ON THE POSSIBILITY AND IMPOSSIBILITY OF TIGHT REDUCTIONS IN CRYPTOGRAPHY

CHRISTOPH BADER
On the Possibility and Impossibility of Tight Reductions in Cryptography

Christoph Bader
(Geburtsort: Bielefeld)

Dissertation zur Erlangung des Grades eines Doktor-Ingenieurs der Fakultät für Elektrotechnik und Informationstechnik an der Ruhr-Universität Bochum

June 2015
# Contents

1 Introduction .......................................................... 1
   1.1 Tight Reductions ............................................ 4
   1.2 Digital Signatures .......................................... 6
   1.3 Contribution ................................................ 9
      1.3.1 Sufficient Conditions for the Impossibility of Tight Reductions .................................. 9
      1.3.2 Tightly Secure Signatures in the Standard Model .................................................. 11
      1.3.3 Tightly Secure Signatures in the Random Oracle Model ......................................... 12

2 Preliminaries .......................................................... 15
   2.1 Notation ....................................................... 15
   2.2 Complexity Theoretical Foundations ...................... 16
      2.2.1 Model of Computation ................................ 16
      2.2.2 Complexity Assumptions ............................. 20
   2.3 Cryptographic Foundations .................................. 25
      2.3.1 The Security Parameter and Symmetric Key Size .................................................. 25
      2.3.2 Asymmetric Key Size and Exact Security ............... 27
      2.3.3 Basic Primitives ........................................ 28

3 Sufficient Conditions for the Impossibility of Tight Reductions in Cryptography ........................................... 33
   3.1 Warm-Up: Improved Bound for Digital Signatures ......... 36
      3.1.1 Simple reductions from non-interactive complexity assumptions to breaking UF-SMA-security ........................................ 38
      3.1.2 Lower Tightness Bound for $r$-simple reductions from any NICA to breaking UF-SMA-security ........................................ 41
## Contents

3.1.3 Interpretation ............................................. 46
3.2 Sufficient Conditions to Rule out Tight Reductions 47
  3.2.1 Definitions .................................. 48
  3.2.2 Main Theorem to Rule Out Tight Reductions .... 50
3.3 Applications .................................................. 55
3.4 Practical Implications ..................................... 59
3.5 Limitations: How to circumvent the bound ........ 60

4 Efficient Tightly Secure Signatures in the Standard Model 63
  4.1 Security Notions for Digital Signatures in the Multi
      User Setting ........................................ 66
  4.2 Generic Construction in the Standard Model ......... 67
     4.2.1 Description of the Scheme ................... 68
     4.2.2 Proof of Security .............................. 69
     4.2.3 Instantiation with Building Blocks from the
           Literature ....................................... 72

5 Practical Tightly Secure Signatures in the Random Or-
    acle Model .................................................. 75
  5.1 Intuition .................................................... 77
  5.2 Description of the Scheme ............................. 79
  5.3 Proof of Security .................................. 81
  5.4 Efficiency of our Scheme. ............................ 86

6 Conclusion ..................................................... 89

Bibliography ................................................... 91
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Game-based Security Game played between $C$ and $A$ (left-hand side) and Solver $R^A$, solving an instance $c$ of a computational problem, when running $A$ as a subroutine (right-hand side).</td>
</tr>
<tr>
<td>1.2</td>
<td>The bit-size of $N$ relative to the number of signatures the adversary is provided with and the security level, following ECRYPTII recommendations [ECR12].</td>
</tr>
<tr>
<td>2.1</td>
<td>Games $k$-$\text{Lin}<em>{\text{real}}^{\text{Gen}}(A, \kappa)$, $k$-$\text{Lin}</em>{\text{rand}}^{\text{Gen}}(A, \kappa)$ and $\text{DLOG}^{\text{Gen}}$.</td>
</tr>
<tr>
<td>2.2</td>
<td>Games $\text{XDH}<em>{c, \text{real}}^{\text{Gen}, \text{asym}}(A, \kappa)$ and $\text{XDH}</em>{c, \text{rand}}^{\text{Gen}, \text{asym}}(A, \kappa)$.</td>
</tr>
<tr>
<td>2.3</td>
<td>Game $\text{DP}_{\text{asym}}^{\text{Gen}}(A, \kappa)$.</td>
</tr>
<tr>
<td>2.4</td>
<td>Game $\text{NICA}_N(A, \kappa)$.</td>
</tr>
<tr>
<td>2.5</td>
<td>Collision Resistance Security Game for Hash-functions.</td>
</tr>
<tr>
<td>2.6</td>
<td>Witness Indistinguishability Security Game.</td>
</tr>
<tr>
<td>2.7</td>
<td>The EUF-CMA-security game for digital signatures.</td>
</tr>
<tr>
<td>2.8</td>
<td>MU-IND-CPA-C-security Game. The attacker has access to an encryption oracle $\mathcal{O}.\text{Encrypt}$ which may be queried only once and a corrupt oracle $\mathcal{O}.\text{Corrupt}$.</td>
</tr>
<tr>
<td>3.1</td>
<td>The UF-SMA-security Game with attacker $A = (A_1, A_2)$.</td>
</tr>
<tr>
<td>3.2</td>
<td>Algorithm $r$-$R^A$ that solves a non-interactive complexity assumption according to Definition 2.8 constructed from a $r$-simple reduction $r$-$R = (R_0, (R_{l,1}, R_{l,2}, R_{l,3})_{l \in [r]}, R_3)$ and an attacker $A = (A_1, A_2)$.</td>
</tr>
</tbody>
</table>
### List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3</td>
<td>The UF-SSA-security Game with attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$</td>
<td>49</td>
</tr>
<tr>
<td>3.4</td>
<td>Algorithm $\Gamma^A$ that solves a non-interactive complexity assumption according to Definition 2.8 constructed from a $r$-simple reduction $r-\Gamma = (\Gamma_0,$ $(\Gamma_{t,1}, \Gamma_{t,2}, \Gamma_{t,3})_{t \in [r]}, \Gamma_3)$ and an attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$</td>
<td>51</td>
</tr>
<tr>
<td>3.5</td>
<td>Theoretically sound parameter size for RSA when used in TLS relative to the number of servers, when parameters are selected according to ECRYPT II recommendations (cf. Section 2.3.1)</td>
<td>60</td>
</tr>
<tr>
<td>4.1</td>
<td>Security Games for Digital Signatures</td>
<td>66</td>
</tr>
<tr>
<td>4.2</td>
<td>Efficiency comparison of DLIN-based MU-EUF-CMA-C-signature schemes when derived from MU-EUF-CMA-secure signatures and DLIN-based Groth-Sahai proofs (NIPS-GS) in symmetric bilinear groups (cf. Section 2.2.2) using our Compiler</td>
<td>73</td>
</tr>
<tr>
<td>5.1</td>
<td>Comparison of our random oracle based scheme and random oracle signature schemes from the literature</td>
<td>86</td>
</tr>
<tr>
<td>5.2</td>
<td>Efficiency Evaluation of our Signature Scheme</td>
<td>87</td>
</tr>
</tbody>
</table>
1 Introduction

Security of a cryptographic scheme is often captured through a security experiment that is played between a challenger $C$ and an adversary $A$, cf. left-hand side of Figure 1.1. When the game finishes, $C$ evaluates a boolean predicate, and, if it evaluates to true, we say that $A$ breaks the scheme with respect to the security notion that is defined by the interaction between $A$ and $C$ and the predicate.

Figure 1.1: Game-based Security Game played between $C$ and $A$ (left-hand side) and Solver $R^A$, solving an instance $c$ of a computational problem, when running $A$ as a subroutine (right-hand side).

When a new cryptographic scheme is proposed, nowadays the construction comes along with a proof of security. Most commonly, the proof is relative to a complexity assumption which states that it is computationally hard to solve certain computational problem. That is, the proof describes an efficient algorithm $R$, the reduction, that turns any successful attacker $A$ against the scheme with respect to the considered security notion into another algorithm $R^A$, the solver, that simulates $C$ to $A$ and that breaks the computational problem if $A$ successfully breaks security of the scheme, cf. Figure 1.1. Now, if no algorithm is known that efficiently solves the problem it is conjec-
tured that an efficient attacker against the scheme cannot exist, since otherwise the solver, given access to the attacker, solves the problem efficiently (i.e., breaks the complexity assumption). In other words, the reduction reduces breaking the complexity assumption to breaking the cryptographic scheme. This concept of provable security dates back to [GM84] and is a fundamental building block of modern cryptography [Gol01, Gol04, KL07].

The Loss of a Reduction. Since the solver $R^A$ needs to simulate $C$ for $A$ it has usually a larger running time compared to $A$. Moreover, it may be the case that the simulation fails to be perfect. In this case, it may happen that $R^A$ does not solve the computational problem, even if $A$ would have broken the scheme. Thus, the solver’s probability of success may be lower than that of the attacker.

Let us follow [BR06, Gal04] and define the work factor of an algorithm as its running time over its probability of success. We measure the quality of a reduction $R$ in terms of the work factor $w_S := \frac{t_S}{\epsilon_S}$ of the solver relative to the work factor $w_A := \frac{t_A}{\epsilon_A}$ of the attacker. We call this ratio the loss of the reduction $R$ [BR96]. Let us elaborate on this. To this end, let us assume that reduction $R$ reduces breaking certain computational problem $N$ to breaking a cryptographic scheme. Let us assume, for reasons of simplicity, that $t_S = \mathcal{O}(t_A)$ and that $N$ cannot be solved with work factor $\leq 2^\kappa$. That is, the probability of breaking $N$ in time $t_S$ is upper bounded by $\frac{t_S}{2^\kappa}$. What does this mean for the security of the considered cryptographic scheme? Let us illustrate this by giving a concrete example. To this end, let us (informally) define the number of bits of security of a scheme as the logarithm of the work factor that is required to break the scheme. This number can easily be estimated for symmetric cryptographic schemes, assuming them to behave optimal. In this case, a $\kappa$-bit cipher provides $\kappa$ bits of security, cf. Section 2.3.

According to [ECR12], a 1248 bit RSA modulus $N$ (heuristically) refers to 80 bits of security. Thus, a solver running in time $t$ has success probability at most $\frac{t}{2^{80}}$, or work factor $2^{80}$.

\footnote{We will define the running time of a Turing machine in Section 2.2.1 as the number of operations that are carried out by an algorithm. The}
Now, let us consider the RSA Full Domain Hash signature scheme that was introduced in [BR93b] and that is proven to be secure under the RSA assumption relative to \( N \). It is well known that a loss linear in the number, say \( \mu \), of signatures the adversary gets to see is unavoidable for RSA-FDH [Cor02] (at least if the public exponent is large [KK12, KKM12]). Now let us follow [BR96] and assume that for the number \( \mu \) of signatures the adversary gets to see it holds that \( \mu = 2^{30} \). Then we have:

\[
2^{30} \geq \frac{w_S}{w_A} \Rightarrow w_A \geq \frac{w_S}{2^{30}} = 2^{50}
\]

Stated otherwise, it holds that \( \epsilon_A \leq \frac{t_A}{2^{50}} \). That is, the bound on the adversary’s work factor is not \( 2^{80} \) but \( 2^{50} \) which roughly refers to breaking an ideal cipher with 50 bit key per brute force. We stress that we do not know if there is an attack strategy that really supports a work factor of \( 2^{50} \). However, from a theoretical point of view we can not rule out such attack.

To overcome this issue we choose an RSA-modulus \( N \) of size roughly 2400 bits. According to [ECR12], a solver breaking RSA with respect to such \( N \) needs a work factor of roughly \( 2^{110} \). Now, still assuming a loss of \( 2^{30} \) we can bound the work factor of an attacker by \( 2^{80} \) with the same reasoning as above and breaking the cryptographic scheme is as hard as breaking the security of an ideal cipher with 80 bit key per brute force. Figure 1.2 shows the size of the modulus \( N \) for RSA-FDH, when determined as discussed above, relative to the number of signatures the adversary is provided with and the required security level.

We stress that, in our example, to compensate for the loss, the system parameters size roughly doubled and, since an exponentiation \( \mod N \) requires roughly \( \log_2(N)^3 \) operations, sign- and verify-operations are 8 times slower in this case. Moreover, more space is needed to store public keys and longer signatures are to be exchanged.

We stress further that our reasoning relied on the fact that we assumed \( \mu = 2^{30} \). If \( \mu > 2^{30} \), then we can not rule out attacks that require work factor less than \( 2^{80} \), even if we choose \( N \) to be of size 2400 bits. On the other hand, if \( \mu < 2^{30} \), we can rule...
1 Introduction

<table>
<thead>
<tr>
<th>Security level</th>
<th>$\mu$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>80</td>
<td>1248</td>
</tr>
<tr>
<td>112</td>
<td>2432</td>
</tr>
</tbody>
</table>

Figure 1.2: The bit-size of $N$ relative to the number of signatures the adversary is provided with and the security level, following ECRYPTII recommendations [ECR12].

out such attacks. However, in this case, we waste computational resources and space.

1.1 Tight Reductions

In light of the above discussion on its practical importance, let us elaborate further on the loss of a reduction. In general, the loss is a function $\ell(\kappa, \pi)$ of the security parameter $\kappa$ and the deployment parameters $\pi$, for example the number of sign-queries the adversary may issue during the EUF-CMA-security experiment for digital signatures, as above. However, how do we choose parameters theoretically sound, if the loss depends on the deployment parameters? We need to estimate them \textit{a priori}, which is at least unsatisfactory, and sometimes may be impossible. Therefore, we ideally have $w_I = \mathcal{O}(w_A)$. In this case, $\ell$ is a constant and we call the reduction \textit{tight} and say that the scheme has \textit{tight security}. If a reduction of a cryptographic scheme is tight there is no need to select larger parameters than required. We can determine precisely the parameter-size we need in practice. Thus, tight reductions are a desirable goal [ADK+13, LJYP14, BKP14, BKKP15, HKS15] with practical applications.

Tight reductions are probably even more interesting to consider from a theoretical point of view, since tight reductions require rigorous design approaches and proof techniques that are fruitful for the community. For example, the dual system encryption paradigm was developed in [Wat09] with the intention
in mind to construct a provably secure hierarchical identity based encryption scheme that does not suffer from an exponential (in the number of levels) loss. The technique was used and improved in many subsequent works, e.g., LW10 LW11 CW13 BKP14 in the area of identity based encryption. The dual form signature paradigm that was introduced in GLOW12 also follows this approach.

For many cryptographic primitives there exist implementations that come along with an (almost) tight reduction in the standard model or the random oracle model, e.g., for digital signatures in the single user setting BR96 Cor02 KW03 Ber08 Sch11 KK12 HJ12, for public key encryption in the multi user setting BBM00 HJ12, for (hierarchical) identity based encryption CW13 BKP14 and for pseudorandom functions NR97 LW09.

However, though this may give insight in how to preferably construct cryptographic schemes, little is known about general properties of a cryptographic primitive or scheme that rule out tight reductions. There are results that rule out tight reductions for specific primitives like unique or re-randomizable signatures in the single user setting MRV99 Cor02 HJK12 KK12, checkable prefix encryption LW14 and message authentication codes in a strong security model BGM04. In addition, there are some works that rule out tight reductions for specific schemes like the Schnorr signature scheme Sch89 PV05 GBL08 Seu12 FJS14 and HMAC BCK96 GPR14.

All these negative results are very valuable for the community as they help to understand theory of cryptographic schemes and their proof techniques. Moreover, the community benefits from the rigorous analysis and new concepts that are used in the above works. In particular, the rewinding-technique (originating from GMR89) that was first used by Coron Cor02 to rule out tight reductions for unique signatures was used in a similar way in, e.g., HJK12 KK12 LW14.

However, it is not clear if the techniques used in the literature to rule out tight reductions for specific primitives or schemes apply to further primitives or schemes as well. For example, we discuss in Chapter 3 why it seems that techniques developed in Cor02 HJK12 KK12 have limited applications beyond dig-
1 Introduction

Digital signatures.

Research Questions. In contrast to the many works on the impossibility of tight reductions for specific schemes, little is known about properties of a cryptographic scheme in general that rule out a tight reduction. In the light of the above discussion on the practical importance of tight reductions, we wonder whether there exist general conditions that rule out tight reductions for (an implementation of) a cryptographic primitive?

1.2 Digital Signatures

Informally, digital signatures provide a means to guarantee integrity and authenticity of a message. The standard security notion in the single user setting today (EUF-CMA-security, cf. Definition 2.11) goes back to Goldwasser et al. [GMR88]. Roughly speaking, here, the adversary is provided with a public key $vk$ and may obtain signatures for messages of its choosing. Finally, the adversary is considered successful if it manages to output a message $m^*$ that was not signed before and a purported signature $\sigma^*$ such that $\sigma^*$ is a valid signature over $m^*$ with respect to $vk$.

In addition to formalizing security for digital signatures, Goldwasser et al. did also propose a signature scheme and proved it to be secure with respect to the newly defined security notion under the assumption that it is hard to factorize integers [GMR88]. After this breakthrough in the area of digital signatures many schemes were proposed, e.g., [Sch89, BR93b, BLS01, KW03, Wat05, HW09, Wat09, HJ12, CW13, BKP14]. All these schemes are provably secure from standard assumptions either in the standard model, like [Wat05, HW09, Wat09, HJ12, CW13, BKP14] or in the random oracle model [Sch89, BR93b, BLS01, KW03, KK12].

A common deployment parameter for signature schemes is the number $\mu$ of signatures the adversary gets to see in the above described experiment. For some of the above schemes [BR93b, BLS01, Wat05, HW09], we know by the results of [Cor02, HJK12, KK12] that there can not be a tight reduction with respect to $\mu$. 
Additionally, for the Schnorr signature scheme [Sch89] a similar result was proven in [PV05, GBL08, Seu12, FJS14].

On the other hand, the constructions from [KW03, HJ12, CW13, BKP14] and the probabilistic scheme from [BR93b] come along with a tight reduction (while the original proof RSA-PSS was non-tight, Coron [Cor02] showed the scheme to be tightly-secure). There are even more schemes that are provably tightly secure in the standard model [Sch11]. However, for these schemes the tight security reduction comes at the cost of a stronger security assumption.

**Digital Signature Schemes in the Multi User Setting.** Though EUF-CMA-security is the standard security notion for digital signatures today, it considers only an ideal world. Namely, the attacker is provided only with a single public key here. But when digital signature schemes are deployed in practice, they are often used by many users, each having its own public key. Due to implementation bugs, e.g., [HDWH12, MMS13, KM14], an adversary may be able to obtain secret keys of some users. A more realistic model would thus provide the adversary with many public keys and, allow the adversary:

1) to adaptively obtain signatures for messages of its choosing with respect to public keys of its choosing, and

2) to adaptively corrupt keys of its choice.

Now, security should still hold with respect to uncorrupted keys. That is, the attacker is considered successful in breaking the security of the system if it manages to produce a signature \( \sigma^* \) for a message \( m^* \) that was not signed before with respect to the target public key and that verifies under an uncorrupted public key (the target public key). We call this notion *multi-user existential unforgeability under chosen message attacks with corruptions* (MU-EUF-CMA-C-security). The reason why EUF-CMA still is considered the standard security notion is the fact that it is well known [MS04] that standard EUF-CMA security implies security in this more realistic multi user setting with corruptions. However, the generic reduction loses a factor of \( \ell \) where \( \ell \) is the
1 Introduction

number of users that use the signature scheme and it is not clear if this loss can be avoided in general.

Thus, while tightness in the single user setting is mostly considered with respect to the number \( \mu \) of sign queries issued by the attacker, in the multi user setting tightness is additionally considered relative to the number \( \ell \) of public keys the adversary has access to and that it may corrupt. Hence, for digital signatures in the multi user setting there are two dimensions to consider tightness in and it is not clear if the above schemes from [BR93b, KW03, HJ12, CW13, BKP14] allow for a security proof that is not only tight with respect to \( \mu \) but also tight with respect to \( \ell \).

Applications of MU-EUF-CMA-C-security. We remark that common security models for authenticated key exchange (AKE) or confidential channel establishment (ACCE), like e.g. [BR93a, CK01, JKSS12], allow the adversary to corrupt long-term secret keys which often are secret keys of a signature scheme, e.g., in ephemeral Diffie-Hellman Ciphersuites of the TLS-Handshake [DR08], the AKE protocol from [BHJ+15] or when compilers lift a passively secure protocol to meet stronger security notions [BCK98, KY03, JKSS10, LSY+14]. Therefore, the strong notion of MU-EUF-CMA-C security notion is implicitly widely used in practice. However, the security proofs for most schemes apply the “polynomial equivalence between EUF-CMA and MU-EUF-CMA-C security” argument which incurs a loss of \( \ell \) for the reduction and makes it hard to estimate the parameter size when a system and thus will require larger parameters when the scheme is implemented in practice. Therefore a signature scheme that comes along with a tight MU-EUF-CMA-C security reduction is a desirable goal with practical applications.

Research Questions. As mentioned above, it is well known that the generic reduction from MU-EUF-CMA-C-security to standard EUF-CMA-security loses a factor of \( \ell \) (the number of public keys the adversary is provided with). Now, given the fact that MU-EUF-CMA-C-security is the “real world”-security notion for digital signatures, we wonder whether the generic loss inherent
1.3 Contribution

Our contribution is threefold.

1.3.1 Sufficient Conditions for the Impossibility of Tight Reductions.

In [BJLS15], we analyze which properties are sufficient to rule out tight reductions for a cryptographic scheme. To this end, we first establish a bound for digital signatures in the single-user setting, refining the results from [Cor02, HJK12, KK12]. Informally, we consider the following security experiment for digital signatures:

1. The attacker is called on input a verification key $vk$ and $n$ distinct messages $m_1, \ldots, m_n$. It outputs an index $j \in [n]$.
2. In response, it non-adaptively obtains signatures $\sigma_i$ for all messages except for $m_j$.
3. Finally, the adversary outputs $\sigma_j$ and is considered successful if this is a valid signature over $m_j$ with respect to $vk$.

Informally, we show in this setting that a tight reduction (with respect to the deployment parameter $n$) can not exist if the signature scheme is efficiently re-randomizable. Since the above security notion, that we call unforgeability under static message attacks (UF-SMA-security), is strictly weaker than standard EUF-CMA-security (an explicit proof can be found in [BJLS15]), our result immediately applies to EUF-CMA-security, as well.

