Hohenberger, Amit Sahai, and Brent Waters. Replacing a random oracle: Full domain hash from indistinguishability obfuscation. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 201–220, Copenhagen, Denmark, May 11–15, 2014. Springer, Berlin, Germany.

[HR04] Chun-Yuan Hsiao and Leonid Reyzin. Finding collisions on a public road, or do secure hash functions need secret coins? In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 92–105, Santa Barbara, CA, USA, August 15–19, 2004. Springer, Berlin, Germany.

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *21st Annual ACM Symposium on Theory of Computing*, pages 44–61, Seattle, Washington, USA, May 15–17, 1989. ACM Press.

[KL08] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman and Hall/CRC, 2008.

[Kho10] Dmitry Khovratovich. *New Approaches to the Cryptanalysis of Symmetric Primitives*. PhD thesis, University of Luxembourg, 2010.

[KOS10] Eike Kiltz, Adam O'Neill, and Adam Smith. Instantiability of RSA-OAEP under chosen-plaintext attack. In Tal Rabin, editor,

Advances in Cryptology – CRYPTO 2010, volume 6223 of *Lecture Notes in Computer Science*, pages 295–313, Santa Barbara, CA, USA, August 15–19, 2010. Springer, Berlin, Germany.

[KP09] Eike Kiltz and Krzysztof Pietrzak. On the security of padding-based encryption schemes - or - why we cannot prove OAEP secure in the standard model. In Antoine Joux, editor, *Advances in Cryptology – EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 389–406, Cologne, Germany, April 26–30, 2009. Springer, Berlin, Germany.

[KM07a] Neal Koblitz and Alfred Menezes. Another look at generic groups. *Adv. in Math. of Comm.*, 1(1):13–28, 2007.

[KM07b] Hidenori Kuwakado and Masakatu Morii. Indifferentiability of single-block-length and rate-1 compression functions. *IEICE Transactions*, 90-A(10):2301–2308, 2007.

[LM92] Xuejia Lai and James L. Massey. Hash function based on block ciphers. In Rainer A. Rueppel, editor, *Advances in Cryptology – EUROCRYPT'92*, volume 658 of *Lecture Notes in Computer Science*, pages 55–70, Balatonfüred, Hungary, May 24–28, 1992. Springer, Berlin, Germany.

[Lam79] Leslie Lamport. Constructing digital signatures from a one-way function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, October 1979.

[LK11] Jooyoung Lee and Daesung Kwon. The security of abreast-dm in the ideal cipher model. *IEICE Transactions*, 94-A(1):104–109, 2011.

[LSS11] Jooyoung Lee, Martijn Stam, and John P. Steinberger. The collision security of tandem-DM in the ideal cipher model. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 561–577, Santa Barbara, CA, USA, August 14–18, 2011. Springer, Berlin, Germany.

[LTW05] Henry Lin, Luca Trevisan, and Hoeteck Wee. On hardness amplification of one-way functions. In Joe Kilian, editor, *TCC 2005: 2nd Theory of Cryptography Conference*, volume 3378 of *Lecture Notes in Computer Science*, pages 34–49, Cambridge, MA, USA, February 10–12, 2005. Springer, Berlin, Germany.

[LOZ12] Yehuda Lindell, Eran Omri, and Hila Zarosim. Completeness for symmetric two-party functionalities - revisited. In Xiaoyun

Wang and Kazue Sako, editors, *Advances in Cryptology – ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Computer Science*, pages 116–133, Beijing, China, December 2–6, 2012. Springer, Berlin, Germany.

[MP12] Mohammad Mahmoody and Rafael Pass. The curious case of non-interactive commitments - on the power of black-box vs. non-black-box use of primitives. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 701–718, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Berlin, Germany.

[MRH04] Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39, Cambridge, MA, USA, February 19–21, 2004. Springer, Berlin, Germany.

[NYWO09] Yusuke Naito, Kazuki Yoneyama, Lei Wang, and Kazuo Ohta. How to confirm cryptosystems security: The original Merkle-Damgård is still alive! In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 382–398, Tokyo, Japan, December 6–10, 2009. Springer, Berlin, Germany.

[Nie02] Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 111–126, Santa Barbara, CA, USA, August 18–22, 2002. Springer, Berlin, Germany.

[PV06] Pascal Paillier and Jorge L. Villar. Trading one-wayness against chosen-ciphertext security in factoring-based encryption. In Xuejia Lai and Kefei Chen, editors, *Advances in Cryptology – ASIACRYPT 2006*, volume 4284 of *Lecture Notes in Computer Science*, pages 252–266, Shanghai, China, December 3–7, 2006. Springer, Berlin, Germany.