Next, we observe that for the technique to work we do not need the full structural properties of a signature scheme. We extract the required properties and describe a generalized security experiment with abstract computable relations that satisfy these properties. The abstract experiment roughly works as follows:
1 Introduction

1. The attacker is called on input a description of a relation $R(x, y)$ and $n$ statements $y_1, \ldots, y_n$. It outputs an index $j \in [n]$.

2. In response, it obtains witnesses $x_i$ for all statements except for $y_j$. That is, it holds $R(x_i, y_i)$ for all obtained witnesses $x_i, i \neq j$.

3. Finally, the adversary outputs $x_j$ and is considered successful if $x_j$ is a valid witness for $y_j$, i.e., if $R(x_j, y_j)$.

On a high level, we show that if the considered relation is efficiently verifiable and re-randomizable, then (under some restrictions) the loss of an efficient reduction from any hard non-interactive complexity assumption to breaking the above experiment, is at least $n$. As mentioned before, the result immediately applies to stronger security notions, as well. Security notions that are encompassed by our result are, e.g, security notions for digital signatures and public key encryption in the multi-user setting with corruptions, or non-interactive key exchange.

Thus, we are the first find out that a tight reduction from breaking the security of a non-interactive complexity assumption to breaking the security of a scheme is impossible in general, whenever

- the security game induces an efficiently computable relation such that witnesses are efficiently re-randomizable and

- for which the adversary may obtain witnesses throughout the security game and

- where it is successful in breaking security when it is able to compute an “uncorrupted” witness.

Thus, our result can be seen as a tool. Whenever the above properties are satisfied, we can immediately rule out tight reductions.

Next to discussing practical implications of our result, we point to several techniques that allow circumventing our impossibility result. This gives insight in how to construct cryptographic schemes when a tight reduction is desired.
1.3 Contribution

1.3.2 Tightly Secure Signatures in the Standard Model

Second, we consider standard-model digital signatures in the multi-user setting with corruptions. Our research in this direction was published at TCC 2015 [BHJ+15]. We develop and formally define the practically deployed notion of MU-EUF-CMA-C-security for digital signatures. Applying techniques pointed to when we discussed how to circumvent the above mentioned impossibility result, we propose a tightness-preserving compiler that generically combines a signature scheme that is secure in the multi-user setting without corruptions and a suitable witness indistinguishable proof of knowledge into an MU-EUF-CMA-C-secure scheme. The security loss is roughly the same, as for the underlying signature scheme that is provably secure without corruptions. Informally, it circumvents the above mentioned impossibility result as follows.

We combine the Naor-Yung paradigm [NY90] with the dual form signature approach [GLOW12]. A bit more detailed: A public key of our new signature scheme \( \text{SIG} \) consists of two public keys of the underlying signature scheme \( \text{SIG}' \) that is secure without corruptions. However, the secret key consists of only one of both corresponding secret keys. A \( \text{SIG} \)-signature is a proof, using the proof system, that the signer actually knows a valid \( \text{SIG}' \)-signature with respect to at least one of the two \( \text{SIG}' \)-public keys.

Now, our result on the impossibility of tight reductions does not apply since there are two types of secret keys per public key and it is in general not possible to efficiently re-randomize over all “valid” secret keys. Moreover, there are actually three “congruence classes” of signatures which will be accepted by the verification algorithm of our scheme. It will be possible to efficiently re-randomize signatures of one type. Re-randomization over all valid signatures will be computationally hard.

The scheme was used to construct the first tightly-secure authenticated key exchange (AKE) protocol [BHJ+15], a cryptographic primitive that is widely used in practice [BR93a, BCK98, JKSS10, JKSS12, KPW13, BJS15]. Prior to our work, known AKE constructions usually had a loss that is at least linear in
1 Introduction

the number of users and/or sessions per user.

On the negative side, when the scheme is instantiated with known tightly MU-EUF-CMA-secure signature schemes, e.g., the scheme from [HJ12], and suitable proof systems, e.g., [GS08], signatures are of size linear in the security parameter. Though this is asymptotically efficient it is far away from being practical.

1.3.3 Tightly Secure Signatures in the Random Oracle Model

Third, we construct a practical signature scheme that tightly satisfies our new notion of MU-EUF-CMA-C-security in the random oracle model. This work was published at CANS 2014 [Bad14]. The scheme is proven to be secure in the random oracle model under the SXDH assumption in bilinear groups. The security reduction loses a factor of roughly 2.

To circumvent our impossibility result on tight reductions, we apply a technique similar to that of Cramer and Shoup [CS98] for secret keys of the scheme. Namely, given a public key, there are exponentially many secret keys that work for that public key and, in general, it is not possible to efficiently re-randomize over all secret keys. A similar argument does also apply to signatures in the scheme. Namely, we have exponentially many congruence classes of signatures, one class per secret key. Again, it is possible to efficiently re-randomize signatures of one type. Again, re-randomization over all valid signatures will, however, not be possible in general.

A signature of our new scheme is estimated (parameter selection according to [ECR12]) to be of size roughly 3100 bits, when 80 bits of security are required. For the same security level, a public key of our scheme has roughly 1250 bits. We note that signatures of our scheme are longer, compared to, e.g., BLS signatures [BLS01] that are also provably secure in the random oracle model. This does also hold, if we select parameters for BLS signatures in a theoretically sound way, i.e., taking the loss into account. Nonetheless, we are able to show that it is possible to construct a practical tightly-secure MU-EUF-CMA-C-secure signature scheme with a security proof in the random-oracle model.
Outline. In Chapter 2, we recall complexity theoretic basics and foundations, as well as cryptographic ones. In Chapter 3, we will formally state and prove sufficient conditions that rule out tight security proofs. We show in Chapters 4 and 5 that it is possible to circumvent the before established impossibility result by violating its underlying assumptions in the standard and the random oracle model. Finally, in Chapter 6 we summarize and conclude our work and point to future research directions.
2 Preliminaries

After introducing some notation in Section 2.1 we will recall complexity theoretic foundations in Section 2.2. These cover the model of computation that will be considered throughout this thesis (Section 2.2.1) and the complexity assumptions that we will rely on (Section 2.2.2). Finally, Section 2.3 recalls cryptographic foundations. Here, we discuss the role of the security parameter (Section 2.3.1), the relation between symmetric and asymmetric key sizes (Section 2.3.2) and recall basic cryptographic primitives along with their respective security notions (Section 2.3.3).

2.1 Notation

By \([n]\) we denote the set \(n := \{1, 2, \ldots, n\} \subset \mathbb{N}\). We let \([n \setminus j] := [n] \setminus \{j\}\) for \(j \in [n]\). Given two strings \(x, y \in \{0, 1\}^*\) we denote by \(x||y\) the concatenation of \(x\) and \(y\). If \(A\) is a set then by \(a \leftarrow^\$ A\) we denote the action of sampling an element uniformly random from \(A\) and denoting it \(a\). Given a set \(A\) we denote by \(U_A\) the uniform distribution on \(A\).

If \(f : \mathbb{N} \to \mathbb{N}\) and there exists \(c \in \mathbb{N}\) such that \(f(n) < n^c\) except for finitely many \(n\) then we say \(f \in \text{poly}\). We call a function \(\epsilon : \mathbb{N} \to [0, 1]\) negligible if for any \(c \in \mathbb{N}\) there exists \(n_c \in \mathbb{N}\) such that \(\epsilon(n) < n^{-c}\) for all \(n > n_c\). We will often omit dependence on \(n\).

When analyzing asymptotic behavior of functions \(\mathbb{N} \to \mathbb{N}\), we will make use of the Landau-Symbols. That is, given \(f, g : \mathbb{N} \to \mathbb{N}\), we say that \(g \in \mathcal{O}(f)\) if there exists \(c \in \mathbb{N}\) such that \(g(n) \leq c \cdot f(n)\) for all but finitely many \(n \in \mathbb{N}\).
2 Preliminaries

2.2 Complexity Theoretical Foundations

In this section we formally define the computational model that will be considered throughout this thesis. To this end, we shortly recall the notion of a Turing Machine and the metric with respect to which running times are analyzed. Here, we will follow Arora and Barak [AB09]. After these basics we will recall the random-oracle model and interactive Turing machines.

2.2.1 Model of Computation

Turing Machines. A $k$-tape Turing Machine (TM) $M$ is a mechanical machine that has read/write access to $k$ tapes [Tur37]. The length of these tapes is bounded only on one side. The tapes contain cells which each contain a symbol. $M$ may access these through read/write heads which are able to read or manipulate the content of exactly one cell per tape at a time. There exists a special tape that is called input tape. The content of any cell on the input tape must not be manipulated.

More formally, a TM $M$ is a three tuple $M = (\Gamma, Q, \delta)$ where $\Gamma$ and $Q$ are finite non-empty sets and $\delta : Q \times \Gamma^k \to Q \times \Gamma^{k-1} \times \{\text{left, stay, right}\}^k$ with the following properties.

Tape alphabet. We call $\Gamma$ the tape alphabet. It contains symbols that $M$ may read or write on its tapes. In particular, $\Gamma$ contains two special symbols $b$ and $s$, the blank and start symbol, respectively. The content of the leftmost cell on each tape is assumed to be $s$. Since it does not add a huge amount to the running time of $M$ (see below) we assume that $\Gamma := \{0, 1, b, s\}$.

Set of States. $Q$ is called the set of states and contains two special states, $q_{\text{start}}$ and $q_{\text{halt}}$, denoting the initial and final state, respectively.

Transition function. We call $\delta$ the transition function of $M$. That is, if $M$ is in state $q \in Q$, reads $\sigma = (\sigma_1, \ldots, \sigma_k)$ on its $k$ tapes where $\sigma_i \in \Gamma, 1 \leq i \leq k$, and $\delta(q, \sigma) = (q', (\sigma'_2, \ldots, \sigma'_k), z)$ where $z \in \{\text{left, stay, right}\}^k$ then the
symbols \((\sigma_2, \ldots, \sigma_k)\) on the respective tapes will be replaced by \((\sigma'_2, \ldots, \sigma'_k)\), \(M\) will assume state \(q'\) and the read/write heads of \(M\) will move according to \(z\).

The initial state \(q_{\text{start}}\) is defined as the content of all (non-start) cells on all tapes except for the first tape being the blank symbol \(b\) and such that the read/write head on the input tape is on \(s\). The content of the input tape is called input. When \(M\) assumes state \(q_{\text{halt}}\) then the computation terminates. The output of \(M\) is the content of tape \(k\) when state \(q_{\text{halt}}\) is assumed and on input \(x\) is denoted by \(M(x)\).

One step of a TM \(M\) is defined as reading the contents of all tapes, evaluating \(\delta\) with respect to the current state and the content of the cells, manipulating the content of the cells according to \(\delta\) and moving the read/write heads according to \(\delta\).

**Definition 2.1** (Running Time of a Turing Machine and Computing a Function). Let \(f : \{0,1\}^* \rightarrow \{0,1\}^*\), \(M\) be a TM and \(T : \mathbb{N} \rightarrow \mathbb{N}\). We say that \(M\) has running time \(T\) if for all \(x \in \{0,1\}^*\) it holds that \(M\), beginning in state \(q_{\text{start}}\) on input \(x\), assumes state \(q_{\text{halt}}\) after at most \(T(|x|)\) steps. We say that \(M\) computes \(f\) in time \(T\) if it runs in time \(T\) and, when \(x\) is the content of the input tape, outputs \(f(x)\).

We will particularly be interested in Turing Machines that have polynomial running time, i.e., where \(T \in \text{poly}\). Given a function \(f : \{0,1\}^* \rightarrow \{0,1\}^*\) we say that \(f\) is *efficiently computable* if there is a TM \(M\) that computes \(f\) in time \(T \in \text{poly}\). It is well known that the property of a function \(f\) to be efficiently computable does not depend on either the number of tapes that \(M\) has access to or the size of the tape alphabet [AB09]. Therefore, throughout this thesis we will assume the tape alphabet to contain the symbols \(\{0, 1, b, s\}\).

**Probabilistic and Oracle Turing Machines.** We will mainly consider Turing machines that have access to some source of randomness. This is formalized as follows.

**Definition 2.2** (Probabilistic Turing Machine). A *probabilistic TM* \(M\) is a TM that has two transition functions \(\delta_0\) and \(\delta_1\). In
each step each of these transition functions is executed with probability equal to $\frac{1}{2}$. The choice is made independently of anything else for each step.

To model the capability of $M$ to choose uniformly random which transition function to evaluate we proceed as follows: We assume that $M$ has read-only access to a special random tape. Let $M$ run in time $T$. When $M$ is run on input $x$, the random tape contains $T(x)$ non-blank, non-start cells the distribution of the content of which is uniformly random over $\{0, 1\}$. Moreover, in each step, the read/write head is assumed to move right on this tape. This way, we can model $M$ as having only one transition function $\delta$ that depends in particular on the content of the random tape: We define a transition function $\delta$ that evaluates $\delta_0$ if the current cell on the random tape contains 0 and $\delta_1$, otherwise. We write $M(x; r)$ to denote the output of $M$ when run on input $x$ and random tape $r$. By $M(x)$ we denote the distribution of $M(x; r)$ over the uniformly random choices of $r$. We will abbreviate probabilistic polynomial time by PPT.

To formalize the issue of a Turing machine $M$ having black-box access to a Turing machine $A$ we recall the notion of oracle-Turing machines.

**Definition 2.3 (Oracle Turing Machine).** Let $M$ and $A$ be two TMs. We say that $M$ has oracle access to $A$ if it has read/write access to a special oracle tape and two additional states $q_{\text{query}}$ and $q_{\text{response}}$. Whenever $M$ enters state $q_{\text{query}}$ and $x$ is written on the oracle tape, in the next step $M$ will be in state $q_{\text{response}}$ and the content of the oracle tape is distributed according to $A(x)$. Regardless of the running time of $A$ this counts as a single step of $M$. We abbreviate notation and denote by $M^A$ that TM $M$ has oracle access to TM $A$.

Note that if $M$ has oracle access to $A$ then it may only deploy the input/output behavior of $A$. In contrast to that we will say that $M$ has black-box access to $A$ if querying $A$ on input $q$ counts for $T(|q|)$ steps of $M$ where $T$ denotes the running time of $A$. This is formalized via the notion of Interactive Turing Machines.

**Interactive Turing Machines.** An interactive Turing Machine is a TM $M$ that has, besides its working tapes, two special com-
communication tapes, an auxiliary input tape and, additionally, a special switching tape. The access to one communication tape is write-only and to the other communication tape is read-only, as is the access to the auxiliary input tape. The access to the switching tape is read/write. Moreover, the switching tape has only a single non-blank (and non-start) cell. An interactive Turing Machine (ITM) is identified with a bit $b \in \{0, 1\}$ and is said to be active if the content of the switching tape equals $b$. Otherwise it is said to be idle. While an ITM $M$ is idle the state of $M$, the position of its heads and the content of all tapes except for the read-only communication tape remain unchanged. The content of the write-only communication tape written by $M$ during an active period is called \textit{message sent} at that time by $M$ and the content of the read-only communication tape is called \textit{message received} by $M$ at that time.

A linked pair of ITMs is a pair of TMs $M_0$ (identified by 0) and $M_1$ (identified by 1) that share the input tape and a pair of communication tapes such that $M_1$ has write access to $M_0$’s write-only communication tape and vice versa. Each machine may have additional “private” input on its auxiliary input tape. When either $M_0$ or $M_1$ assumes state $q_{\text{halt}}$ when it is active then the other machine halts as well. The output of each ITM is the content of its output tape at that time.

\textbf{Games.} In this thesis we will only consider game-based security notions for cryptographic schemes. As introduced in [BR06], a game $G$, as well as its procedures, will be described in pseudocode. Each game is defined through its code and that of its procedures. We will denote by $G^A \Rightarrow Q$ the event that $Q$ is output by game $G$ when it is run with $A$ (that is, when $A$ has oracle access to the respective procedures). Here, the probability will be taken over the random coins consumed by $G$ and its procedures, as well as the random coins of $A$. If not stated otherwise, procedures of a game sample coins independently and uniform.

\textbf{Algorithms.} The paragraphs above recalled basics about Turing machines. This formalization allows us to rigorously analyze, e.g., the running time that a certain machine takes to evaluate a particular function $f$. However, it is often rather unpractical to
describe Turing machines. Therefore we will resort to algorithms and pseudocode when we want to describe a certain computation. Nonetheless, we keep in mind that the computation is actually done by a TM. Therefore, we will also speak about randomized algorithms, oracle access to algorithms or interactive algorithms.

2.2.2 Complexity Assumptions

Here, we will generically define non-interactive complexity assumptions as well as state concrete assumed-to-be-hard computational problems.

(Bilinear) Groups. By $\mathbb{G} = (\mathcal{G}, g, p)$ we denote the description of a cyclic multiplicative group $\mathcal{G}$ of order $p$ generated by $g$. It is well known that there is an efficient algorithm that on input $1^\kappa$ returns $\mathbb{G}$ such that $2^{\kappa-1} < p \leq 2^\kappa$. We denote this algorithm by $\text{GGen}(1^\kappa)$. Sometimes we want these groups to be equipped with additional properties that are described in the following.

By $\mathbb{G}_\text{asym} = (e, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_T, g_1, g_2, p)$ we denote the description of an asymmetric bilinear group. That is, $e : \mathcal{G}_1 \times \mathcal{G}_2 \rightarrow \mathcal{G}_T$ is an efficiently computable, non-degenerate bilinear map, $g_b$ is a generator of $\mathcal{G}_b$, $|\mathcal{G}_1| = |\mathcal{G}_2| = |\mathcal{G}_T| = p$ where $p$ is prime and there is no efficiently computable homomorphism between $\mathcal{G}_1$ and $\mathcal{G}_2$. It is well known that there is a PPT algorithm that on input $1^\kappa$ returns $\mathbb{G}_\text{asym}$ such that $2^{\kappa-1} < p \leq 2^\kappa$ [GPS06]. We denote this algorithm by $\text{GEN.asym}(1^\kappa)$.

By $\mathbb{G}_\text{sym} = (e, \mathcal{G}, \mathcal{G}_T, g, p)$ we denote the description of a symmetric bilinear group. That is, $e : \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{G}_T$ is an efficiently computable, non-degenerate bilinear map, $g$ is a generator of $\mathcal{G}$ and $|\mathcal{G}| = |\mathcal{G}_T| = p$ where $p$ is prime. It is well known that there is a PPT algorithm that on input $1^\kappa$ returns $\mathbb{G}_\text{sym}$ such that $2^{\kappa-1} < p \leq 2^\kappa$ [GPS06]. We denote this algorithm by $\text{GEN.sym}(1^\kappa)$.

Concrete Complexity Assumptions

Here, we will recall the complexity assumptions that we will refer to later. In particular, we will recall the DLOG assumption, the
2.2 Complexity Theoretical Foundations

Games $k$-Lin$_{real}^{GGen}(A,\kappa)$ and $k$-Lin$_{rand}^{GGen}(A,\kappa)$

$G = (G, g, p) \leftarrow \xi \ GGen(1^\kappa)$
$g_1, \ldots, g_k \leftarrow \xi \ G; \ r_1, \ldots, r_k \leftarrow \xi \ Z_p$
$T \leftarrow g\sum_{i=1}^k r_i; \ T \leftarrow \xi \ G$
$b \leftarrow A(G, g_1, \ldots, g_k, g_1, \ldots, g_k, T)$
return $b$

Game DLOG$_{\text{real}}^{GGen}(A,\kappa)$

$G = (G, g, p) \leftarrow \xi \ GGen(1^\kappa)$
$x \leftarrow \xi \ Z_p$
$x' \leftarrow A(G, g, x)$
return $x \leftarrow ? x' \mod p$

Given generators $g, h_1, \ldots, h_k \in G$ we denote by $k$-Lin$(g, h_1, \ldots, h_k)$ the set

$k$-Lin$(g, h_1, \ldots, h_k) := \left\{ \left( \hat{g}, \hat{h}_1, \ldots, \hat{h}_k \right) : \hat{h}_i = h_i^{r_i} \land \hat{g} = g\sum_{i} r_i \right\}$.

Let $\bar{g} \in G^{k+1}$ be a vector of group elements. We call $k$-Lin$(\bar{g})$ the set of $k$-Lin tuples to base $\bar{g}$ and $(\bar{g}, \bar{h}) \in (G^{k+1})^2$ an instance of $k$-Lin or a $k$-Lin challenge.
We will mainly consider the 1-Lin assumption (the DDH assumption) and the 2-Lin assumption (the DLIN assumption) relative to $\mathcal{G}_{\text{Gen}}$ and define the sets $\text{DDH}(g, h)$ and $\text{DLIN}(g, h, k)$ accordingly. It is well known \cite{NR97, BBM00, LW09, HJ12} that it is possible to efficiently re-randomize instances of DDH and DLIN relative to $\mathcal{G}_{\text{Gen}}$. That is, let $k \in \{1, 2\}$ and $\mathcal{G} \leftarrow \$ \mathcal{G}_{\text{Gen}}(1^\kappa)$. There is an efficient algorithm that, on input a tuple of generators $\vec{g} \in \mathcal{G}_{k+1}$, a tuple $\vec{h} \in \mathcal{G}_{k+1}$, $1^q$ and $\beta \in \{0, 1\}$ outputs $\vec{c}, \vec{f} \in \mathcal{G}_{k+1}, i \in [q]$ such that:

<table>
<thead>
<tr>
<th>$\beta = 0$</th>
<th>$\beta = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\vec{h} \in k\text{-Lin}(\vec{g})$</td>
<td>$\vec{c}_i = \vec{g} \land \vec{f}_i \leftarrow $ k-Lin($\vec{c}_i$)</td>
</tr>
<tr>
<td>$\vec{h} \notin k\text{-Lin}(\vec{g})$</td>
<td>$\vec{c}_i \leftarrow {\mathcal{G} \setminus {1}}^{k+1}$ $\land \vec{f}_i \leftarrow $ k-Lin($\vec{c}_i$)</td>
</tr>
</tbody>
</table>

Next, we define the external Diffie-Hellman assumptions and the double pairing assumptions in bilinear groups. Let $\mathcal{G} = (e, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_T, g_1, g_2, p) \leftarrow \$ \mathcal{G}_{\text{Gen}}(1^\kappa)$ and $c \in \{1, 2\}$.

**Definition 2.6 (S)XDH assumption.** We say that an algorithm $(t, \epsilon)$-breaks the external Diffie-Hellman assumption in $\mathcal{G}_c$ ($c \in \{1, 2\}$) relative to $\mathcal{G}_{\text{Gen}}$ (XDH$_c$ assumption) if it runs in time $t(\kappa)$ and

$$\Pr [\text{XDH}^\text{GEN.asym}_c(\mathcal{A}, \kappa) \Rightarrow 1] - \Pr [\text{XDH}^\text{GEN.asym}_c(\mathcal{A}, \kappa) \Rightarrow 1] \geq \epsilon(\kappa)$$

where the respective games are depicted in Figure 2.2. We say that an attacker $(t, \epsilon)$-breaks the symmetric external Diffie-Hellman assumption relative to $\mathcal{G}_{\text{Gen}}$ (SXDH-assumption) if it $(t, \epsilon)$-breaks XDH$_1$ or XDH$_2$.

Since DDH challenges are efficiently re-randomizable relative to $\mathcal{G}_{\text{Gen}}$, the same holds for XDH$_c$ for $c \in \{1, 2\}$.

**Definition 2.7 (DP assumption \cite{AFG10}).** We say that an algorithm $\mathcal{A}$ $(t, \epsilon)$-breaks the double pairing assumption in $\mathcal{G}_2$ relative to $\mathcal{G}_{\text{Gen}}$ (DP$_2$ assumption) if it runs in time $t(\kappa)$ and

$$\Pr [\text{DP}^\text{GEN.asym}_2(\mathcal{A}, \kappa) \Rightarrow 1] \geq \epsilon$$
2.2 Complexity Theoretical Foundations

Games \( \text{XDH}^\text{GEN.asym}_{c,\text{real}}(\mathcal{A}, \kappa) \) and \( \text{XDH}^\text{GEN.asym}_{c,\text{rand}}(\mathcal{A}, \kappa) \)

\[
\begin{align*}
\mathbb{G} &= (e, G_1, G_2, G_T, g_1, g_2, p) \leftarrow \$ \text{GEN.asym}(1^\kappa) \\
h &\leftarrow \$ G_c; a \leftarrow \$ \mathbb{Z}_p \\
T &\leftarrow \$ h^a; T &\leftarrow \$ G_c \\
b &\leftarrow \$ A(\mathbb{G}, g, h, g^a, T) \\
\text{return } b
\end{align*}
\]

Figure 2.2: Games \( \text{XDH}^\text{GEN.asym}_{c,\text{real}}(\mathcal{A}, \kappa) \) and \( \text{XDH}^\text{GEN.asym}_{c,\text{rand}}(\mathcal{A}, \kappa) \).

where Game DP\(_2(\text{GGen})\) is described in Figure 2.3. We define the DP\(_1\) assumption analogously.

**Lemma 2.1** ([AFG+10]). For any attacker \( A \) that \((t_{\text{DP}_c}, \epsilon_{\text{DP}_c})\)-breaks the DP\(_c\) assumption relative to \( \text{GEN.asym} \) (where \( c \in \{1, 2\} \)) there exists an attacker \( A \) that \((t_{\text{XDH}_c}, \epsilon_{\text{XDH}_c})\)-breaks the XDH\(_c\) assumption relative to \( \text{GEN.asym} \) where \( t_{\text{DP}_c} \approx t_{\text{XDH}_c} \) and \( \epsilon_{\text{XDH}_c} \geq \epsilon_{\text{DP}_c} \).

**Proof.** Let wlog \( c = 2 \). Algorithm \( A \), given an XDH\(_2\) instance \((\mathbb{G}, g, h, \hat{g}, \hat{h})\), runs \( A \) as a subroutine on input \((\mathbb{G}, g, \hat{g})\). When \( A \) outputs \((1, 1) \neq (z, r)\) such that \( e(z, g) \cdot e(r, \hat{g}) = 1 \) we know that \( \log_z(r) = -\log_{\hat{g}}(g) \). If it also holds that \( e(z, h) \cdot e(r, \hat{h}) = 1 \) we know that \( \log_z(r) = -\log_{\hat{h}}(h) \). In this case, \( A \) will output 1. Otherwise, it will return 0. \( \square \)

\[
\begin{align*}
\text{Game DP}_2^\text{GEN.asym}(\mathcal{A}, \kappa) \\
\mathbb{G} = (e, G_1, G_2, G_T, g_1, g_2, p) &\leftarrow \$ \text{GEN.asym}(1^\kappa) \\
(g_z, g_r) &\leftarrow \$ G^2_2 \\
(z, r) &\leftarrow A(\mathbb{G}, g_z, g_r) \\
\text{return } (z, r) \neq (1, 1) \land e(z, g_z) \cdot e(r, g_r) = ? 1
\end{align*}
\]

Figure 2.3: Game DP\(_2^\text{GEN.asym}(\mathcal{A}, \kappa)\).
Generic Non-interactive Complexity Assumptions.

For our general impossibility result on tight reductions in Chapter 3 we need the following definition of a interactive complexity assumption which is due to [AGO11].

**Definition 2.8 (Non-interactive Complexity Assumption).** A non-interactive complexity assumption $N = (\text{Gen}, \text{Vfy}, \text{Triv})$ is a triple of algorithms. The PPT algorithm $\text{Gen}$ is called the instance generator. On input $1^\kappa$ it outputs a pair $(c, w)$ where $c$ is called challenge and $w$ is called witness. We silently assume that $1^\kappa$ is included in $c$. The verification algorithm $\text{Vfy}$ takes as input $(c, w)$ and a proposed solution $s$ and outputs 1 if $s$ is indeed a solution to the challenge $c$ and 0 otherwise. The PPT algorithm $\text{Triv}$ implements a trivial attack strategy. Let $t : \mathbb{N} \to \mathbb{N}$ and $\epsilon : \mathbb{N} \to [0, 1]$. We define Game $\text{NICA}_N$ in Figure 2.4 and say that an algorithm $\mathcal{A}$ $(t, \epsilon)$-breaks $N$ if $\mathcal{A}$ runs in time $t = t(1^\kappa)$ and it holds that

$$|\Pr [\text{NICA}_N(\mathcal{A}, \kappa) \Rightarrow 1] - \Pr [\text{NICA}_N(\text{Triv}, \kappa) \Rightarrow 1]| \geq \epsilon = \epsilon(\kappa).$$

Via $\text{Triv}$, the above definitions covers both search problems (where the size of the set $S$ of solutions is large and $\text{Triv}$ samples uniformly random from $S$) and decisional complexity assumptions (where $\text{Triv}$ outputs 1 with probability $1/2$ and 0 otherwise and the set of solution is $S = \{0, 1\}$).

**Example 2.1.** We note that the concrete assumptions from the previous subsection may also be defined in terms of definition 2.8. For instance, for the Diffie-Hellman assumption relative to $\text{GGen}$ we define the following algorithms.

---

**Figure 2.4: Game $\text{NICA}_N(\mathcal{A}, \kappa)$.**

<table>
<thead>
<tr>
<th>Game $\text{NICA}_N(\mathcal{A}, \kappa)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(c, w) \leftarrow $ $N.\text{Gen}(1^\kappa)$</td>
</tr>
<tr>
<td>$s \leftarrow \mathcal{A}(c)$</td>
</tr>
<tr>
<td><strong>return</strong> $N.\text{Vfy}(c, w, s)$</td>
</tr>
</tbody>
</table>
2.3 Cryptographic Foundations

**Instance Generation.** Algorithm $\text{Gen}(1^\kappa)$ runs $G \leftarrow (G, g, p) \leftarrow \$ G\text{Gen}(1^\kappa)$ and $h \leftarrow \$ G$ Next, it samples $a \leftarrow \$ \mathbb{Z}_p$ and $\Psi \leftarrow \$ G$ uniformly at random. Finally, it flips a random coin $\delta \leftarrow \$ \{0, 1\}$ and sets $T = (h^a)^\delta \cdot \Psi^{1-\delta} \in G$. It returns

$$(c, w) \leftarrow ((G, g, g^a, h, T), (a, \delta, \Psi)).$$

*Note that we have $S = \{0, 1\}$.*

**Verification.** $\text{Vfy}(c, w, b')$ returns $T = (h^a)^b' \cdot \Psi^{1-b'}$.

**Trivial Attack Strategy.** $\text{Triv}$ samples $b \leftarrow \$ \{0, 1\}$.

For ease of readability we will not express concrete assumptions this way.

### 2.3 Cryptographic Foundations

We expect from a good cryptographic scheme to be secure. This is easily understandable. However, formalizing what it means for a scheme to be secure is not straightforward. On the one hand we have to carefully define a scheme’s security properties and on the other hand we have to define when an adversary is considered successful in breaking these properties. Informally, we call a scheme secure with respect to a security notion if an attacker that uses a reasonable amount of computational power is not very likely to succeed in breaking the system. After discussing these issues in the following subsections the last subsection of this chapter recalls the cryptographic primitives and their respective security notions that we will refer to later.

#### 2.3.1 The Security Parameter and Symmetric Key Size

Security of cryptographic schemes is stated with respect to a security parameter $\kappa$. The larger $\kappa$ the more computational power is needed to break the system. We consider the running times of all algorithms that are needed to implement a specific scheme relative to $\kappa$ (in unary) and want the success probability of an efficient adversary in “breaking” the scheme to tend to zero very
fast as the security parameter grows. Namely, the success prob-
ability of any efficient adversary should be upper bounded by a
negligible function \( \epsilon(\kappa) \) as \( \kappa \) grows. If this the case, then the
success probability of the adversary is still upper bounded by a
negligible function \( \epsilon'(\kappa) \) (and thus close to 0 for suitably large \( \kappa \))
even if the adversary runs its strategy polynomially (in \( \kappa \)) many
times.

Roughly saying, the security parameter refers to the key length
of a symmetric cipher. That is, when we say that a scheme has \( \kappa \)
bits security (which refers to the security parameter being \( \kappa \)) we
mean that the scheme is as hard to break as a symmetric cipher
(like DES [DES77] or AES [AES01]) with an \( \kappa \) bit key, assuming
the best possible attacks against these ciphers are brute force
attacks, i.e., assuming the ciphers to be ideal. Stated otherwise,
when a scheme has \( \kappa \) bits of security, an efficient adversary has
about the same success probability against the scheme as it has
when mounting a brute-force attack against an \( \kappa \) bit symmetric
cipher which can be attacked best by brute force. More formally,
we define the security level as follows.

**Definition 2.9 (Security Level).** We define the work factor \( w_A \)
of an algorithm \( A \) as its running time \( t_A \) over its probability of
success \( \epsilon_A \), \( w_A := \frac{t_A}{\epsilon_A} \). We say that a cryptographic scheme has \( \kappa \)
bits of security when a work factor of \( 2^\kappa \) is sufficient and required
to break it. In this case, we say that the scheme has \( \kappa \) bits of
security, or that it has security level \( \kappa \).

**Remark 2.1 (On the running time of the adversary).** Looking
ahead, we note that though we consider only the success proba-
bility of adversaries that have polynomial running time we also
have to take into account computationally unbounded adversaries
when constructing reductions for our schemes. This is due to the
fact that the adversary may be able to efficiently solve a computa-
tional problem that we assume to be hard. However, if this is
the case, we want to extract this capability from the adversary
in the sense that we want to run the adversary as a subroutine
(in a black-box manner) to solve an assumed-to-be-hard problem
ourselves. Now, if the adversary, i.e., the subroutine and the
calling algorithm are efficient, we obtain an efficient algorithm
that solves the assumed-to-be-hard problem.
2.3.2 Asymmetric Key Size and Exact Security

This thesis focuses on constructing and analyzing asymmetric cryptographic schemes. These schemes are often implemented over, e.g., integer rings, finite fields or elliptic curves. It is not straightforward to tell of which size these structures have to be to meet a certain level of security. For example, we know that the best (publicly known) algorithm to factorize integers is the general number field sieve which is estimated to factor $n$-bit integers in asymptotic time

$$L(n) = Ae^{(C+o(1)) \cdot n^{\frac{1}{3}} \cdot (\ln(n))^{\frac{2}{3}}}$$

where $A$ and $C = \left( \frac{64}{9} \right)^{\frac{1}{3}}$ are constants [ECR12]. To which security level does this refer? Here, we will follow the approach taken by the European Network of Excellence in Cryptology which proceeds as follows [ECR12, Section 6.2.1]. For factoring based schemes it is first estimated to which symmetric key size a 512-bit composite modulus refers. Once this size is determined the attack complexity of the general number field sieve is extrapolated. This leads to the following expression determining the effective key size of an $n$-bit factoring based modulus:

$$s(n) = \left( \frac{64}{9} \right)^{\frac{1}{3}} \cdot \log_2(e) \cdot (n \cdot \ln(2))^{\frac{1}{3}} \cdot (\ln(n \cdot \ln(2)))^{\frac{2}{3}} - 14 \quad (2.2)$$

where $\log_x(y)$ denotes the logarithm of $y$ to base $x$ and $\ln(y) = \log_e(y)$ denotes the natural logarithm.

For DLOG based schemes the “half-size principle” is recommended. I.e., when $n$ bits of security are required it is recommended to implement the scheme in groups of order at least $2^{2n}$. However, when such schemes are to be implemented over finite prime-order fields, then, to thwart attacks that deploy the structure of the field, the same considerations are made for DLOG based schemes as for factoring based schemes. Namely, the size $2^{2n}$ group needs to be embedded in a group of higher order. We also use equation 2.2 to determine the effective key size of that higher order group since the difficulty of solving DLOG in $n$ bit prime order fields is asymptotically equivalent to factoring $n$ bit integers [KL07].


2 Preliminaries

$\begin{align*}
\text{Game \text{coll-res}_H}(\mathcal{A}, \kappa) \\
k \leftarrow \$ \text{H.Gen}(1^\kappa) \\
(m, m') \leftarrow \mathcal{A}(1^\kappa, k) \\
\text{return } H(m) = ? H(m') \land m \neq m'
\end{align*}$

Figure 2.5: Collision Resistance Security Game for Hash-functions.

2.3.3 Basic Primitives

Hash Functions

Let $\mathcal{K} = \{\mathcal{K}_\kappa\}_{\kappa \in \mathbb{N}}$ and $R = \{R_\kappa\}_{\kappa \in \mathbb{N}}$ be families of sets. A hash function $H$ is a pair of efficient algorithms $H = (\text{Gen}, \text{Eval})$ such that $k \leftarrow \$ \text{Gen}(1^\kappa)$ samples a key and for any $m \in \{0, 1\}^*$ it holds that $\text{Eval}(k, m) \in R_\kappa$ if $k \in \mathcal{K}_\kappa$. If it is clear from the context we will write $H(m)$ instead of $\text{Eval}(k, m)$.

We say that an attacker $\mathcal{A}(t, \epsilon)$-breaks the collision-resistance of a hash function $H$ if it runs in time $t(\kappa)$ and

$$\Pr[\text{coll-res}_H(\mathcal{A}, \kappa) \Rightarrow 1] \geq \epsilon$$

where $\text{Game \text{coll-res}_H(\mathcal{A}, \kappa)}$ is described in Figure 2.5.

Non-interactive Proof Systems

Given a binary relation $R \subseteq X \times W$ and $(x, w)$ such that $R(x, w)$ we call $x$ the statement and $w$ the witness. A non-interactive proof system $\text{NIPS} = (\text{Gen}, \text{Prove}, \text{Vfy})$ for witness relation $R$ is a three-tuple of PPT algorithms.

Common Reference String. The algorithm $\text{CRS} \leftarrow \$ \text{Gen}(1^\kappa)$, on input $1^\kappa$, returns a common reference string.

Prove. The prove algorithm inputs $(x, w)$ such that $R(x, w)$ and returns a proof $\pi \leftarrow \$ \text{Prove}(\text{CRS}, x, w)$ with respect to $\text{CRS}$.

Verification. The verification algorithm accepts or rejects a purported proof, $\text{Vfy}(\text{CRS}, x, \pi) \in \{0, 1\}$. 
2.3 Cryptographic Foundations

Figure 2.6: Witness Indistinguishability Security Game

**Definition 2.10.** We call NIPS a Witness Indistinguishable Proof of Knowledge (NIWI-PoK) for \( R \), if the following conditions are satisfied:

**Perfect completeness.** For all \( \kappa \in \mathbb{N} \) it holds that if \( R(x, w) \)

\[
\Pr \left[ \text{NIPS.Vfy}(\text{CRS}, x, \pi) = 1 : \text{CRS} \leftarrow \text{NIPS.Gen}(1^{\kappa}) \wedge \pi \leftarrow \text{Prove}(\text{CRS}, x) \right] = 1
\]

**Perfect Witness Indistinguishability.** The game for witness indistinguishability is depicted in Figure 2.6. We require perfect witness indistinguishability, that is:

\[
\Pr \left[ W_{R,0}^{\text{NIPS}}(A, \kappa) \Rightarrow 1 \right] = \Pr \left[ W_{R,1}^{\text{NIPS}}(A, \kappa) \Rightarrow 1 \right]
\]

**Simulated CRS.** There exists an algorithm \((\text{CRS}_{\text{sim}}, \tau) \leftarrow \mathcal{E}_0 \)

that, on input \( 1^{\kappa} \), outputs a simulated common reference string \( \text{CRS}_{\text{sim}} \) and a trapdoor \( \tau \).

**Perfect Knowledge Extraction on Simulated CRS.** Let \( \text{CRS}_{\text{sim}} \)

be a simulated CRS, i.e., \((\text{CRS}_{\text{sim}}, \tau) \leftarrow \mathcal{E}_0(1^{\kappa}) \). Then we require the existence of an algorithm \( \mathcal{E}_1 \) such that for all \((\pi, x) \leftarrow A \) such that \( \text{NIPS.Vfy}(\text{CRS}_{\text{sim}}, x, \pi) = 1 \) it holds that

\[
\Pr \left[ w \leftarrow \mathcal{E}_1(\text{CRS}_{\text{sim}}, \pi, x, \tau) : (x, w) \in R \right] = 1
\]

**Secure NIWI-PoK.** Let \( \text{CRS}_{\text{real}} \leftarrow \text{NIPS.Gen}(1^{\kappa}) \) and \((\text{CRS}_{\text{sim}}, \tau) \leftarrow \mathcal{E}_0(1^{\kappa}) \). We say that an algorithm \((t, \epsilon_{\text{CRS}})\)-breaks
the security of a NIWI-PoK if it runs in time $t$ and it holds that
\[
\Pr[A(\text{CRS}_{\text{real}}) = 1] - \Pr[A(\text{CRS}_{\text{sim}}) = 1] \geq \epsilon_{\text{CRS}}
\]

If $\text{CRS} \leftarrow \mathcal{S} \text{Gen}(1^\kappa)$ we call CRS hiding and if $(\text{CRS}_{\text{sim}}, \cdot) \leftarrow \mathcal{S} \mathcal{E}_0(1^\kappa)$ we call $\text{CRS}_{\text{sim}}$ binding. It is easy to verify that perfect witness indistinguishability on a hiding CRS is preserved if many statements are proven. An explicit proof is given in [BHJ+15, Appendix B].

**Digital Signature Schemes**

A digital signature scheme $\text{SIG}$ is a four-tuple of PPT algorithms $\text{SIG} = (\text{Setup, Gen, Sign, Vfy})$ with the following syntax:

**Public Parameters.** The parameter generation algorithm inputs the security parameter $1^\kappa$ and returns public parameters $\Pi \leftarrow \mathcal{S} \text{Setup}(1^\kappa)$. These define a message space $\mathcal{M}$, a signature space $\Sigma$ and a key space $\mathcal{K}$. We will silently assume that $1^\kappa$ is included in $\Pi$. If this algorithm is not explicitly required, it just outputs $1^\kappa$.

**Key Generation.** On input $\Pi$, the key generation algorithm outputs a key pair $(vk, sk) \in \mathcal{K}$, $(vk, sk) \leftarrow \mathcal{S} \text{Gen}(\Pi)$. Even if not explicitly stated we assume that $vk$ contains at least $1^\kappa$ and that $sk$ contains $vk$.

**Signature Generation.** On input a secret key $sk$ and a message $m$, the signing algorithm returns a signature $\sigma \leftarrow \mathcal{S} \text{Sign}(sk, m)$.

**Verification.** The verification algorithm accepts or rejects a purported signature $\sigma$ over message $m$ with respect to a given verification key $vk$ indicating whether the signature is valid or not. That is, $\text{Vfy}(vk, m, \sigma) \in \{0, 1\}$.

For correctness we require that for all $\kappa \in \mathbb{N}$ and any message $m$ it holds that
\[
\Pr\left[\begin{array}{l}
\Pi \leftarrow \mathcal{S} \text{Setup}(1^\kappa); \\
\text{Vfy}(vk, m, \sigma) = 1 : (vk, sk) \leftarrow \mathcal{S} \text{Gen}(\Pi); \\
\sigma \leftarrow \mathcal{S} \text{Sign}(sk, m)
\end{array}\right] = 1
\]
In the sequel, we will assume the message space to be $\{0,1\}^*$. It is well-known that such scheme can be constructed from a signature scheme with arbitrary message space $\mathcal{M}$ by applying a collision-resistant hash function $H : \{0,1\}^* \rightarrow \mathcal{M}$ to the message before signing (assuming that such hash function exists).

Today, the standard security notion for digital signatures is existential-unforgeability under chosen message attacks (EUF-CMA) security $[GMR88]$. The EUF-CMA-security game is depicted in Figure 2.7.

**Definition 2.11 (EUF-CMA-security).** We say that an attacker $(t, \mu, \epsilon)$-breaks the EUF-CMA-security of a signature scheme $\text{SIG}$ if it runs in time $t(\kappa)$ and

$$\Pr \left[ \text{EUF-CMA}^{\text{SIG}}(A, \mu, \kappa) \Rightarrow 1 \right] \geq \epsilon(\kappa)$$

### Public Key Encryption

A public key encryption scheme $\text{PKE} = (\text{Setup}, \text{Gen}, \text{Enc}, \text{Dec})$ is a four-tuple of PPT algorithms with the following syntax:

**Public Parameters.** The parameter generation algorithm, on input the security parameter $1^\kappa$, returns public parameters $\Pi \leftarrow ^{\$} \text{Setup}(1^\kappa)$. These define a message space $\mathcal{M}$, a ciphertext space $\Gamma$ and a key space $\mathcal{K}$. We will silently assume that $1^\kappa$ is included in $\Pi$. If this algorithm is not explicitly required, it just outputs $1^\kappa$.