[Pas11] Rafael Pass. Limits of provable security from standard assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd Annual ACM Symposium on Theory of Computing*, pages 109–118, San Jose, California, USA, June 6–8, 2011. ACM Press.

[PTV11] Rafael Pass, Wei-Lung Dustin Tseng, and Muthuramakrishnan Venkitasubramaniam. Towards non-black-box lower bounds in

cryptography. In Yuval Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 579–596, Providence, RI, USA, March 28–30, 2011. Springer, Berlin, Germany.

[Pie08] Krzysztof Pietrzak. Compression from collisions, or why CRHF combiners have a long output. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 413–432, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Berlin, Germany.

[PGV93] Bart Preneel, René Govaerts, and Joos Vandewalle. Hash functions based on block ciphers: A synthetic approach. In Douglas R. Stinson, editor, *Advances in Cryptology – CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 368–378, Santa Barbara, CA, USA, August 22–26, 1993. Springer, Berlin, Germany.

[RTV04] Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In Moni Naor, editor, *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 1–20, Cambridge, MA, USA, February 19–21, 2004. Springer, Berlin, Germany.

[RSS11] Thomas Ristenpart, Hovav Shacham, and Thomas Shrimpton. Careful with composition: Limitations of the indifferentiability framework. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 487–506, Tallinn, Estonia, May 15–19, 2011. Springer, Berlin, Germany.

[Rog06] Phillip Rogaway. Formalizing human ignorance. In Phong Q. Nguyen, editor, *Progress in Cryptology - VIETCRYPT 06: 1st International Conference on Cryptology in Vietnam*, volume 4341 of *Lecture Notes in Computer Science*, pages 211–228, Hanoi, Vietnam, September 25–28, 2006. Springer, Berlin, Germany.

[RS04] Phillip Rogaway and Thomas Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In Bimal K. Roy and Willi Meier, editors, *Fast Software Encryption – FSE 2004*, volume 3017 of *Lecture Notes in Computer Science*, pages 371–388, New Delhi, India, February 5–7, 2004. Springer, Berlin, Germany.

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd Annual ACM Symposium on Theory of*

Computing, pages 387–394, Baltimore, Maryland, USA, May 14–16, 1990. ACM Press.

[Sch91] Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[Seu12] Yannick Seurin. On the exact security of schnorr-type signatures in the random oracle model. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 554–571, Cambridge, UK, April 15–19, 2012. Springer, Berlin, Germany.

[Sim98] Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In Kaisa Nyberg, editor, *Advances in Cryptology – EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 334–345, Espoo, Finland, May 31 – June 4, 1998. Springer, Berlin, Germany.

[Sta08] Martijn Stam. Beyond uniformity: Better security/efficiency tradeoffs for compression functions. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 397–412, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Berlin, Germany.

[Sta09] Martijn Stam. Blockcipher-based hashing revisited. In Orr Dunkelman, editor, *Fast Software Encryption – FSE 2009*, volume 5665 of *Lecture Notes in Computer Science*, pages 67–83, Leuven, Belgium, February 22–25, 2009. Springer, Berlin, Germany.

[TY98] Yiannis Tsiounis and Moti Yung. On the security of ElGamal based encryption. In Hideki Imai and Yuliang Zheng, editors, *PKC'98: 1st International Workshop on Theory and Practice in Public Key Cryptography*, volume 1431 of *Lecture Notes in Computer Science*, pages 117–134, Pacifico Yokohama, Japan, February 5–6, 1998. Springer, Berlin, Germany.

[WPS+12] Lei Wei, Thomas Peyrin, Przemyslaw Sokolowski, San Ling, Josef Pieprzyk, and Huaxiong Wang. On the (in)security of IDEA in various hashing modes. In Anne Canteaut, editor, *Fast Software Encryption – FSE 2012*, volume 7549 of *Lecture Notes in Computer Science*, pages 163–179, Washington, DC, USA, March 19–21, 2012. Springer, Berlin, Germany.

[Yao82] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science*, pages 80–91, Chicago, Illinois, November 3–5, 1982. IEEE Computer Society Press.

[YMO08] Kazuki Yoneyama, Satoshi Miyagawa, and Kazuo Ohta. Leaky random oracle (extended abstract). In Joonsang Baek, Feng Bao, Kefei Chen, and Xuejia Lai, editors, *ProvSec 2008: 2nd International Conference on Provable Security*, volume 5324 of *Lecture Notes in Computer Science*, pages 226–240, Shanghai, China, October 31 – November 1, 2008. Springer, Berlin, Germany.