**Key Generation.** On input $\Pi$, the key generation algorithm outputs a key pair, $(pk, sk) \leftarrow ^{\$} \text{Gen}(\Pi)$. Even if not explicitly stated we assume that $pk$ contains at least $\Pi$ (and thus $1^\kappa$) and that $sk$ contains $pk$. 

---

**Figure 2.7:** The EUF-CMA-security game for digital signatures.
Encryption. The encryption algorithm takes as input a message $m$ and a public key $pk$ and returns a ciphertext $c \leftarrow \$ \text{Enc}(pk, m)$.

Decryption. The decryption algorithm on input a secret key $sk$ and a ciphertext $c$ returns a message $m \leftarrow \text{Dec}(sk, c)$ or an error symbol $\perp$.

For correctness we require that for all $\kappa \in \mathbb{N}$ and any message $m$ it holds that

$$\Pr \left[ \text{Dec}(sk, c) = m : \Pi \leftarrow \$ \text{Setup}(1^\kappa); (pk, sk) \leftarrow \$ \text{Gen}(\Pi); c \leftarrow \$ \text{Enc}(pk, m) \right] = 1.$$ 

The standard security notion for public key encryption in the multi-user setting goes back to Bellare et al. \cite{BBM00}. Here, we differ from their security notion, as we allow adaptive corruptions. The security game is depicted in Figure 2.8.

Definition 2.12 (MU-IND-CPA-C-security). We say that an attacker $(t, n, \epsilon)$-breaks the MU-IND-CPA-C-security of a public key encryption scheme $\text{PKE}$ if it runs in time $t$ and

$$\Pr \left[ \text{MU-IND-CPA-C}_{\text{PKE}}^n(A, \kappa) \Rightarrow 1 \right] \geq \epsilon$$

Figure 2.8: MU-IND-CPA-C-security Game. The attacker has access to an encryption oracle $\mathcal{O}.\text{Encrypt}$ which may be queried only once and a corrupt oracle $\mathcal{O}.\text{Corrupt}$.
3 Sufficient Conditions for the Impossibility of Tight Reductions in Cryptography

In this chapter, we consider conditions of cryptographic schemes that generally rule out tight reductions. To this end, we first recall the impossibility result of Coron and its refinements.

A closer look at Coron’s result and its refinements. Coron analyzed reductions from breaking a computational hardness assumption to breaking the EUF-CMA-security (Definition 2.11) of a unique signature scheme [Cor02]. The paper rules out tight security proofs for any such signature scheme if the reduction is simple\(^1\) (cf. Definition 3.8). More detailed, the result says that for any simple reduction \(R\) there exists an algorithm \(B\) that runs in roughly the same time as \(R\) and breaks the considered computational hardness assumption directly with probability

\[
\epsilon_B \geq \epsilon_R - \frac{\epsilon_A}{\exp(1) \cdot n} \cdot \left(1 - \frac{n}{|\mathcal{M}|}\right)^{-1}.
\]

(3.1)

Here, \(\epsilon_R\) is the success probability of \(R\), \(\epsilon_A\) is the success probability of the attacker \(A\), \(\mathcal{M}\) is the message space, and \(n\) is the number of signatures the adversary may obtain (i.e. the number of signature queries).

If the underlying computational problem is hard, we must have \(\epsilon_B = \text{negl}(\kappa)\). Assuming that \(|\mathcal{M}| \gg n\) we obtain that \(\epsilon_R \leq \text{negl}(\kappa)\).

\(^1\)The term simple was coined only recently in [LW14], but Coron actually considers simple reductions.
On the Impossibility of Tight Reductions

\[ \epsilon_A / (\exp(1) \cdot n) + \text{negl}(\kappa). \]

The result was revisited by \[\text{[KK12]}\] and generalized by Hofheinz \textit{et al.} \[\text{[HJK12]}\] to hold for any efficiently re-randomizable signature scheme.

Limitations of known meta-reductions

Excepting the result of Lewko and Waters \[\text{[LW14]}\], the technique of Coron was applied exclusively to digital signatures \[\text{[Cor02, HJK12, KK12]}\] (\[\text{[LW14]}\] considers \textit{hierarchical identity based encryption}). A reason for this seems to be the fact that the bound in Equation 3.1 vanishes if \( n \approx |M| \). This can be easily seen by setting \( n = |M| - 1 \). In this case:

\[ \epsilon_R \leq \epsilon_A \cdot \frac{\exp(-1)}{n} \cdot \left( \frac{1}{n+1} \right)^{-1} + \text{negl}(\kappa) \]

The arguments from \[\text{[Cor02, HJK12, KK12]}\] strongly rely on the fact that \( |M| >> n \). Therefore, it is not clear how to apply this technique if \( |M| \approx n \). We note that \( |M| \approx n \) may occur whenever \( |M| \) is polynomially bounded. Though uninteresting for digital signatures, such situation occurs when we consider cryptographic primitives, like digital signatures or PKE in the multi-user setting where we allow the adversary to corrupt parties of its choosing. In such model, the adversary is provided with polynomially many public keys \( \{pk_1, \ldots, pk_n\} \) at the beginning of the security experiment. Next, it may corrupt all but one of these keys and finally, it is able to “break security” of the considered scheme if it is able to compute a secret key for an uncorrupted public key. A similar situation occurs when we consider non-interactive key exchange in a multi-user model with corruptions. How can we analyze the existence of inherent tightness bounds in this setting?

Contribution. We develop a new meta-reduction that is applicable in settings where \( n \) is polynomially bounded. This allows us to achieve a better bound:

\[ \epsilon_B \geq \epsilon_R - \frac{1}{n} \]

Compared to Equation 3.1, this bound is much simpler and is also applicable if \( n \approx |M| \). In contrast to \[\text{[Cor02, HJK12, KK12]}\],
we rule out tight reductions from any non-interactive complexity assumption (cf. Definition 2.8). This includes decisional assumptions, as well as search problems. Our analysis is simple: we avoid the use of [Cor02 Lemma 1] which has a quite technical proof.

The works of [Cor02, HJK12, KK12] considered the EUF-CMA-security notion for digital signatures. We consider a security notion where the attacker is called on a public key \( vk \) and \( n \) distinct messages. It may non-adaptively obtain signatures for all-but-one of these messages. Finally, it is considered successful if it manages to output a signature for the unsigned message. We develop a technique to rule out tight reductions in this setting. Because our security notion is weaker, our impossibility result applies immediately to stronger notions, as well.

After refining the result for digital signatures as described above, we consider general computable relations \( R(x, y) \) and the following abstract security experiment (The reader may think of \( R \) as a relation on messages and signatures as given by a public key of a signature scheme).

1. The attacker is called on input a description of a relation \( R(x, y) \) and \( n \) statements \( y_1, \ldots, y_n \). It outputs an index \( j \in [n] \).

2. In response, it obtains witnesses \( x_i \) for all statements except for \( y_j \). That is, it holds \( R(x_i, y_i) \) for all obtained witnesses \( x_i, i \neq j \).

3. When the game terminates, the adversary outputs \( x_j \) and is considered successful if \( R(x_j, y_j) \).

We show that standard security notions for many cryptographic primitives imply security with respect to this abstract experiment. Our main result says that any simple reduction that reduces breaking a non-interactive complexity assumption \( N \) to breaking the “security” of \( R \) either loses a factor of at least \( n \) if \( R \) is efficiently re-randomizable and efficiently computable or the underlying problem is easy.

The results are contained in [BJLS15]. My contribution is the elaboration of the proof technique and the extension of this
3 On the Impossibility of Tight Reductions

technique to settings where the reduction is allowed to rewind the adversary multiple times sequentially, as well as applying our result to, e.g., public key encryption and digital signatures in the multi-user setting.

Outline. In Section 3.1 we introduce the topic of metareductions by improving the above mentioned bound due Coron, Hohenheinz et al. and Kakvi and Kiltz for digital signatures in the multi user setting. To this end, we develop a new weak security notion for digital signature in the single user setting (Section 3.1.1) and show that under certain assumptions on the signature scheme there can not exist a tight reduction from any non-interactive complexity assumption to breaking security with respect to this notion (Sections 3.1.2 and Section 3.1.3). Section 3.2 will generalize the technique from Section 3.1 by considering general families of computable relations. We first define these families of relations and the properties we require (Section 3.2.1) and then prove our main result in this chapter (Section 3.2.2). In Section 3.3 we show how this generalization allows us to apply the main result from Section 3.2 to, e.g., public key encryption in the multi user setting with corruptions and discuss practical of our result in Section 3.4. Finally, we discuss how our result can be circumvented when constructing cryptographic schemes in Section 3.5.

3.1 Warm-Up: Improved Bound for Digital Signatures

Unique and re-randomizable signatures. Let \( \Sigma(vk,m) := \{\sigma : \text{Vfy}(vk,m,\sigma) = 1\} \) denote the set of all valid signatures \( \sigma \) w.r.t. a given message \( m \) and verification key \( vk \).

Definition 3.1 (Unique signatures). We say that a signature scheme \( \text{SIG} \) is unique, if \( |\Sigma(vk,m)| = 1 \) for all \( vk \) and \( m \).

Definition 3.2 (Re-randomizable signatures). We say that \( \text{SIG} \) is \( t_{\text{ReRand}} \)-re-randomizable, if there is an algorithm \( \text{SIG.ReRand} \) that takes as input \( (vk,m,\sigma) \) and outputs a signature \( \sigma' \leftarrow \$ \text{SIG.ReRand}(vk,m,\sigma) \) with the following properties:
3.1 Warm-Up: Improved Bound for Digital Signatures

1. SIG.ReRand runs in time at most $t_{\text{ReRand}}$

2. If $Vfy(vk,m,\sigma) = 1$, then $\sigma'$ is distributed uniformly over $\Sigma(vk,m)$.

Remark 3.1. Note that we do not put any bounds on $t_{\text{ReRand}}$. Thus, any signature scheme is $t_{\text{ReRand}}$-re-randomizable for sufficiently large $t_{\text{ReRand}}$. However, there are many examples of signature schemes which are efficiently re-randomizable, like the class of schemes considered in [HJK12]. In particular, every unique signature scheme is efficiently re-randomizable.

Unforgeability under Static Message Attacks. Recall that the lower tightness bounds of [Cor02, KK12, HJK12] consider reductions to breaking the existential unforgeability under adaptive chosen-message attacks (EUF-CMA, see Definition 2.11) of a signature scheme. We will consider notion of universal unforgeability under static chosen-message attacks (UF-SMA) instead. This notion is strictly weaker than EUF-CMA-security [BJLS15].

Remark 3.2. Note that a lower tightness bound for UF-SMA security is a strictly stronger result than a corresponding bound for EUF-CMA security, because it even rules out the existence of tight reductions for this weaker definition.

The UF-SMA security experiment is depicted in Figure 3.1. The essential difference to EUF-CMA security is that the attacker is provided with a public key $vk$ and $n$ random, but pairwise distinct messages $m_1, \ldots, m_n$. It may non-adaptively ask for signatures for all but one of these messages, but not for any other messages. The attacker is successful, if it finally outputs a valid signature over the message that was not signed before. In contrast, an EUF-CMA attacker may ask for signatures of arbitrary messages $m_1, \ldots, m_n$, and may forge a signature for an arbitrary message $m^*$, with the only restriction that $m^* \not\in \{m_1, \ldots, m_n\}$.

Definition 3.3. Let $\text{UF-SMA}^n_{\text{SIG}}(A, \kappa)$ denote the UF-SMA security experiment depicted in Figure 3.1, executed with signature scheme $\text{SIG}$ and attacker $A = (A_1, A_2)$. We say that $A (t_A, n, \epsilon_A)$-breaks the UF-SMA-security of $\text{SIG}$, if it runs in time $t_A$ and

$$\Pr [\text{UF-SMA}^n_{\text{SIG}}(A, \kappa) \Rightarrow 1] \geq \epsilon_A$$
Game $\text{UF-SMA}_\text{SIG}^n(\mathcal{A}, \kappa)$

$\Pi \leftarrow \$ \text{SIG.Setup}(1^\kappa); \rho_\mathcal{A} \leftarrow \$ \text{SIG.Gen}(\Pi)$

$m_1, \ldots, m_n \leftarrow \$ \mathcal{M}$ s.t. $m_i \neq m_j$ for all $i \neq j$

$\sigma_i \leftarrow \$ \text{SIG.Sign}(sk, m_i)$ for all $i \in [n]$

$(j, st) \leftarrow \mathcal{A}_1(vk, (m_i)_{i \in [n]}; \rho_\mathcal{A})$

$\sigma_j \leftarrow \mathcal{A}_2(st, (\sigma_i)_{i \in [n] \setminus j})$

return $\text{SIG.Vfy}(vk, m_j, \sigma_j)$

Figure 3.1: The UF-SMA-security Game with attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$.

Remark 3.3. Observe that the messages in the UF-SMA security experiment from Figure 3.1 are chosen at random (but pairwise distinct). We do this for simplicity, but stress that for our tightness bound we actually do not have to make any assumption about the distribution of messages, apart from being pairwise distinct. For instance, the messages could alternatively be the lexicographically first $n$ messages of the message space, for instance.

3.1.1 Simple reductions from non-interactive complexity assumptions to breaking UF-SMA-security.

A reduction from breaking the UF-SMA-security of a digital signature scheme $\text{SIG}$ to breaking the security of a non-interactive complexity assumption $N = (\text{Gen}, \text{Vfy}, \text{Triv})$ (cf. Definition 2.8) is an algorithm, which turns an attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ according to Definition 3.3 into an algorithm $r$-$R^\mathcal{A}$ according to Definition 2.8.

Following [Cor02, KK12, HJK12, LW14], we will consider a specific class of reductions in the sequel. We consider reductions having black-box access to the attacker, and which execute the attacker only $r$ times sequentially. Following [LW14], we call such reductions $r$-simple. At first sight we heavily constrain the class of reductions to that our result applies. However, as ex-
Algorithm $r-R^A(c; \rho_R)$

\[ st_{R_0,1} \leftarrow R_0(c, \rho_R) \]

for $1 \leq l \leq r$ do:

\[ (vk^l, (m_i^l)_{i \in [n]}, \rho_A, st_{R_l,2}) \leftarrow R_{l,1}(st_{R_l,1}) \]

\[ (j^{l*}, st_A) \leftarrow A_1(vk^l, (m_i^l)_{i \in [n]}; \rho_A) \]

\[ ((\sigma^l_i)_{i \in [n] \setminus j^{l*}}, st_{R_l,3}) \leftarrow R_{l,2}(j^{l*}, st_{R_l,2}) \]

\[ \sigma_{j^{l*}} \leftarrow A_2((\sigma^l_i)_{i \in [n] \setminus j^{l*}}, st_A) \]

\[ st_{R_{l+1},1} \leftarrow R_{l,3}(\sigma_{j^{l*}}, j^{l*}, st_{R_l,3}) \]

\[ s \leftarrow R_3(st_{R_{r+1,1}}) \]

return $s$

Figure 3.2: Algorithm $r-R^A$ that solves a non-interactive complexity assumption according to Definition 2.8, constructed from a $r$-simple reduction $r-R = (R_0, (R_{l,1}, R_{l,2}, R_{l,3})_{l \in [r]}, R_3)$ and an attacker $A = (A_1, A_2)$.

plained in [LW14], we include reductions that perform hybrid steps. Moreover, most reductions in cryptography are simple.

For preciseness and clarity, we define such a reduction $r-R$ as a $3 \cdot r + 2$-tuple of algorithms $r-R = (R_0, (R_{l,1}, R_{l,2}, R_{l,3})_{l \in [r]}, R_3)$. Let now $A = (A_1, A_2)$ be an attacker against the UF-SMA-security of SIG. From these algorithms we construct an algorithm $r-R^A$ that solves a NICA $N$ as depicted in Figure 3.2. Let us explain Figure 3.2 here.

$R_0$. $r-R$ inputs a challenge $c$ of the considered non-interactive complexity assumption and random coins $\rho_R$. It processes these inputs by running $R_0$ which outputs a state $st_R$.

$R_l = (R_{l,1}, R_{l,2}, R_{l,3})$. Now, for each $l \in [r]$, we have a triplet of algorithms $R_l = (R_{l,1}, R_{l,2}, R_{l,3})$ that has black box access to attacker $A = (A_1, A_2)$. Note that the state $st_R$ may be passed over from $R_{l,3}$ to $R_{l+1,1}$ (and $R_3$) while the state
st, of A2 may not be passed over to the next execution of A1.

Rl,1. Rl,1 inputs the current state stRl,1 and outputs a public key vk, distinct messages ml, i ∈ [n], a random tape ρA for A1 and a state stRl,2. Next, A1 is run on input (vk, (mi)i∈[n]; ρA) and returns a state stA and an index j*l.

Rl,2. On input index j*l and state stRl,2, Rl,2 returns signatures (σl)i∈[n\{j*]} and state stRl,2. Now, A2 is run on ((σl)i∈[n\{j*}], stA) and returns σl,j*.

Rl,3. Rl,3 inputs the signature output by A2 and the current state stRl,2. It returns the state stRl+1,1.

R3. Finally, R3 inputs the current state of r-R and returns s. r-R is considered successful if Vfy(e, w, s) = 1.

Definition 3.4. We say that an algorithm r-R = (R0, (Rl,1, Rl,2, Rl,3), R3) is an r-simple (tR, n, εR, εA)-reduction from breaking N = (Gen, Vfy, Triv) to breaking the UF-SMA-security of SIG, if for any attacker A that (tA, n, εA)-breaks the UF-SMA security of SIG, algorithm r-RA (as constructed above) (tR + r · tA, εR)-breaks N.

Definition 3.5. Let ℓ : N → N. We say that an r-simple reduction R from breaking a non-interactive complexity assumption N to breaking the UF-SMA security of a signature scheme SIG loses ℓ if there exists an adversary A that (tA, n, εA)-breaks SIG such that RA (tR + r · tA, εR)-breaks N where

\[
\frac{t_R(\kappa) + r \cdot t_A(\kappa)}{\epsilon_R} \geq \ell(\kappa) \cdot \frac{t_A(\kappa)}{\epsilon_A(\kappa)}
\]

Remark 3.4. We note that in the introduction, we introduced the loss as a function of the security parameter κ and the deployment parameters π. However, since these usually depend on κ, we will consider the loss simply as a function of κ.
According to Definition 3.4 the loss relates the work factor of attacker $A$ to the work factor of algorithm $R^A$, which allows us to measure the tightness of a cryptographic reduction. The smaller $\ell$, the tighter is the reduction.

### 3.1.2 Lower Tightness Bound for $r$-simple reductions from any NICA to breaking UF-SMA-security

**Theorem 1.** Let $N = (\text{Gen}, \text{Vfy}, \text{Triv})$ be a non-interactive complexity assumption, $n, r \in \text{poly}$ and let $\text{SIG}$ be a signature scheme. Then for any $r$-simple $(t_\Lambda, n, \epsilon_\Lambda, 1)$-reduction $r$-$\Lambda$ from breaking $N$ to breaking the UF-SMA-security of $\text{SIG}$ there exists an algorithm $B$ that $(t_B, \epsilon_B)$-breaks $N$ where

$$t_B \leq r \cdot n \cdot t_\Lambda + r \cdot n \cdot (n - 1) \cdot t_{\text{Vfy}} + r \cdot t_{\text{ReRand}}$$

$$\epsilon_B \geq \epsilon_\Lambda - \frac{r}{n}$$

Here, $t_{\text{ReRand}}$ is the time to re-randomize a given valid signature over a message and $t_{\text{Vfy}}$ is the time needed to run the verification algorithm of $\text{SIG}$.

**Proof.** Our proof structure follows the structure of [HJK12] (which is also used in [LW14]). That is, it proceeds in two steps.

1. In a first step, we describe a hypothetical adversary $A$ that $(t_A, \epsilon_A)$-breaks the UF-SMA-security of $\text{SIG}$. We call this adversary hypothetical, because we do not know how to instantiate it efficiently.

2. In the second step, we show that we can construct an algorithm $B$ which simulates $A$ for $r$-$\Lambda$. The claim then follows from the analysis of the running time and the success probability of $B$.

**The Hypothetical Adversary.** The hypothetical adversary $A = (A_1, A_2)$ consists of two procedures that work as follows.
$A_1(vk, (m_i)_{i \in [n]}; \rho_A)$. On input a public key $vk$, messages $m_1, \ldots, m_n$ and random tape $\rho_A$, $A_1$ samples $j \leftarrow \$ [n]$ uniformly random and outputs $(j, st)$, with $st = (vk, (m_i)_{i \in [n]}, j)$.

$A_2((\sigma_i)_{i \in [n \setminus j]}, st)$. $A_2$ checks whether $\text{SIG.Vfy}(vk, m_i, \sigma_i) = 1$ for all $i \in [n \setminus j]$. If this holds, then it samples a uniformly random signature $\sigma_j \leftarrow \$ \Sigma(vk, m_j)$ for $m_j$. Otherwise, it sets $\sigma_j \leftarrow \perp$. Finally, it outputs $\sigma_j$.

Note that $A (t_A, 1)$-breaks the UF-SMA-security of $\text{SIG}$. By definition 3.5 this suffices to rule out a tight reduction. Note also that the second step of this adversary may not be efficiently computable, which is why we call this adversary hypothetical.

We will argue that, in any round $l \in [r]$, once $r$-$\Lambda$ has called $A_1$ and thus has “committed” to a verification key $vk$ and messages $m_1, \ldots, m_n$, there can only be a single choice of $j^*$ for which $r$-$\Lambda$ is able to output valid signatures $\sigma_i$ for all $i \neq j^*$. We prove this by contradiction, by showing essentially that if $r$-$\Lambda$ is successful for two distinct choices of $j^*$ (or if there is no such $j^*$) we can simulate $A$ efficiently and thus $r$-$\Lambda$ can be used to construct an algorithm breaking the underlying security assumption directly, without interaction with an adversary.

Essentially, by “rewinding” $r$-$\Lambda$ $n$ times to the point after it has output the verification key and the messages, $B$ will search for signatures on all messages. If it has found valid signatures on all messages (which will be the case if $r$-$\Lambda$ is successful for two distinct choices of $j^*$) then $B$ is able to simulate $A$ perfectly. Let us now proceed with the description of $B$.

**Simulating $A$ $r$ times sequentially.** We describe an algorithm $B$ that runs algorithm $r$-$\Lambda = (\Lambda_0, (\Lambda_{l,1}, \Lambda_{l,2}, \Lambda_{l,3}), \Lambda_3)$ as a subroutine. Recall that the goal of $B$ is to break $N$ and that $B$ is called on input $c \leftarrow \$ \text{Gen}(1^\kappa)$ and random tape $\rho$. $B$ proceeds as follows:

i. Run $st_{\Lambda_{l,1}} \leftarrow \Lambda_0(c, \rho_A)$ for uniformly $\rho_A$. If $\Lambda_0$ does not output $st_{\Lambda_{l,1}}$, $B$ aborts. Note that, since the input to $\Lambda_0$ is fixed (including random coins $\rho_A$), we may view $\Lambda_0$ (and all following subroutines of $\Lambda$) as deterministic.
iii. Finally, $B$ performs the following steps.

**Round** $l$. Initialize an array $A^l$ that has $n$ entries, all initialized to $\emptyset$.

a) Run $\Lambda_{l,1}(st_{\Lambda,l,1})$. If $r-\Lambda$ outputs $\left( vk^l, (m^l_i)_{i \in [n]}, \rho_A, st_{\Lambda,l,2} \right)$, continue. Otherwise stop.

b) Next, run $\left( (\sigma^l_{i,j})_{i \in [n \backslash j]}, st_{\Lambda,l,3} \right) \leftarrow \Lambda_{l,2}(j, st_{\Lambda,l,2})$ for each $j \in [n]$. In case all signatures returned by $\Lambda_{l,2}$ are valid, i.e., if

$$\bigwedge_{i \in [n \backslash j]} \text{SIG.Vfy}(vk^l, m^l_i, \sigma^l_{i,j}) = 1$$

set $A^l[i] \leftarrow \sigma^l_{i,j}$ for all $i \in [n \backslash j]$. Here, $\sigma^l_{i,j}$ refers to the signature over $m_i$ that is returned, when $\Lambda_{l,2}$ is called on index $j$ (and the current state).

c) Sample $j^{l*} \leftarrow \$ [n]. We distinguish between two cases:

1) Set $\sigma^* \leftarrow \bot$ if for any $i \in [n \backslash j^{l*}]$ it holds that

$$\text{SIG.Vfy}(vk^l, m^l_i, \sigma^l_{i,j^{l*}}) \neq 1.$$ 

2) Otherwise, i.e., if $\text{SIG.Vfy}(vk^l, m^l_i, \sigma^l_{i,j^{l*}}) = 1$ for all $i \in [n \backslash j^{l*}]$ set $\sigma^* \leftarrow \$ \text{SIG.ReRand}(vk^l, m^l_{j^{l*}}, A^l[j^{l*}])$.

d) Run $st_{\Lambda_{l+1,1}} \leftarrow \Lambda_{l,3}(\sigma^*, st_{\Lambda,l,3})$.

iii. Finally, $B$ runs $s \leftarrow \Lambda_3(st_{\Lambda_{r+1,1}})$ and returns $s$.

**Analysis.** Let us first note that the running time of $B$ is essentially bounded by the running time of $r-\Lambda$ plus the running time to execute $\Lambda_{l,2}$ $n - 1$ times, the time to run $\text{SIG.Vfy}$ for $n \cdot (n - 1)$ times and the time to run $\text{SIG.ReRand}$ for all $1 \leq l \leq r$. We thus obtain an upper bound on the total running time $t$ of $\Lambda$ by

$$t \leq r \cdot n \cdot t_\Lambda + r \cdot n \cdot (n - 1) \cdot t_{\text{Vfy}} + r \cdot t_{\text{ReRand}}.$$
Success probability of $B$. We define events $\text{bad}[l]$, $l \in [r]$, to analyze the success probability of $B$. Informally, $\text{bad}[l]$ occurs if, given $st_{\Lambda_{l,2}}$, $j^*_{l}$ is the only value such that $\Lambda_{l,2}(j, st_{\Lambda_{l,2}})$ outputs signatures that are all valid (note that $vk^l$, as well as $m^l_1, \ldots, m^l_n$ are determined once $\Lambda_{l,1}$ outputs $st_{\Lambda_{l,2}}$). Let us denote by $st_{\Lambda_{l,2}}$ the unique state computed by $\Lambda_{l,1}$ and let $j^*_{l}$ denote the unique value input to $\Lambda_{l,3}(\sigma^*, j^*_{l}, st_{\Lambda_{l,3}})$. Since $\Lambda_{l,1}$ and $\Lambda_{l,3}$ are called only once during the simulation $st_{\Lambda_{l,2}}$ as well as $j^*_{l}$ are well-defined in both experiments $\text{NICA}_N^B(1^k)$ and $\text{NICA}_N^A(1^k)$. We say that $\text{bad}[l]$ occurs if $\text{pred}(st_{\Lambda_{l,2}}, j^*_{l}) = 1 \land \text{pred}(st_{\Lambda_{l,2}}, j) = 0 \ \forall j \in [n \setminus j^*_{l}]$ where

\[
\text{pred}(st_{\Lambda_{l,2}}, j) = 1 \\
\Leftrightarrow \bigwedge_{i \in [n \setminus j]} \text{SIG}.\text{Vfy}(vk^l, m^l_i, \sigma^l_i) = 1 : \big(\sigma^l_i\big)_{i \in [n \setminus j], st_{\Lambda_{l,3}}} \leftarrow \Lambda_{l,2}(st_{\Lambda_{l,2}}, j)
\]

Now, we define event $\text{bad} = \bigvee_{l \in [r]} \text{bad}[l]$.

Let us denote by $S(F)$ the event $\text{NICA}_N(F, \kappa) \Rightarrow 1$. Then, following Shoup’s Difference Lemma [Sho04], it holds that

\[
\Pr[S(r-\Lambda^A)] - \Pr[S(B)] \\
\leq \Pr[S(r-\Lambda^A) \cap \neg \text{bad}] - \Pr[S(B) \cap \neg \text{bad}] + \Pr[\text{bad}]
\] (3.2)

Bounding $\Pr[\text{bad}]$. Recall that event $\text{bad}$ occurs only if there exists any $l \in [r]$ such that $\text{bad}[l]$ occurs, i.e.,

\[
\text{pred}(st_{\Lambda_{l,2}}, j^*_{l}) = 1 \land \text{pred}(st_{\Lambda_{l,2}}, j) = 0 \ \forall j \in [n \setminus j^*_{l}]
\] (3.3)

where $st_{\Lambda_{l,2}}$ is the value computed by $\Lambda_{l,1}(st_{\Lambda_{l,1}})$, and $j^*_{l}$ is the value given as input to $\Lambda_{l,3}(\sigma^*, j^*_{l}, st_{\Lambda_{l,3}})$. We claim that then we have $\Pr[\text{bad}[l]] \leq 1/n$. To see this, note first that for $\text{bad}[l]$ to occur it is necessary that there is only one value $j^*_{l}$ that satisfies $\text{pred}(st_{\Lambda_{l,2}}, j^*_{l})$. Moreover, $\text{bad}[l]$ does only occur if this index $j^*_{l}$ is input to $\Lambda_{l,3}$. Note that both the hypothetical adversary $A$ and the adversary simulated by $B$ choose $j^*_{l}$ independently and uniformly random, which yields the claim. Now, by the union bound, we obtain

\[
\Pr[\text{bad}] \leq r/n
\] (3.4)
3.1 Warm-Up: Improved Bound for Digital Signatures

Pr[S(B) ∩ ¬bad] = Pr[S(Λ^A) ∩ ¬bad]. Note that B executes in particular:

1) \( st_{\Lambda_1,1} \leftarrow \Lambda_0(c; \rho_\Lambda) \)

2a) \((vk^l, (m_i^l)_{i \in [n]}, st_{\Lambda_2,1}) \leftarrow \$ \Lambda_{l,1}(st_{\Lambda_1,1})\)

2b) \(((\sigma_{i,j^*})_{i \in [n \setminus j^*]}, st_{\Lambda_2,3}) \leftarrow \$ \Lambda_{l,2}(j^*, st_{\Lambda_2,1})\)

2c) \( st_{\Lambda_{l+1,1}} \leftarrow \Lambda_{l,3}(\sigma^*, j^*, st_{\Lambda_{l,3}})\)

3) \( s \leftarrow \Lambda_3(st_{\Lambda_{r+1,1}})\)

where steps 2a) - 2c) are carried out for each \( l \in [r] \). We show that if bad[^l] does not occur, then B simulates the hypothetical adversary A perfectly for \( \Lambda_l = (\Lambda_{l,1}, \Lambda_{l,2}, \Lambda_{l,3}) \). To this end, consider the distribution of \( \sigma^* \) that is input to \( \Lambda_{l,3} \) in following two cases.

1. Algorithm \( \Lambda_{l,2}(j^*, st_{\Lambda_{l,2}}) \) outputs \(((\sigma_{i,j^*})_{i \in [n \setminus j^*]}, st_{\Lambda_{l,3}})\) such that there exists an index \( i \in [n \setminus j^*] \) for which SIG.Vfy(vk^l, m_i^l, \sigma_{i,j^*}) \neq 1. In this case, A would compute \( \sigma^* := \bot \) which is also output by B.

2. Algorithm \( \Lambda_{l,2}(j^*, st_{\Lambda_{l,2}}) \) outputs \(((\sigma_{i,j^*})_{i \in [n \setminus j^*]}, st_{\Lambda_{l,3}})\) such that

\[
\text{SIG.Vfy(vk^l, m_i^l, \sigma_{i,j^*}) = 1}
\]

for all \( i \in [n \setminus j^*] \). In this case, A would output a uniformly random signature \( \sigma^* \leftarrow \$ \Sigma(vk^l, m_i^l_{j^*}) \). Note that in this case B outputs a re-randomized signature \( \sigma^* \leftarrow \$ \text{SIG.ReRand}(vk^l, m_i^l_{j^*}, \Lambda_l[j^*]) \), which is a uniformly distributed valid signature for \( m_i^l_{j^*} \) provided that \( \Lambda_l[j^*] \neq \emptyset \). The latter happens whenever bad[^l] does not occur.

Thus, B simulates A perfectly for \( \Lambda \) provided that ¬bad. This implies \( S(B) \cap ¬bad \iff S(R\Lambda) \cap ¬bad \), which yields

\[
\text{Pr}[S(B) \cap ¬bad] = \text{Pr}[S(R\Lambda^A) \cap ¬bad] \quad (3.5)
\]
Finishing the proof. By plugging (3.4) and (3.5) into Inequality (3.2), we obtain

\[ \Pr[S(r-A^A)] - \Pr[S(B)] \leq \frac{r}{n} \Rightarrow \epsilon_B \geq \epsilon_R - \frac{r}{n} \]

\[ \square \]

3.1.3 Interpretation

Assuming that no adversary \( B \) is able to \((t_N, \epsilon_N)\)-break the security of \( \text{NICA} \) with \( t_N = t_B = r \cdot n \cdot t_A + r \cdot n \cdot (n-1) \cdot t_{\text{Vfy}} + r \cdot t_{\text{ReRand}} \), we must have \( \epsilon_B \leq \epsilon_N \). By Theorem 1, we thus must have

\[ \epsilon_A \leq \epsilon_B + \frac{r}{n} \leq \epsilon_N + \frac{r}{n} \]

for all reductions \( \Lambda \). In particular, the hypothetical adversary \( A \) constructed above is an example of an adversary such that

\[ \frac{t_A + r \cdot t_A}{\epsilon_A} \geq \frac{r \cdot t_A}{\epsilon_N + \frac{r}{n}} = (\epsilon_N + \frac{r}{n})^{-1} \cdot \frac{r \cdot t_A}{1} = (\epsilon_N + \frac{r}{n})^{-1} \cdot r \cdot \frac{t_A}{\epsilon_A} \]

Thus, any reduction \( \Lambda \) from breaking the security of \( \text{NICA} N \) to breaking the UF-SMA-security (or any stronger security notion, like EFU-CMA-security, cf. Definition 2.11) of signature scheme \( \text{SIG} \) that provides efficient signature re-randomization loses a factor that is at least linear in the number \( n \) of sign queries issued by the attacker, or \( N \) is easy to solve.

Theorem 2 (informal). Any \( r \)-simple reduction from breaking the security of \( \text{NICA} N \) to breaking the UF-SMA-security (or any stronger security notion, like EUF-CMA-security, cf. Definition 2.11) of signature scheme \( \text{SIG} \) that provides efficient signature re-randomization loses a factor that is at least linear in the number \( n \) of sign queries issued by the attacker, or \( N \) is easy to solve.

Theorem 2 follows from the fact that UF-SMA-security is strictly weaker than EUF-CMA-security [BJLS15].

Remark 3.5. As mentioned above, a unique signature scheme is trivially efficiently re-randomizable. Thus, Theorem 2 applies also to unique signature schemes.
3.2 Sufficient Conditions to Rule out Tight Reductions

In this section we state and prove our main impossibility result with respect to tight reductions which generalizes the results from Section 3.1. Essentially, we observe that for the proof to work we do not need all structural elements a signature scheme possesses. In particular we do not require dedicated parameter generation-, key generation- and sign-algorithms. Instead, we consider an abstract security experiment with the following properties:

1. The values that are publicly available “induce a relation” $R(x, y)$ that is efficiently computable for the adversary during the security experiment.

2. The adversary is provided with statements $y_1, \ldots, y_n$ at the beginning of the security experiment and has access to an oracle that when queried $y_i$ returns $x_i$ such that $R(x_i, y_i), i \in [n]$.

3. If the adversary is able to output $x_j$ such that $R(x_j, y_j)$ and it did not query its oracle on $y_j$, this is sufficient to win the security game.

To show the usefulness of such an abstract experiment, we note that, for instance, the security experiments for public key encryption or key encapsulation mechanisms in the multi-user setting with corruptions [BHJ+15], or digital signature schemes in the multi-user (MU) setting with corruptions [Bad14, BHJ+15], naturally satisfy these properties as follows [BJLS15]. Essentially, we define a relation $R(sk, pk)$ over pairs of public keys and secret keys such that $R(sk, pk) = 1$ whenever $sk$ “matches“ $pk$. The adversary is provided with public keys at the beginning of the experiment, and is able to obtain secret keys corresponding to public keys of its choice. Finally, if the adversary is able to output an uncorrupted secret key, it is clearly able to compute a signature over a message that was not signed before (i.e., winning the signature security game) or decrypt the challenge ciphertext (i.e., winning the PKE/KEM seurity game). Thus,
3 On the Impossibility of Tight Reductions

all three points are satisfied. Details on how to apply the result
to, e.g., PKE in the multi user setting with corruptions we refer
to Section 3.3.

3.2.1 Definitions

Re-randomizable relations. Let $R \subseteq X \times Y$ be a relation. For
$(x, y)$ with $R(x, y) = 1$ we call $x$ the *witness* and $y$ the *statement*.
We use $X(R, y)$ to denote the set

$$X(R, y) := \{x : R(x, y) = 1\}$$

of all witnesses $x$ for statement $y$ with respect to $R$. We denote by
$L(R) := \{y : \exists x \text{ s.t. } R(x, y) = 1\} \subseteq Y$ the language consisting
of statements in $R$.

In the sequel we will consider *computable* relations. We will
therefore identify a relation $R$ with an algorithm $\hat{R}$ that com-
putes $R$. We say that a relation $R$ is $t_{\text{Vfy}}$-computable, if there is a
deterministic algorithm $\hat{R}$ that runs in time at most $t_{\text{Vfy}}(|x| + |y|)$
such that $\hat{R}(x, y) = R(x, y)$.

**Definition 3.6.** Let $\mathcal{R} := \{R_i\}_{i \in I}$ be a family of computable
relations. We say that $\mathcal{R}$ is $t_{\text{ReRand}}$-re-randomizable if there is a
(possibly probabilistic) algorithm $\mathcal{R}.\text{ReRand}$ that inputs $(\hat{R}_i, y, x)$,
runs in time at most $t_{\text{ReRand}}$, and outputs $x'$ which is uniformly
distributed over $X(R, y_i)$ whenever $R_i(x, y) = 1$, with probability
$1$.

**Example 3.1.** Digital signatures in the single user setting, as
considered in Section 3.1, may be described in terms of families
of relations. We set $R_{\Pi,\nu k}$ to the relation over signatures and
messages that is defined by a verification key $\nu k$. In this case, we
have that $X(R, y) = \Sigma(\nu k, y)$ is the set of all valid signatures over
message $y$ with respect to public key $\nu k$. Note that the family of
relations $(R_{\Pi,\nu k})_{\Pi,\nu k}$ is $t_{\text{ReRand}}$-re-randomizable, if the signature
scheme is $t_{\text{ReRand}}$-re-randomizable (cf. Definition 3.2).

**Witness unforgeability under static statement attacks.** We
will consider a weak security experiment for computable rela-
tions, which is inspired by the UF-SMA-security experiment con-
sidered in Section 3.1 but abstract and general enough to be
3.2 Sufficient Conditions to Rule out Tight Reductions

<table>
<thead>
<tr>
<th>Game UF-SSA$_R^n$ $(\mathcal{A}, \kappa)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R = R_i \leftarrow $ \mathcal{R}; \rho_A \leftarrow $ {0, 1}^\kappa$</td>
</tr>
<tr>
<td>$y_1, \ldots, y_n \leftarrow $ L(R)$ s.t. $y_i \neq y_j$ for all $i \neq j$</td>
</tr>
<tr>
<td>$x_i \leftarrow $ X(R, y_i)$ for all $i \in [n]$</td>
</tr>
<tr>
<td>$(j, st) \leftarrow \mathcal{A}<em>1(\tilde{R}, (y_i)</em>{i \in [n]}; \rho_A)$</td>
</tr>
<tr>
<td>$x_j \leftarrow \mathcal{A}<em>2(st, (x_i)</em>{i \in [n \setminus j]})$</td>
</tr>
<tr>
<td>return $R(x_j, y_j)$</td>
</tr>
</tbody>
</table>

Figure 3.3: The UF-SSA-security Game with attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$.

applicable in other useful settings. Jumping slightly ahead, we will show in Section 3.3 that this includes applications to public-key encryption or key encapsulation mechanisms in the multi-user setting with corruptions. In a similar way, the result can be applied to signatures in the multi-user setting with corruptions and also to non-interactive key exchange [BJLS15].

The security experiment is described in Figure 3.3. It is parameterized by a family $\mathcal{R}$ of computable relations, $\mathcal{R} = \{R_i\}_{i \in I}$, and the number $n$ of statements the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is provided with. These statements need to be pairwise distinct. $\mathcal{A}$ may non-adaptively ask for witnesses for all but one statement, and is considered successful if it manages to output a “valid” witness for the remaining statement.

**Definition 3.7.** Let $\mathcal{R} = \{R_i\}_{i \in I}$ be a family of computable relations. We say that an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ $(t, n, \epsilon)$-breaks the witness unforgeability under static statement attacks of $\mathcal{R}$ if it runs in time $t$ and

$$\Pr \left[ \text{UF-SSA}_{\mathcal{R}}^n (\mathcal{A}, \kappa) \Rightarrow 1 \right] \geq \epsilon$$

where $\text{UF-SSA}_{\mathcal{R}}^n (\mathcal{A}, \kappa)$ is the security game that is depicted in Figure 3.3.

**Simple reductions from non-interactive complexity assumptions to breaking UF-SSA-security.** Informally, a reduction
from breaking the UF-SSA-security of a family of relations \( R \) to breaking the security of a non-interactive complexity assumption \( N \) is an algorithm \( r-\Gamma \), which turns an attacker \( A = (A_1, A_2) \) against \( R \) according to Definition 3.7 into an algorithm \( r-\Gamma^A \) that breaks \( N \) according to Definition 2.8. As in Section 3.1, we will only consider simple reductions, i.e., reductions that have black-box access to the attacker and that may run the attacker at most \( r \) times sequentially.

We define a reduction from breaking the security of \( R \) to breaking \( N \) as a \((3r + 2)\)-tuple of algorithms \( r-\Gamma = (\Gamma_0, (\Gamma_{l,1}, \Gamma_{l,2}, \Gamma_{l,3})_{l \in [r]}, \Gamma_3)\), which turn an algorithm \( A \) breaking the security of \( R \) into an algorithm \( r-\Gamma^A \) breaking \( N \), as described in Figure 3.4. Note that this algorithm works almost identical to that considered in Section 3.1.1 except that we consider a more general class of relations.

**Definition 3.8.** We say that an algorithm \( r-\Gamma = (\Gamma_0, (\Gamma_{l,1}, \Gamma_{l,2}, \Gamma_{l,3})_{l \in [r]}, \Gamma_3) \) is an \( r \)-simple \((t_\Gamma, n, \epsilon_\Gamma, \epsilon_A)\)-reduction from breaking \( N = (\text{Gen}, \text{Vfy}, \text{Triv}) \) to breaking the UF-SSA-security of a family of relations \( R \), if for any algorithm \( A \) that \((t_A, n, \epsilon_A)\)-breaks the UF-SSA security of \( R \), algorithm \( r-\Gamma^A \) (cf. Figure 3.4) \((t_\Gamma + r \cdot t_A, \epsilon_\Gamma)\)-breaks \( N \).

We define the los of an \( r \)-simple reduction \( r-\Gamma \) from breaking \( N \) to breaking the UF-SSA-security of a family of computable relations \( R \) similar to Definition 3.5.

### 3.2.2 Main Theorem to Rule Out Tight Reductions

In this Section we establish the following result that generalizes Theorem 1.

**Theorem 3.** Let \( N = (\text{Gen}, \text{Vfy}, \text{Triv}) \) be a non-interactive complexity assumption, \( n, r \in \text{poly} \) and let \( R \) be a family of computable relations. Then for any \( r \)-simple \((t_\Gamma, n, \epsilon_\Gamma, 1)\)-reduction
Algorithm \( r-\Gamma^A(c, \rho_\Gamma) \)
\[ st_{\Gamma_{1,1}} \leftarrow \Gamma_0(c, \rho_\Gamma) \]
\[ \text{for } 1 \leq l \leq r \text{ do:} \]
\[ \left( \widehat{R}_l, (y_i^l)_{i \in [n]} ; \rho_\mathcal{A}, st_{\Gamma_{l,2}} \right) \leftarrow \Gamma_{l,1}(st_{\Gamma_{l,1}}) \]
\[ (j^l, st_\mathcal{A}) \leftarrow A_1 \left( \widehat{R}_l, (y_i^l)_{i \in [n]} ; \rho_\mathcal{A} \right) \]
\[ \left( (x_i^l)_{i \in [n \setminus j^l]} ; st_{\Gamma_{l,3}} \right) \leftarrow \Gamma_{l,2} \left( j^l, st_{\Gamma_{l,2}} \right) \]
\[ x_j^l \leftarrow A_2 \left( (x_i^l)_{i \in [n \setminus j^l]} ; st_\mathcal{A} \right) \]
\[ st_{\Gamma_{l+1,1}} \leftarrow \Gamma_{l,3} \left( x_j^l, st_{\Gamma_{l,3}} \right) \]
\[ s \leftarrow \Gamma_3 \left( st_{\Gamma_{r+1,1}} \right) \]
\[ \text{return } s \]

Figure 3.4: Algorithm \( r-\Gamma^A \) that solves a non-interactive complexity assumption according to Definition 2.8, constructed from a \( r \)-simple reduction \( r-\Gamma = \left( \Gamma_0, \left( \Gamma_{l,1}, \Gamma_{l,2}, \Gamma_{l,3} \right)_{l \in [r]}, \Gamma_3 \right) \) and an attacker \( \mathcal{A} = (A_1, A_2) \).

\( r-\Gamma \) from breaking \( N \) to breaking the UF-SSA-security of \( \mathcal{R} \) there exists an algorithm \( \mathcal{B} \) that \( (t_\mathcal{B}, \epsilon_\mathcal{B}) \)-breaks \( N \) where
\[ t_\mathcal{B} \leq r \cdot n \cdot t_\Gamma + r \cdot n \cdot (n - 1) \cdot t_{\mathcal{Vfy}} + r \cdot t_{\text{ReRand}} \]
\[ \epsilon_\mathcal{B} \geq \epsilon_\Gamma - \frac{r}{n} \]

Here, \( t_{\text{ReRand}} \) is the time to re-randomize a given valid witness and \( t_{\mathcal{Vfy}} \) is the maximum time needed to compute \( R \in \mathcal{R} \).

Proof. The proof of Theorem 3 is nearly identical to the proof of Theorem 1. For reasons of completeness, we include it here. As in the above proof, we first describe a hypothetical attacker that \( (t_\mathcal{A}, \epsilon_\mathcal{A}) \)-breaks the UF-SSA-security of a family of computable relations. Next, we show how to construct an algorithm \( \mathcal{B} \) that simulates \( \mathcal{A} \) for \( r-\Gamma \) and conclude by analyzing \( \mathcal{B} \). The reasoning behind the steps below is explained in the proof of Theorem 1.
3 On the Impossibility of Tight Reductions

**The Hypothetical Adversary.** Similar to the proof of Theorem 1, the hypothetical adversary \( \mathcal{A} = (A_1, A_2) \) works as follows.

\( A_1 \left( \hat{R}, (y_i)_{i \in [n]}; \rho_A \right) \). On input a description of a relation, statements \( y_1, \ldots, y_n \) and random tape \( \rho_A \), \( A_1 \) samples \( j \leftarrow \$ [n] \) and outputs \((j, st)\), where \( st = (\hat{R}, (y_i)_{i \in [n]}, j) \).

\( A_2 \left( st, (x_i)_{i \in [n \setminus j]} \right) \). \( A_2 \) checks whether \( R(x_i, y_i) = 1 \) for all \( i \in [n \setminus j] \). If this holds, then it samples a uniformly random witness \( x_j \leftarrow \$ X(R, y_j) \) for \( y_j \). Otherwise, it sets \( x_j \leftarrow \bot \). Finally, it outputs \( x_j \).

**Simulating \( A \) \( r \) times sequentially.** We describe an algorithm \( \mathcal{B} \) that runs algorithm \( r-\Gamma = \left( \Gamma_0, (\Gamma_{l,1}, \Gamma_{l,2}, \Gamma_{l,3})_{l \in [r]}, \Gamma_3 \right) \) as a subroutine. Recall that the goal of \( \mathcal{B} \) is to break \( N \) and that \( \mathcal{B} \) is called on input \( c \leftarrow \$ \text{Gen}(1^\kappa) \) and random tape \( \rho \). \( \mathcal{B} \) proceeds as follows:

i. Run \( st_{\Gamma_{1,1}} \leftarrow \Gamma_0(c, \rho_\Gamma) \) for uniformly \( \rho_\Gamma \). If \( \Gamma_0 \) does not output \( st_{\Lambda_{1,1}} \), \( \mathcal{B} \) aborts. Note that, since the input to \( \Gamma_0 \) is fixed (including random coins \( \rho_\Gamma \)), we may view \( \Gamma_0 \) (and all following subroutines of \( \Gamma \)) as deterministic.

ii. For \( 1 \leq l \leq r \), \( \mathcal{B} \) performs the following steps.

**Round \( l \).** Initialize an array \( A^l \) that has \( n \) entries, all initialized to \( \emptyset \).

- **a)** Run \( \Gamma_{l,1}(st_{\Gamma_{1,1}}) \). If this algorithm outputs \( \left( R^l, (y_i^l)_{i \in [n]}, \rho_A, st_{\Gamma_{l,2}} \right) \), continue. Otherwise stop.

- **b)** Next, run \( \left( (x_i^l)_{i \in [n \setminus j]}, st_{\Gamma_{l,3}} \right) \leftarrow \Gamma_{l,2}(j, st_{\Gamma_{l,2}}) \) for each \( j \in [n] \). In case all witnesses returned by \( \Gamma_{l,2} \) are valid, i.e., if \( \bigwedge_{i \in [n \setminus j]} R^l(x_i^l, y_{i,j}^l) = 1 \)
set \( A_l[i] \leftarrow x_{i,j}^l \) for all \( i \in [n \setminus j] \). Here, \( x_{i,j}^l \) refers to the witness for statement \( y_i \) that is returned, when \( \Gamma_{l,2} \) is called on index \( j \) (and the respective state).

c) Sample \( j^* \leftarrow \$ [n] \). We distinguish between two cases:

1) Set \( x^* \leftarrow \perp \) if for any \( i \in [n \setminus j^*] \) it holds that \( R^l(x_{i,j^*}^l, y_i^l) \neq 1 \).

2) Otherwise, i.e., if \( R^l(x_{i,j^*}^l, y_i^l) = 1 \) for all \( i \in [n \setminus j^*] \), set \( x^* \leftarrow \$ R.\text{ReRand}(\hat{R}^l, y_{j^*}^l, A_l[j^*]) \).

d) Run \( s_{\Gamma_{l+1,1}} \leftarrow \Gamma_{l,3}(x^*, s_{\Gamma_{l,3}}) \).

Analysis. Let us first note that the running time of \( B \) is essentially bounded by the running time of \( r-\Gamma \) plus the running time to execute \( \Gamma_{l,2} \) \( n-1 \) times, the time to compute \( R \) for \( n \cdot (n-1) \) times and the time to run \( \text{R.\text{ReRand}} \) for all \( 1 \leq l \leq r \). We thus obtain an upper bound on the total running time \( t \) of \( \Gamma \) by

\[
t \leq r \cdot n \cdot t_\Gamma + r \cdot n \cdot (n-1) \cdot t_{\text{Vfy}} + r \cdot t_{\text{ReRand}} .
\]

Success probability of \( B \). We define events \( \text{bad}[l], l \in [r] \), to analyze the success probability of \( B \). Informally, \( \text{bad}[l] \) occurs if, given \( s_{\Gamma_{l,2}} \), \( j^* \) is the only value such that \( \Gamma_{l,2}(j, s_{\Gamma_{l,2}}) \) outputs witnesses that are all valid (note that \( R^l \), as well as \( y_1^l, \ldots, y_n^l \) are determined once \( \Gamma_{l,1} \) outputs \( s_{\Gamma_{l,2}} \)). Let us denote by \( s_{\Gamma_{l,2}} \) the unique state computed by \( \Gamma_{l,1} \) and let \( j^* \) denote the unique value input to \( \Gamma_{l,3} (x^*, j^*, s_{\Gamma_{l,3}}) \). Since \( \Gamma_{l,1} \) and \( \Gamma_{l,3} \) are called only once during the simulation, \( s_{\Gamma_{l,2}} \) as well as \( j^* \) are well-defined in both experiments \( \text{NICA}_N^B(1^k) \) and \( \text{NICA}_N^A(1^k) \). We say that \( \text{bad}[l] \) occurs if \( \text{pred}(s_{\Gamma_{l,2}}, j^*) = 1 \land \text{pred}(s_{\Gamma_{l,2}}, j) = 0 \forall j \in [n \setminus j^*] \) where

\[
\text{pred}(s_{\Gamma_{l,2}}, j) = 1 \Leftrightarrow \bigwedge_{i \in [n \setminus j]} R^l(x_{i,j}^l, y_i^l) = 1 : (x_{i,j}^l)_{i \in [n \setminus j], s_{\Gamma_{l,3}}} \leftarrow \Gamma_{l,2}(s_{\Gamma_{l,2}}, j)
\]

Now, we define event \( \text{bad} = \bigvee_{l \in [r]} \text{bad}[l] \).
Let us denote by $\mathsf{S}(\mathcal{F})$ the event $\mathsf{NICA}_N(\mathcal{F}, \kappa) \Rightarrow 1$. Then, following Shoup’s Difference Lemma [Sho04], it holds that

$$\Pr[\mathsf{S}(r, \Gamma^A)] - \Pr[\mathsf{S}(B)] \leq \Pr[\mathsf{S}(r, \Gamma^A) \cap \neg \text{bad}] - \Pr[\mathsf{S}(B) \cap \neg \text{bad}] + \Pr[\text{bad}]$$

(3.6)

**Bounding $\Pr[\text{bad}]$.** Recall that event bad occurs only if there exists any $l \in [r]$ such that bad[$l$] occurs, i.e.,

$$\text{pred}(st_{\Gamma^A, 1}, j^{*\Gamma^A}) = 1 \land \text{pred}(st_{\Gamma^A, 2}, j) = 0 \forall j \in [n] \setminus j^{*\Gamma^A}$$

(3.7)

where $st_{\Gamma^A, 2}$ is the value computed by $\Gamma^A_1(st_{\Gamma^A, 1})$, and $j^{*\Gamma^A}$ is the value given as input to $\Gamma^A_3(x^*, j^{*\Gamma^A}, st_{\Gamma^A, 3})$. We claim that then we have $\Pr[\text{bad}[l]] \leq 1/n$. To see this, note first that for bad[$l$] to occur it is necessary that there is only one value $j^{*\Gamma^A}$ that satisfies $\text{pred}(st_{\Gamma^A, 2}, j^{*\Gamma^A})$. Moreover, bad[$l$] does only occur if this index $j^{*\Gamma^A}$ is input to $\Gamma^A_3$. Note that both the hypothetical adversary $\mathcal{A}$ and the adversary simulated by $\mathcal{B}$ choose $j^{*\Gamma^A}$ independently and uniformly random, which yields the claim. Now, by the union bound, we obtain

$$\Pr[\text{bad}] \leq r/n$$

(3.8)

$\Pr[\mathsf{S}(\mathcal{B}) \cap \neg \text{bad}] = \Pr[\mathsf{S}(\Gamma^A) \cap \neg \text{bad}]$. Note that $\mathcal{B}$ executes in particular:

1) $st_{\Gamma^A, 1} \leftarrow \Gamma_0(c, \rho)$

2a) $(\tilde{R}^l, (y_i^l)_{i \in [n]}, st_{\Gamma^A, 2}) \leftarrow \Gamma_1(st_{\Gamma^A, 1})$

2b) $((x_i^l)_{i \in [n] \setminus j^{*\Gamma^A}}, st_{\Gamma^A, 3}) \leftarrow \Gamma_2(j^{*\Gamma^A}, st_{\Gamma^A, 2})$

2c) $st_{\Gamma^A, 1} \leftarrow \Gamma_3(x^{*\Gamma^A}, st_{\Gamma^A, 3})$

3) $s \leftarrow \Gamma_3(st_{\Gamma^A, 1})$

where steps 2a) - 2c) are carried out for each $l \in [r]$. We show that if bad[$l$] does not occur, then $\mathcal{B}$ simulates the hypothetical adversary $\mathcal{A}$ perfectly for $\Gamma_l = (\Gamma^A_1, \Gamma^A_2, \Gamma^A_3)$. To this end, consider the distribution of $x^*$ that is input to $\Gamma^A_3$ in following two cases.
1. $\Gamma_{l,2}(j^{l*}, st_{\Gamma_{l,3}})$ outputs $((x_{i,j^{l*}})_{i\in[n \setminus j^{l*}]}, st_{\Gamma_{l,3}})$ such that it exists an index $i \in [n \setminus j^{l*}]$ for which $R_{l}(x_{i,j^{l*}}, y_{i}^{l}) \neq 1$. In this case, $A$ would compute $x^{*} := \perp$ which is also output by $B$.

2. $\Gamma_{l,2}(j^{l*}, st_{\Gamma_{l,3}})$ outputs $((x_{i,j^{l*}})_{i\in[n \setminus j^{l*}]}, st_{\Gamma_{l,3}})$ such that $R_{l}(x_{i,j^{l*}}, y_{i}^{l}) = 1$ for all $i \in [n \setminus j^{l*}]$. In this case, $A$ would output a uniformly random witness $x^{*} \leftarrow X(R_{l}, y_{j^{l*}})$. We note that in this case $B$ outputs a re-randomized witness $x^{*} \leftarrow \mathcal{R}.\text{ReRand}(\hat{R}_{l}, y_{j^{l*}}, A[l^{j^{l*}}])$, which is valid witness for $y_{j^{l*}}$ that is uniformly distributed over all valid witnesses provided that $A[l^{j^{l*}}] \neq \emptyset$. The latter happens whenever $\text{bad}[l]$ does not occur.

Thus, $B$ simulates $A$ perfectly for $\Gamma$ provided that $\neg \text{bad}$. This implies $S(B) \cap \neg \text{bad} \iff S(RA) \cap \neg \text{bad}$, which yields

$$\Pr[S(B) \cap \neg \text{bad}] = \Pr[S(r-\Gamma A) \cap \neg \text{bad}] \quad (3.9)$$

**Finishing the proof.** By plugging (3.8) and (3.9) into Inequality (3.6), we obtain

$$\Pr[S(r-\Gamma A)] - \Pr[S(B)] \leq r/n \implies \epsilon_B \geq \epsilon_R - r/n$$

The interpretation of Theorem 3 is nearly identical to the interpretation described in Section 3.1.3. Assuming that no adversary $B$ is able to $(t_N, \epsilon_N)$-break the security of NICA with $t_N = t_B = r \cdot n \cdot t_{\Gamma} + r \cdot n \cdot (n - 1) \cdot t_{\text{Vfy}} + r \cdot t_{\text{ReRand}}$, we must have $\epsilon_B \leq \epsilon_N$. Thus, if $\mathcal{R}$ is efficiently computable and re-randomizable, the loss of any simple reduction from breaking $N$ to breaking the UF-SSA-security of $\mathcal{R}$ is at least linear in $n$.

### 3.3 Applications

We give an example of how to apply the above result to cryptographic primitives. Namely, we apply the result to public key
encryption in the multi-user setting with corruptions. Similarly, the technique can be applied to key encapsulation mechanisms or digital signatures in the multi-user setting with corruptions (cf. Definition 4.2). We refer to [BJLS15] for details on how the technique can be applied to, e.g., non-interactive key exchange.

MU-IND-CPA-C-security for public key encryption schemes is defined in Section 2.3.

**Definition 3.9.** We say that an algorithm \( \Gamma \) is an \( r \)-simple \((t_\Gamma, n, \mu, \epsilon_\Gamma, \epsilon_A)\)-reduction from breaking \( N = (\text{Gen}, \text{Vfy}, \text{Triv}) \) to breaking the MU-IND-CPA-C-security of \( \text{PKE} \), if algorithm \( \Gamma^A \) \((t_\Gamma + r \cdot t_A, \epsilon_\Gamma)\)-breaks \( N \) if algorithm \( A \) \((t_A, n, \mu, \epsilon_A)\)-breaks the MU-IND-CPA-C security of \( \text{PKE} \).

We define the loss of an \( r \)-simple reduction \( \Gamma \) from breaking \( N \) to breaking the MU-IND-CPA-C-security of \( \text{PKE} \) similar to definition 3.5.

**Remark 3.6.** For simplicity, we consider chosen-plaintext security in the multi-user setting. The result generalizes easily to other security notions for public-key encryption, like chosen-ciphertext security, for example.

**Additional properties.** To apply Theorem 3 we require an additional algorithm \( \text{SKCheck}_\Pi \) (where \( \Pi \leftarrow^\$ \text{Setup}(1^n) \)) with the following property:

\[
\text{SKCheck}_\Pi(pk, sk) = 1 \iff \Pr \left[ \text{Dec}(sk, \text{Enc}(pk, m)) = m : m \leftarrow^\$ \mathcal{M} \right] = 1
\]

That is, \( \text{SKCheck} \) takes inputs \( sk \) and \( pk \) and returns 1 if and only if \( pk \) is a \( \text{PKE} \) public key and \( sk \) is a secret key corresponding to public key \( pk \). Since we require perfect correctness for public key encryption schemes, we have \( \text{SKCheck}(pk, sk) = 1 \) whenever \((pk, sk) \leftarrow^\$ \text{Gen}(\Pi)\). We denote by \( t_{\text{Vfy}} \) the running time of \( \text{SKCheck} \) and call \( \text{PKE} \) \( t_{\text{Vfy}} \)-key checkable if \( \text{SKCheck} \) runs in time \( t_{\text{Vfy}} \).

**Definition 3.10** (Key re-randomization.). We say that a public key encryption scheme \( \text{PKE} \) is \( t_{\text{ReRand}} \)-key re-randomizable
if there exists an algorithm ReRand that runs in time at most \( t_{\text{ReRand}} \), takes as input \((\Pi, pk, sk)\) and returns \( sk \) uniformly distributed over \( \{sk : \text{SKCheck}_\Pi(vk, sk) = 1\} \) whenever it holds that \( \text{SKCheck}_\Pi(pk, sk) = 1 \).

**Example 3.2.** Let us consider linear encryption from \([BBS04]\).

Here, a public key is a tuple of group elements \( g, h, k \in G \) where \( G \) is a group of prime order \( p \). A corresponding secret key, as generated by the key generation algorithm, consists of \( \alpha, \beta \in \mathbb{Z}_p \) such that \( h = g^\alpha \) and \( k = g^\beta \). For further investigation we recall the encryption and decryption procedures. The encryption algorithm outputs \( c = (c_2, c_1, c_0) = (h^s, k^t, g^{s+t} \cdot m) \) where \( s,t \) are chosen uniformly random from \( \mathbb{Z}_p \). Decryption returns \( m \leftarrow c_0 \cdot c_1 \cdot c_2^{-1} \cdot g^{-\alpha^{-1}} \). Now suppose some tuple \((\alpha', \beta')\) that will be accepted by \( \text{SKCheck} \). By definition of \( \text{SKCheck} \) the following holds for all \( m \) and \( s,t \leftarrow \mathbb{Z}_p \):

\[
0 = s + t - \log(h) \cdot t \cdot \beta'^{-1} - \log(g) \cdot s \cdot \alpha'^{-1} \\
= t \cdot (1 - \log(k) \cdot \beta'^{-1}) + s \cdot (1 - \log(h) \cdot \alpha'^{-1})
\]

We conclude that for \( \text{SKCheck} \) to accept a proposed secret key, \((\alpha', \beta')\) must satisfy \( \alpha' = \log(h) \) and \( \beta' = \log(k) \). Thus, secret keys are unique and in particular efficiently re-randomizable. Moreover, there is an efficient \( \text{SKCheck} \)-procedure that, given a purported key \((\alpha', \beta')\) returns \( h = g^{\alpha'} \land k = g^{\beta'} \).

**Defining a suitable relation.**

Let \( \text{PKE} = (\text{Setup}, \text{Gen}, \text{Enc}, \text{Dec}) \) be a public key encryption scheme and let \( I \) be the range of \( \text{Setup} \). We set \( \mathcal{R}_{\text{PKE}} = \{R_\Pi\}_{\Pi \in I} \) where \( R_\Pi(x, y) := \text{SKCheck}_\Pi(y, x) \). Now, if \( \text{PKE} \) is \( t_{\text{ReRand}} \)-key re-randomizable then \( \mathcal{R}_{\text{PKE}} \) is \( t_{\text{ReRand}} \) re-randomizable.

Next, we show that any adversary that breaks the UF-SSA-security of \( \mathcal{R}_{\text{PKE}} \) then there is an attacker that breaks the MU-IND-CPA-C-security of \( \text{PKE} \).

**Claim.** If there is an attacker \( A \) that \((t, n, \epsilon)\)-breaks the UF-SSA-security of \( \mathcal{R}_{\text{PKE}} \) then there is an attacker \( B \) that \((t', n, \epsilon')\)-breaks the MU-IND-CPA-C-security of \( \text{PKE} \) with \( t' = O(t) \) and \( \epsilon' \geq \epsilon \).
Proof. We construct $\mathcal{B}$ that $(t', n, \epsilon')$-breaks the MU-IND-CPA-C-security of PKE, given black box access to $\mathcal{A}$ as follows:

1. $\mathcal{B}$ is called on input a public key $(pk)_{i \in [n]}$ and random tape $\rho$. Recall that $\Pi$ is contained in $pk$. First, $\mathcal{B}$ samples $\rho_\mathcal{A}$, the random coins of $\mathcal{A}$. After that, it runs $(j, st_\mathcal{A}) \leftarrow \mathcal{A}_1(\Pi, (pk)_{i \in [n]}, \rho_\mathcal{A})$.

2. $\mathcal{B}$ will issue a corrupt-query to oracle $O$.Corrupt for all $i \in [n \setminus j]$. It will obtain $sk_i$ such that $SKCheck_\Pi(vk_i, sk_i)$. After that, $\mathcal{B}$ runs $sk_j \leftarrow \mathcal{A}_2((sk_i)_{i \in [n \setminus j]}, st_\mathcal{A})$. Note that $SKCheck_\Pi(vk_j, sk_j) = 1$ with probability at least $\epsilon$.

3. Next, $\mathcal{B}$ samples $m_0 \neq m_1 \leftarrow \mathcal{M}$ such that $|m_0| = |m_1|$ and issues query $(m_0, m_1, j)$ to oracle $O$.Encrypt which will respond with $c \leftarrow \mathcal{E}nc(pk_j, m_b)$. $\mathcal{B}$ runs $m \leftarrow \mathcal{D}ec(sk_j, c)$ and returns $b' \leftarrow m =? m_1$. Note that $pk_j \notin Q^{Corrupt}$. Moreover, by the perfect correctness of PKE and the property of SKCheck, we have that $b' = b$ whenever $\mathcal{A}$ is successful.

\[\square\]

Tightness Bound.

**Theorem 4** (informal). Any simple reduction from breaking the security of a NICA $N$ to breaking the MU-IND-CPA-C-security of a perfectly correct public key encryption scheme PKE (cf. Definition 2.12) that provides efficient key re-randomization and that supports an efficient SKCheck loses a factor that is linear in the number of public keys the attacker is provided with and that it may corrupt, or $N$ is easy to solve.

We prove the above Theorem via the following technical Theorem, which is a direct application of Theorem 3.

**Theorem 5.** Let $N = (\text{Gen}, \text{Vfy}, \text{Triv})$ be a non-interactive complexity assumption, $n, r \in \text{poly}$ and let $\mathcal{R}_{\text{PKE}}$ be a family of computable relations as described above. Then for any $r$-simple
(\(t_\Gamma, n, \epsilon_\Gamma, 1\))-reduction \(\Gamma\) from breaking \(N\) to breaking the UF-SSA-security of \(R_{\text{PKE}}\) there exists an algorithm \(B\) that \((t_B, \epsilon_B)\)-breaks \(N\) where

\[
t_B \leq r \cdot n \cdot t_\Gamma + r \cdot n \cdot (n - 1) \cdot t_{\text{Vfy}} + r \cdot t_{\text{ReRand}}
\]

\[
\epsilon_B \geq \epsilon_\Gamma - \frac{r}{n}
\]

Here, \(t_{\text{ReRand}}\) is the time to re-randomize a given valid witness and \(t_{\text{Vfy}}\) is the maximum time needed to compute \(R \in R_{\text{PKE}}\).

3.4 Practical Implications

Here, we discuss the implications of our results when they are applied in practice. To this end, we note that our result from the previous section applies to the RSA encryption scheme \([\text{RSA}78]\) which has unique secret keys. PKCS#1v1.5, the most common negotiation mechanism of the TLS handshake protocol \([\text{DR}08]\) is based on the basic RSA cryptosystem and uses unique secret keys as well. Recently, proofs for various ciphersuites of the TLS protocol were published \([\text{JKSS}12, \text{KPW}13, \text{KSS}13]\). In particular \([\text{KPW}13]\) analyzed the security of the TLS-RSA ciphersuites in a security model called selective ACCE with server only authentication.

During the experiment the adversary is allowed to corrupt any server except for a server \(S\) that it will attack. It is considered successful if it manages to distinguish a particular ciphertext computed by \(S\) from a ciphertext encrypting a random message. We note that, in the case of TLS-RSA distinguishing these ciphertexts is trivial if the secret key of \(S\) is known since the key that encrypts the ciphertext of interest can be recovered using the decryption key of \(S\). Therefore a model where the adversary needs to output an uncorrupted secret key (as the UF-SSA-security model from the previous section) is weaker than a model where the adversary needs to distinguish a real ciphertext from random (as in the ACCE model).

In the selective ACCE model the adversary must choose the server \(S\) he wants to attack before seeing the parameters of the system. \([\text{KPW}13]\) proof that breaking TLS-RSA in this security
On the Impossibility of Tight Reductions

<table>
<thead>
<tr>
<th>Security level</th>
<th>1 server</th>
<th>$2^{16}$ server</th>
<th>$2^{32}$ server</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>1248</td>
<td>$\approx 1700$</td>
<td>2432</td>
</tr>
<tr>
<td>112</td>
<td>2432</td>
<td>3248</td>
<td>$\approx 4000$</td>
</tr>
</tbody>
</table>

Figure 3.5: Theoretically sound parameter size for RSA when used in TLS relative to the number of servers, when parameters are selected according to ECRYPT II recommendations (cf. Section 2.3.1).

model is roughly as hard as breaking RSA or forging a signature. The reduction is tight. The authors justify the selective security model by proving it polynomially equivalent to the non-selective model. Applying our result from Section 3.3 directly to the non-selective model we observe that any reduction from breaking the RSA assumption to breaking (non-selective) ACCE security of TLS-RSA must have a loss of at least $n$.

Recommended parameter bitlengths relative to the security level and number of TLS servers that support TLS-RSA are depicted in Figure 3.5.

3.5 Limitations: How to circumvent the bound

Given the generality of our impossibility result it is interesting to explore ways to circumvent it. We have applied Theorem 3.2 to cryptographic reductions from breaking a non-interactive complexity assumption to breaking an implementations of cryptographic primitives. Here, we required the induced relation to be efficiently re-randomizable. However, if the relation is not known to be efficiently re-randomizable we cannot apply the theorem. Here, we give some examples from the literature that show that it is possible to circumvent our tightness bound if, indeed, re-randomization is not efficiently possible.
3.5 Limitations: How to circumvent the bound

**Lossy Public Keys.** Kakvi and Kiltz \cite{kk12} circumvent the impossibility result of Coron \cite{cor02}. They showed that the result of Coron silently assumes that the underlying RSA public key specifies a trapdoor permutation. However, it can be shown that there exist special RSA public keys that can define a lossy trapdoor function \cite{pw08}. Moreover for appropriate RSA public keys these two instantiations are indistinguishable under the $\phi$-hiding assumption (which is a non-interactive complexity assumption). While the re-randomization algorithm for injective RSA keys is simply the identity map on the signature space there is no algorithm known that efficiently re-randomizes signatures for lossy keys. This enables Kakvi and Kiltz to give a tight proof of RSA-FDH in the random oracle model under the stronger $\phi$-hiding assumption.

**Naor-Yung Paradigm.** Another idea to circumvent our impossibility result is to apply variants of the Naor-Yung paradigm \cite{ny90}. In their original scheme a public key $pk$ consists of two distinct public keys $pk = (pk_0, pk_1)$. However, the user’s secret key $sk$ consist of only one secret key $sk_b$, either $b = 0$ or $b = 1$. Thus there are two possible secret keys for each public key and these secret keys are not re-randomizable in general. In the simulation, we can now allow the reduction to know the remaining secret key. This technique was used by Hofheinz and Jager \cite{hj12} to construct an IND-CCA secure public key encryption scheme in the multi user setting with many challenge ciphertexts that comes along with a tight security proof. Although it is not explicitly shown in \cite{hj12} their scheme also allows the adversary to corrupt secret keys. (Of course, a corrupted secret key may not be used to trivially win the security game. Therefore, their security model has to be adapted).

A similar technique has successfully been applied to signature schemes by Katz and Wang \cite{kw03} to obtain a tight security proof for a variant of the RSA full-domain hash signature scheme.

**Dual Form Signatures.** The dual form signatures paradigm that was introduced in \cite{glow12} is also a leverage to circumvent our impossibility result. Here, there are two signing algorithms,
SIG.Sign and SIG.Sign’, corresponding to a public key. Signatures output by both algorithms are accepted by the verification algorithm. For a dual form signature scheme to be secure it is required that given only signatures output by SIG.Sign it is hard to create a signature that is in the range of SIG.Sign’ and vice versa. Moreover, it is required that, given only the public parameters, it is hard to distinguish between signatures output by SIG.Sign and SIG.Sign’. The signature scheme from [CW13, BKP14] that comes along with an almost tight security proof to the DLIN assumption follows this approach.
4 Efficient Tightly Secure Signatures in the Standard Model

As explained in the introduction, the standard security notion for digital signatures considers only an ideal setting, where the adversary is provided with one public key. As also argued in the introduction, when signatures are deployed in practice, we need to assume that the adversary has the ability

1) to adaptively get signatures for messages of its choosing with respect to public keys of its choosing, and
2) to adaptively corrupt keys of its choice.

In this case, security should still hold with respect to uncorrupted keys. A more realistic security notion, compared to EUF-CMA-security, thus provides the adversary with polynomially many public keys and allows it to adaptively corrupt some of these. Moreover, it should obtain signatures with respect to adaptively chosen public keys and messages. Finally, the attacker is considered successful in breaking the scheme if it is able to output a valid signature $\sigma^*$ for a message $m^*$ that was not signed before with respect to an uncorrupted public key. We call this notion *multi-user existential unforgeability under chosen message attacks with corruptions.* (MU-EUF-CMA-C-security). As mentioned in the introduction, there is a standard reduction from MU-EUF-CMA-C-security to EUF-CMA-security \[\text{[MS04]}\] that roughly works as follows (Note that the reduction aims to break EUF-CMA-security giving access to an MU-EUF-CMA-C-attacker).

1) The reduction obtains a public key $vk^*$ for the respective signature scheme as input.
2) It samples, say $\ell - 1$, key-pairs for the given signature scheme. Next, it samples $j \leftarrow \mathbb{I} \lfloor \ell \rfloor$ uniformly at random and runs the MU-EUF-CMA-C-adversary on all public keys, such that the $j$-th public key equals $vk^*$.

3) It can simulate the MU-EUF-CMA-C-challenger perfectly if the adversary does not corrupt $vk^*$: When the adversary issues a sign query for any public key, except for $vk^*$ it signs the message itself. This is possible, since it knows the corresponding secret key. When the adversary issues a sign query for $vk^*$ it forwards this query to its EUF-CMA-challenger. Similarly, it can respond to all corrupt-queries, except when $vk^*$ is to be corrupted, by simply outputting the secret key.

4) When the adversary, however, attacks $vk^*$ by outputting a forgery for this public key, the reduction can forward this forgery as its own to its challenger in the EUF-CMA-security game. In this case, if the MU-EUF-CMA-C-attacker is successful, the reduction will also be.

Since the reduction needs to guess a priori the index $j$ of the challenge public key, it loses a factor of $\ell$. By Theorem 3, this loss is unavoidable in general. In particular, we show in [BJLS15] that for the Waters signature scheme [Wat05], there is can be no tight reduction in the multi-user setting with corruptions.

**The Difficulty of Tightly MU-EUF-CMA-C-secure Signatures.** When designing a signature scheme with tight MU-EUF-CMA-C-security we are faced with the following problem: On the one hand we need to be able to reveal the secret key to any public key (note that guessing the target public key would cause a loss of $\ell$) and on the other hand we must be able to extract a solution to a hard problem from (almost) any forgery that is output by the adversary. That is, we must be able to extract a solution from a forgery even if we know the secret key corresponding to the target public key.

**Contribution.** In this chapter we propose a signature scheme that tightly satisfies MU-EUF-CMA-C security, i.e., the running time and the success probability of the reduction are roughly
the same as the running time and the success probability of the adversary (and in particular independent of $\mu$ and $\ell$ except for a negligible fraction). The security reduction for the schemes loses roughly a factor of 2.

The scheme is a generic construction which is provably secure in the standard model. It compiles a signature scheme that is secure in the multi-user setting without corruptions and a suitable witness indistinguishable proof of knowledge, cf. Section 2.3.3, into an MU-EUF-CMA-C-secure scheme. The construction is tightness preserving. That is, the security loss is roughly the same, as for the underlying signature scheme that is provably secure without corruptions.

Following [NY90], a public key of our signature scheme consists of two public keys of the underlying signature scheme that is secure without corruptions whereas the signer keeps only one of the two corresponding secret keys. A signature in our new scheme will be a proof using the proof system that the signer actually knows a valid signature of the underlying scheme with respect to at least one of the two public keys. Relying on the witness indistinguishability of the proof system, the reduction will hide away from the adversary which secret key is known. At the same time, relying on the extractability of the proof system, with high probability, we can break security of the underlying signature scheme, whenever the attacker is succesful in outputting a valid forgery.

The scheme can be implemented with any structure preserving signature scheme [AGHO11] in combination with Groth-Sahai proofs [GS08]. In particular, when it is instantiated with the known tightly MU-EUF-CMA-secure and structure preserving signature scheme from [HJ12], we obtain the first tightly MU-EUF-CMA-C-secure signature scheme in the standard model. However, signatures are quite long. Our compiler was published at TCC 2015 in [BHJ+15].

**Organization.** After defining MU-EUF-CMA-C-security for digital signature schemes formally in Section 4.1, we will describe our standard model scheme in Section 4.2. This section contains a detailed description of the scheme (Section 4.2.1), a proof of security (Section 4.2.2) and a discussion of instantiations.
Games \( \text{MU-EUF-CMA}^{\text{SIG}}(A, \mu, \ell, \kappa) \) and \( \text{MU-EUF-CMA-C}^{\text{SIG}}(A, \mu, \ell, \kappa) \)

\[
\begin{aligned}
\Pi & \leftarrow \$ \text{SIG.Setup}(1^n) \\
(vk_i, sk_i) & \leftarrow \$ \text{SIG.Gen}(\Pi), i \in [\ell] \\
S^{\text{Corrupt}} & \leftarrow \emptyset; S_i & \leftarrow \emptyset, i \in [\ell] \\
(i^*, m^*, \sigma^*) & \leftarrow \$ \text{A}\text{Sign}(\cdot, \cdot), \text{Corrupt}(\cdot) \left( \Pi, \bigcup_{i \in [\ell]} \{vk_i\} \right) \\
\text{return} & \text{SIG.Vfy}(vk_i^*, m^*, \sigma^*) \wedge i^* \notin S^{\text{Corrupt}} \wedge m^* \notin S_i^* \\
\text{return} & \text{SIG.Vfy}(vk_i^*, m^*, \sigma^*) \wedge S^{\text{Corrupt}} = \emptyset \wedge m^* \notin S_i^* \\
\end{aligned}
\]

\text{Sign}(i, m):

\[
\begin{aligned}
\text{if } |S_i| > \mu \text{ return } \perp \\
\sigma & \leftarrow \$ \text{SIG.Sign}(sk_i, m) \\
S_i & \leftarrow S_i \cup \{m\} \\
\text{return} & \sigma \\
\end{aligned}
\]

\text{Corrupt}(i):

\[
\text{S^{Corrupt}} \leftarrow S^{\text{Corrupt}} \cup \{i\} \\
\text{return} & sk_i
\]

Figure 4.1: Security Games for Digital Signatures.

of our compiler with implementations of signature schemes from the literature (Section 4.2.3).

### 4.1 Security Notions for Digital Signatures in the Multi User Setting

The standard security notion for digital signatures (in the single user setting) is \textit{existential unforgeability under chosen message attacks} (EUF-CMA-security, cf. Definition 2.11) which goes back to [GMR88]. This notion was later extended to the multi-user setting [MS04] (MU-EUF-CMA-security notion). We note that EUF-CMA-security and MU-EUF-CMA-security coincide if \( \ell = 1 \) and recall only MU-EUF-CMA security here (EUF-CMA-security is defined in Definition 2.11). The security game is depicted in Figure 4.1 when only the non-boxed lines of code are carried out.

**Definition 4.1** (MU-EUF-CMA-security). \textit{We say that an attacker} \( A (t, \mu, \ell, \epsilon) \)-breaks the multi user existential unforgeability under chosen message attacks \textit{security of a signature scheme}
4.2 Generic Construction in the Standard Model

If it runs in time \( t(\kappa) \) and

\[
\Pr \left[ \text{MU-EUF-CMA}^{\text{SIG}}(A, \mu, \ell, \kappa) \Rightarrow 1 \right] \geq \epsilon
\]

In [BHJ+15] we put forward a security notion for digital signatures in the multi user setting (MU-EUF-CMA-C-security) that also allows the adversary to corrupt secret keys. In this case, unforgeability should still hold for uncorrupted parties. The security game is depicted in Figure 4.1 when the non-boxed and full-boxed lines of code are carried out.

**Definition 4.2 (MU-EUF-CMA-C-security).** We say that an adversary \( A(t, \mu, \ell, \epsilon) \)-breaks the multi user existential unforgeability under chosen message attacks with adaptive corruptions security of a signature scheme \( \text{SIG} \) if it runs in time \( t(\kappa) \) and

\[
\Pr \left[ \text{MU-EUF-CMA-C}^{\text{SIG}}(A, \mu, \ell, \kappa) \Rightarrow 1 \right] \geq \epsilon
\]

**Remark 4.1 (On the polynomial equivalence of the above security notions).** We remark that MU-EUF-CMA-security and MU-EUF-CMA-C-security are polynomially equivalent to EUF-CMA-security. However, we are interested in tight reductions. That is, why we explicitly define MU-EUF-CMA-C-security.

### 4.2 Generic Construction in the Standard Model

Our standard model scheme generically compiles an MU-EUF-CMA-secure signature scheme \( \text{SIG}_{\text{MU}} \) and a suitable NIWI-PoK \( \text{NIPS} \) (cf. Section 2.3.3) into an MU-EUF-CMA-C-secure signature scheme \( \text{SIG}_{\text{MU-C}} \). The scheme is actually a two copy version of the underlying signature scheme. Namely, following Naor-Yung [NY90], a public key of \( \text{SIG}_{\text{MU-C}} \) consists of two \( \text{SIG}_{\text{MU}} \) public keys whereas a \( \text{SIG}_{\text{MU-C}} \) secret key contains only one of the two corresponding \( \text{SIG}_{\text{MU}} \) secret keys. Now, a signature over a message \( m \) is a \( \text{NIPS} \) proof of knowledge of a signature over \( m \) with respect to at least one of the two \( \text{SIG}_{\text{MU}} \) public keys.
More formally, for a signature scheme \( \text{SIG} = (\text{SIG}.\text{Setup}, \text{SIG}.\text{Gen}, \text{SIG}.\text{Sign}, \text{SIG}.\text{Vfy}) \), let us consider the relation

\[
R := \left\{ \left( (v_{k_0}, v_{k_1}, m), (\sigma_0, \sigma_1) \right) : \begin{array}{ll}
\text{SIG}.\text{Vfy}(v_{k_0}, m, \sigma_0) = 1 \\
\vee \text{SIG}.\text{Vfy}(v_{k_1}, m, \sigma_1) = 1
\end{array} \right\}.
\]

That is, \( R \) consists of statements of the form \( (v_{k_0}, v_{k_1}, m) \), where \( (v_{k_0}, v_{k_1}) \) are verification keys for signature scheme \( \text{SIG} \), and \( m \) is a message. Witnesses are tuples \( (\sigma_0, \sigma_1) \) such that either \( \sigma_0 \) is a valid signature for \( m \) under \( v_{k_0} \), or \( \sigma_1 \) is a valid signature for \( m \) under \( v_{k_1} \), or both.

In the sequel, let \( \text{NIPS} = (\text{NIPS}.\text{Gen}, \text{NIPS}.\text{Prove}, \text{NIPS}.\text{Vfy}) \) be a NIWI-PoK for \( R \). A signature over message \( m \) with respect to \( v_k = (v_{k_0}, v_{k_1}) \) will actually be a \( \text{NIPS} \)-proof for relation \( R \). A witness for the statement \( (v_{k_0}, v_{k_1}, m) \) that is given by the public key and the message may be computed with the use of \( sk \).

### 4.2.1 Description of the Scheme

The scheme \( \text{SIG}_{\text{MU-C}} = \text{SIG}_{\text{MU-C}}(\text{SIG}_{\text{MU}}, \text{NIPS}) \) works as follows:

#### \( \text{SIG}_{\text{MU-C}}.\text{Setup}(1^\kappa) \). The parameter generation algorithm inputs the security parameter. It runs \( \text{CRS} \leftarrow \$ \ \text{NIPS}.\text{Gen}(1^\kappa), \Pi_{\text{SIG}_{\text{MU}}} \leftarrow \$ \ \text{SIG}_{\text{MU}}.\text{Setup}(1^\kappa) \) and returns \( \Pi_{\text{SIG}_{\text{MU-C}}} \leftarrow (\text{CRS}, \Pi_{\text{SIG}_{\text{MU}}}) \).

#### \( \text{SIG}_{\text{MU-C}}.\text{Gen}(\Pi_{\text{SIG}_{\text{MU-C}}}) \). The key generation algorithm parses the public parameters \( \Pi_{\text{SIG}_{\text{MU-C}}} \) as \( (\text{CRS}, \Pi_{\text{SIG}_{\text{MU}}}) \) and runs \( (v_k, s_k_i) \leftarrow \$ \ \text{SIG}_{\text{MU}}.\text{Gen}(\Pi_{\text{SIG}_{\text{MU}}}) \), for \( i \in \{0, 1\} \). It flips a uniformly random coin \( \delta \leftarrow \$ \ \{0, 1\} \), discards \( s_k_{1-\delta} \), and returns \( (v_k, s_k) = ((v_{k_0}, v_{k_1}), (s_k_\delta, \delta)) \).

#### \( \text{SIG}_{\text{MU-C}}.\text{Sign}(s_k, m) \). The signing algorithm parses \( s_k \) as \( (s_k_\delta, \delta) \) and computes \( \sigma \leftarrow \$ \ \text{SIG}_{\text{MU}}.\text{Sign}(s_k_\delta, m) \). Next, it computes and returns \( \pi \) as

\[
\pi = \begin{cases} 
\text{NIPS}.\text{Prove} \left( (\text{CRS}, (v_{k_0}, v_{k_1}, m), (\sigma, \perp)) \right), & \text{if } \delta = 0 \\
\text{NIPS}.\text{Prove} \left( (\text{CRS}, (v_{k_0}, v_{k_1}, m), (\perp, \sigma)) \right), & \text{if } \delta = 1
\end{cases}.
\]

#### \( \text{SIG}_{\text{MU-C}}.\text{Vfy}(v_k, m, \sigma) \). The verification algorithm parses \( v_k \) as \( v_k = (v_{k_0}, v_{k_1}) \) and returns \( \text{NIPS}.\text{Vfy}(\text{CRS}, (v_{k_0}, v_{k_1}, m), \sigma) \).
By the correctness of the proof system the signature scheme is also correct.

### 4.2.2 Proof of Security

**Theorem 6.** For any attacker $A$ that $(t, \mu, \ell, \epsilon)$-breaks the MU-EUF-CMA-C-security (with corruptions) of $\text{SIG}_{\text{MU-C}}$ there exists an algorithm $B = (B_{\text{NIPS}}, B_{\text{SIG}})$ such that either $B_{\text{NIPS}}(t_{\text{CRS}}, \epsilon_{\text{CRS}})$-breaks the security of $\text{NIPS}$ or $B_{\text{SIG}}(t_{\text{SIG-MU}}, \mu, \ell, \epsilon_{\text{SIG-MU}})$-breaks the MU-EUF-CMA-C-security (without corruptions) of $\text{SIG}_{\text{MU}}$ with $t \approx t_{\text{CRS}} \approx t_{\text{SIG-MU}}$ and

$$\epsilon < 2 \cdot \epsilon_{\text{SIG-MU}} + \epsilon_{\text{CRS}}$$

**Proof.** We proceed in a sequence of games [Sho04, BR06]. The first game is the real MU-EUF-CMA-C-security game, as described in Section 4.1.

We denote by $\chi_i$ the event that $A$ outputs $(i^*, m^*, \sigma^*)$ such that $i^* \notin S^* \land m^* \notin S_{i^*} \land \text{SIG}_{\text{MU-C}}.\text{Vfy}(vk^*, m^*, \sigma^*) = 1$ in Game $i$.

**GAME 0.** This is the real MU-EUF-CMA-C-security game, cf. Figure 4.1. Thus, we have:

$$\Pr[\chi_0] = \epsilon.$$

**GAME 1.** In this game we change the way keys are generated. Namely, when generating a key pair by running $\text{SIG}_{\text{MU-C}}.\text{Gen}$, we do not discard $sk_1 - \delta$. That is, we set $sk = ((sk_0, sk_1), \delta)$. However, corruption queries by the attacker are still answered by responding only with $sk_{\delta}$. Therefore this change is completely oblivious to $A$. Thus, we have:

$$\Pr[\chi_1] = \Pr[\chi_0]$$

**GAME 2.** In this game, we proceed exactly like in Game 1 except that we change the behavior of the $\text{Sign}$ procedure that $A$ may call. To explain the change, recall that a $\text{SIG}_{\text{MU-C}}$-signature in Game 1 consists of a proof $\pi \leftarrow \$ \text{NIPS}.\text{Prove}(\text{CRS}, (vk_0, vk_1, m), w)$, where either $w = (\sigma_0, \bot)$ or $w = (\bot, \sigma_1)$ where $\delta \in \{0, 1\}$.
is specified by the secret key and $\sigma_\delta \leftarrow \$ \text{SIG}_\text{MU}.\text{Sign}(sk_\delta, m)$.
In Game 2 we modify the \text{Sign} procedure as follows. When called on $(i, m)$, we compute $\sigma_\delta \leftarrow \$ \text{SIG}_\text{MU}.\text{Sign}(sk_\delta, m)$ for all $\delta \in \{0, 1\}$ (note that both $\text{SIG}_\text{MU}$ secret keys are available due to Game 1) and set $w \leftarrow (\sigma_0, \sigma_1)$. Now, $\text{Sign}(i, m)$ returns $\pi \leftarrow \$ \text{NIPS}.\text{Prove}(\text{CRS}, (vk_i, m), w)$. Due to the perfect witness indistinguishability property of $\text{NIPS}$ we have:

$$\Pr[\chi^1] = \Pr[\chi^2]$$

**GAME 3.** This game is very similar to the previous game, except that we change the way the CRS is generated. Now, we run $(\text{CRS}_\text{sim}, \tau) \leftarrow \$ \mathcal{E}_0$ and all proofs are generated with respect to $\text{CRS}_\text{sim}$. Since the contrary would allow $\mathcal{B}_\text{NIPS}$ to break the $(t, \epsilon_{\text{CRS}})$-security of $\text{NIPS}$ we have

$$|\Pr[\chi^2] - \Pr[\chi^3]| < \epsilon_{\text{CRS}}$$

**GAME 4.** This game is similar to Game 3 except for the following. Before verifying the forgery $(i^*, m^*, \sigma^*)$ output by $\mathcal{A}$ we run $w = (s_0, s_1) \leftarrow \$ \mathcal{E}_1(\text{CRS}_\text{sim}, \sigma^*, (vk^*, m^*), \tau)$ and abort (i.e., the Game returns 0) if $((vk, m^*), w) \notin R$. By the perfect knowledge extraction of $\text{NIPS}$ on $\text{CRS}_\text{sim} \leftarrow \$ \mathcal{E}_0(1^\kappa)$ we have that $((vk^*, m^*), w) \notin R \iff \text{SIG}_\text{MU}.\text{Vfy}(vk^*, m^*, \sigma^*) = 0$ and thus:

$$\Pr[\chi^3] = \Pr[\chi^4]$$

**GAME 5.** In Game 5 we proceed similar to Game 4 except that we add another abortion rule. Namely, after running $w = (s_0, s_1) \leftarrow \$ \mathcal{E}_1(\text{CRS}_\text{sim}, \sigma^*, (vk^*, m^*), \tau)$, we abort if

$$\text{SIG}_\text{MU}.\text{Vfy}\left(vk^{(i^*)}_{1-\delta(i^*)}, m^*, s_{1-\delta(i^*)}\right) = 0.$$ (4.1)

Recall here that $\delta^{(i^*)}$ denotes the random bit chosen during the generation of the long-term secret key of user $i^*$. In particular, $\mathcal{A}$ will obtain $sk_\delta^{i^*}$ when querying oracle $\text{Corrupt}$ on input $i^*$.

If Equation (4.1) is satisfied, then the game is aborted. Said otherwise, the challenger aborts, if the witness $s_{1-\delta(i^*)}$ is not a valid signature for $m^*$ under $vk^{(i^*)}_{1-\delta(i^*)}$.
Since \( A \) is not allowed to corrupt the secret key of user \( i^* \) and sees only proofs which use two valid signatures \((s_0, s_1)\) as witnesses (cf. Game 2), the random bit \( \delta(i^*) \) is information-theoretically perfectly hidden from \( A \). Therefore, we have

\[
\Pr[\chi_4] \leq 2 \cdot \Pr[\chi_5]
\]

Claim.

\[
\epsilon_{\text{SIGMU}} \geq \Pr[\chi_5]
\]

Given the above claim, we can conclude the proof of Theorem 6. In summary we have

\[
\epsilon \leq \epsilon_{\text{CRS}} + 2 \cdot \epsilon_{\text{SIGMU}}.
\]

Proof. We describe an attacker \( B_{\text{SIGMU}} \) that \((t, \mu, \ell, \epsilon)\)-breaks the MU-EUF-CMA-C security (cf. Figure 4.1) of SIGMU, if \( \chi_5 \) occurs.

\( B_{\text{SIGMU}} \) is called on input \((\Pi_{\text{SIGMU}}, \cup_{i \in [\ell]} \{vk(i)\})\). Next, it samples \( \ell \) key pairs \((vk^{(i)}, sk^{(i)}) \leftarrow \$ \text{SIGMU}.\text{Gen}(\Pi_{\text{SIGMU}}), i \in \{\mu + 1, \ldots, 2\mu\}\) and chooses a random vector \( \delta = (\delta(1), \ldots, \delta(\mu)) \in \{0, 1\}^\mu \). It sets

\[
(vk^{(i)}, sk^{(i)}) \leftarrow \left((vk(\delta(i)\mu + i), vk((1 - \delta(i))\mu + i)), (sk^{\mu + i}, 1 - \delta(i))\right).
\]

Note that now each SIGMU-verification key contains one SIGMU-verification key that \( B_{\text{SIGMU}} \) received as input, and one that was generated by \( B_{\text{SIGMU}} \) itself. \( B_{\text{SIGMU}} \) furthermore generates a “simulated” CRS for the NIWI-PoK along with a trapdoor by running \((\text{CRS}_{\text{sim}}, \tau) \leftarrow \$ \text{E}_0\). It sets \( \Pi \leftarrow (\text{CRS}_{\text{sim}}, \Pi_{\text{SIGMU}}), \) exactly as in Game 5 and runs \( A \) on input \((\Pi, \cup_{i \in [\ell]} \{vk(i)\})\).

\( B_{\text{SIGMU}} \) simulates procedures \( \text{Sign}(\cdot, \cdot) \) and \( \text{Corrupt}(\cdot) \) as follows:

**Sign**\((i, m)\). When asked to sign a message \( m \) under public key \( vk^{(i)} \), \( B_{\text{SIGMU}} \) proceeds as follows. Let \( \delta(i) = 0 \) without loss of generality. Then it computes \( \sigma_1 = \text{Sign}(sk^{(\mu + i)}, m) \) and requests a signature for public key \( vk^{(i)} \) and message \( m \) by calling oracle \( \text{Sign}(i, m) \) in the MU-EUF-CMA security
game. Let $\sigma_0$ be the response. $B_{\text{Sig}_\mu}$ computes the signature for $m$ using the witness $w = (\sigma_0, \sigma_1)$. Note that this is a perfect simulation of Game 5.

$\text{Corrupt}(i)$. $B_{\text{Sig}_\mu}$ simply returns $sk^{(i)}$. We note that, given $vk^{(i)}$, $sk^{(i)}$ is distributed correctly. Thus, this is a perfect simulation of oracle $\text{Corrupt}$ in Game 5.

Now, suppose that $A$ outputs $(i^*, m^*, \sigma^*)$ such that Game 5 will return 1. In this case, by Game 4 and Game 5 we know that $B_{\text{Sig}_\mu}$ is able to extract a witness $w = (s_0, s_1)$ from $\sigma^*$ which satisfies $\text{Sig}_\mu.\text{Vfy}(vk^{(i^*)}, m^*, s_\delta^{(i^*)}) = 1$. In this case, $B_{\text{Sig}_\mu}$ will return $(i^*, m^*, s_\delta^{(i^*)})$ which is a valid forgery in Game MU-EUF-CMA. The claim follows.

4.2.3 Instantiation with Building Blocks from the Literature

Here, we show how to instantiate our signature scheme from the previous section with existing Building Blocks to obtain a tightly MU-EUF-CMA-C-secure signature scheme. Since the proof for our compiler from Section 4.2 is tightness preserving, the tightness of the overall construction depends on the tightness of the underlying signature scheme $\text{Sig}$.

If this underlying signature scheme $\text{Sig}$ is structure preserving $[\text{AFG}^{+10}]$ (i.e., the scheme’s public keys and signatures consists only of source group elements of a bilinear pairing and verifying a signature is checking a conjunction of pairing product equations for satisfiability) we can apply techniques from $[\text{Gro06}]$ to obtain a set of PPEs for our relation $R$. In this case, the Groth-Sahai proof system $[\text{GS08}]$ is a suitable NIWI-PoK for $R$: There are two types of common reference strings, one type is hiding and the other type is binding both of which are indistinguishable under a computational assumption. A hiding CRS provides prefect witness indistinguishability while a binding CRS provides perfect witness extractability of group elements. This extraction property suffices for our purposes if we use structure preserving signature schemes since, here, a signature consists only of group elements and thus may be extracted.
4.2 Generic Construction in the Standard Model

|               | $|\Pi|$ | $|vk|$ | $|\sigma|$ | Loss     |
|---------------|-------|-------|-----------|----------|
| SIG_{MU}(Wat09, NIPS-GS) | $O(1) \cdot |G|$ | $O(1) \cdot |G|$ | $O(1) \cdot |G|$ | $O(\mu)$ |
| SIG_{MU}(CW13, NIPS-GS)    | $O(1) \cdot |G|$ | $O(\kappa) \cdot |G|$ | $O(1) \cdot |G|$ | $O(\kappa)$ |
| SIG_{MU}(HJ12, NIPS-GS)    | $O(1) \cdot |G|$ | $O(1) \cdot |G|$ | $O(\kappa) \cdot |G|$ | $O(1)$     |

Figure 4.2: Efficiency comparison of DLIN-based MU-EUF-CMA-C-signature schemes when derived from MU-EUF-CMA-secure signatures and DLIN-based Groth-Sahai proofs (NIPS-GS) in symmetric bilinear groups (cf. Section 2.2.2) using our Compiler.

Figure 4.2 asymptotically compares the efficiency of signature schemes derived using our compiler when instantiated with signature schemes from the literature that allow to be combined with Groth-Sahai proofs. Since the schemes are not very practical, we consider only asymptotic signature sizes. Here, $\mu$ denotes the number of sign queries per public key and $\kappa$ is the security parameter. We note that either the size of a signature or the security loss is at least linear in $\kappa$ which is unsatisfactory, at least from a theoretical point of view. Therefore, in the next Chapter, we provide a practical solution which is provably tightly MU-EUF-CMA-C-secure in the random oracle model.
5 Practical Tightly Secure Signatures in the Random Oracle Model

Given the practical importance of MU-EUF-CMA-C-security (we refer to Section 1.2 for a discussion on the importance of MU-EUF-CMA-C-security), we note that our full tight construction from Chapter 4 is not of practical importance since its signatures are quite long. In this Chapter, we consider random oracle based signatures in the multi-user setting with corruptions.

Probably the most natural way to construct a tightly secure signature scheme in the ROM would be to apply OR proofs as introduced in \[\text{CDS94}\] to Fiat-Shamir like signature schemes that have a tight reduction, e.g. \[\text{KW03}\] in a way similar to the construction from Chapter 4 (i.e., following the Naor-Yung paradigm \[\text{NY90}\]).

The OR proofs from \[\text{CDS94}\] provide \textit{perfect} witness indistinguishability and thus it remains information theoretically hidden from the view of the adversary which secret key is known by the reduction. Unfortunately perfect witness indistinguishability makes the reduction fail to actually extract knowledge from the forgery output by the adversary without rewinding. Thus, it is unclear if a tight reduction exists in this setting.

If we are to apply pairings we can resort to Groth-Sahai proofs \[\text{GS08}\] and could apply a similar technique. However, in this case we need to prove satisfiability of a set of \textit{quadratic} equations (as in Chapter 4) which makes the proofs expensive, i.e., large. Since we are interested in efficient schemes we do not apply this technique.

\textbf{Contribution.} In this chapter, we develop a practical signature scheme that tightly satisfies MU-EUF-CMA-C security. The se-


5 Practical Tightly Secure Signatures in the ROM

curity reduction loses a factor of roughly 2. The scheme works over asymmetric bilinear groups $\mathbb{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ equipped with an efficiently computable pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ (cf. Section 2.2.2). Public parameters contain a description of the group, one additional element from $\mathbb{G}_1$, two additional elements from $\mathbb{G}_2$ and the description of a Hash-function. A public key is a single group element from $\mathbb{G}_2$ and signatures live in $\mathbb{G}_1^2 \times \mathbb{G}_2^2$.

A signature will roughly be a proof (using a suitable proof system) that the signer 'knows' a one time signature with respect to a given one-time signature public key. Now, to hide all critical information from the adversary all proofs output by the reduction need to be hiding. At the same time, to extract a solution from an adversarially generated signature the proof output by the adversary needs to be binding. Note that we do not know the target public key and message up front.

To achieve this we apply the random oracle in a way similar to Katz-Wang [KW03] to the (standard model, DLIN-based) linearly homomorphic signature scheme from [LPJY14] converted to the SXDH-setting. In our scheme, public parameters contain part of a Groth-Sahai CRS [GS08]. To sign a message $m$, a bit $b$ is sampled uniformly at random. The message is hashed together with $b$ and the public key of the signer to complete the CRS. Finally, using the secret key, a one time signature over $m$ is computed and correctness of the computed signature is proved with respect to the CRS. During the security reduction the random oracle will be programmed such that for each pair of message $m$ and public key $vk$ one out of two possible CRS (recall that $m$ and $vk$ are hashed together with $b$) is perfectly hiding and the other one is perfectly binding. Both are indistinguishable under a computational assumption. Now, the reduction will make all proofs on a hiding CRS (and thus leak no information about $sk$) and with high probability the adversary will output a forgery on a binding CRS from which we can extract a solution to a hard problem with overwhelming probability.

Using the programmable hash function paradigm [HK08] on Groth-Sahai proof common-reference strings [GS08], the proof can be lifted to the standard model. The price for this is that the public parameters grow linearly in $\kappa$ (the security parameter) and that the loss grows linearly in $\ell \cdot \mu$.  


The scheme was published in [Bad14].

**Remark 5.1** (On the random oracle model.). *The value of a proof in the random oracle model is discussed controversially in the literature. It is known that there are cryptographic schemes that are secure in the random oracle model but whenever the random oracle is replaced with a real hash-function they become insecure [CGH98]. Thus, the random-oracle methodology is not sound in general. However, as has been admitted by the authors themselves, the constructions from [CGH98] are contrived, as are other “counterexamples”, e.g., [BBP04, GK03]. Koblitz and Menezes conclude [KM07] is that if these bizarre constructions are the best possible counterexamples then this is a good reason to have even more confidence in the random oracle. A recent overview of the discussion can be found in [KM15]. Given the fact that there are actually no security weaknesses in practice that stem from using a random oracle for the security proof of a scheme [KM15], we think that a proof in the random oracle model is valuable. E.g., the RSA full domain hash scheme [BR93b] is proven to be secure in the random oracle model and has withstood attacks on the bare scheme for two decades [KL07].*

**Outline.** After giving some intuition on our scheme in Section 5.1, we will describe our new scheme in Section 5.2 and give a formal proof of security in Section 5.3. Finally, we will evaluate efficiency of our scheme in Section 5.4.

### 5.1 Intuition

Before we introduce our scheme formally we would like to give some intuition what is behind the scheme. To this end, consider the following equation over \((z, r)\):

\[
1 = e(z, k_z) \cdot e(r, k_r) \cdot e(m, k) \tag{5.1}
\]

The core of our signature scheme will be the assumption (which we will justify later) that given \((k_z, k_r, k, m)\) it is hard to compute \((z, r)\) that satisfy Equation 5.1. From the above equation we
define a relation as follows:

\[ R((k_z, k_r, k, m), (z, r)) = \begin{cases} 
1, & \text{if } 1 = e(z, k_z) \cdot e(r, k_r) \cdot e(m, k) \\
0, & \text{else} 
\end{cases} \] (5.2)

In our signature scheme, \( k_z, k_r \) and \( k \) will be fixed as public parameters and public key, respectively. A signature over \( m \) is now a NIWI-PoK of a satisfying assignment for equation 5.1 using a suitable proof system for relation \( R \). The system parameters contain part of a CRS for relation \( R \). The hash of a message, the verification key of the signer and a uniformly random bit completes the CRS. Now, the signer (using sk) computes \((z, r)\) that satisfies equation 5.1 and generates a proof of this fact using the proof system for relation \( R \). We note that there are many possible solutions to equation 5.1. The secret key of our signature scheme allows to compute exactly one satisfying solution to equation 5.1. However, two distinct solutions yield a solution to the instance \((k_z, k_r)\) of the DP\(_2\) problem, cf. Definition 2.7.

**Suitable Proof Systems.** The SXDH-based Groth-Sahai proof system [GS08] is an efficient proof system for witness relation \( R \). Note that equation 5.1 is linear where the variables live in \( G_1 \). In this case each commitment costs two elements from \( G_1 \) and a proof element costs additional two elements from \( G_2 \) (instead of four elements from \( G_1 \) and \( G_2 \) if we had quadratic equations as in Section 4.2).

Since we need the notation for our signature scheme we recall SXDH-based Groth-Sahai proofs [GS08] for relation \( R \) here.

\[ \text{CRS} \leftarrow \$ \text{Gen}(1^\kappa): \text{The common reference string generation algorithm runs } \mathbb{G} = (e, G_1, G_2, G_T, g_1, g_2, p) \leftarrow \$ \text{GEN.asym}(1^\kappa), \]
\[ \bar{v}_1 = (g_1, f_1) \leftarrow \$ G_1^2 \text{ and } \bar{v}_2 = (\hat{g}_1, f_1) \notin \text{DDH}(g_1, f_1). \] It returns \((\mathbb{G}, \bar{v}_1, \bar{v}_2)\).

\[ \pi \leftarrow \$ \text{Prove}(\text{CRS}, (k_z, k_r, k, m), (z, r)): \text{The prove algorithm first commits to } z \text{ and } r \text{ via} \]

\[ C_z = (1, z) \cdot \bar{v}_1^{\delta_{z,1}} \cdot \bar{v}_2^{\delta_{z,2}} \]
\[ C_r = (1, r) \cdot \bar{v}_1^{\delta_{r,1}} \cdot \bar{v}_2^{\delta_{r,2}} \]
5.2 Description of the Scheme

where multiplication is done component-wise. Next, it computes proofs that the commitments actually contain a solution to equation 5.1. These are computed as

$$\pi' = (\pi'_1, \pi'_2) = \left( k_z^{-\delta z, 1} \cdot k_r^{-\delta r, 1}, k_z^{-\delta z, 2} \cdot k_r^{-\delta r, 2} \right)$$

The proof is returned as

$$\pi = (C_z, C_r, \pi') \in G_1^4 \times G_2^2.$$

**Vfy**(CRS, (k_z, k_r, k, m, π)): The verification algorithm outputs 1 iff

$$\left( E((1, m), k) \right)^{-1} = E(C_z, k_z) \cdot E(C_r, k_r) \cdot E(\vec{v}_1, \pi'_1) \cdot E(\vec{v}_2, \pi'_2) \quad (5.3)$$

Here, given vectors $\vec{g} \in G_2^1$ and $\vec{h} \in G_2^2$, $E(\vec{g}, \vec{h})$ evaluates the pairing $e: G_1 \times G_2 \to G_T$ component-wise. By $E(\vec{g}, k)$ with $\vec{g}$ as above and $h' \in G_2$ we abbreviate $E(\vec{g}, (h', h'))$ and similarly, given $g' \in G_1$ and $\vec{h}$ as above, we abbreviate $E((g', g'), \vec{h})$ by $E(g', \vec{h})$.

(CRS_{sim}, td) $\leftarrow E_0(1^\kappa)$: The simulated CRS generation algorithm runs $\mathbb{G} = (e, G_1, G_2, G_T, g_1, g_2, p) \leftarrow \text{GEN.asym}(1^\kappa)$, sets $\vec{v}_1 = (g_1, f_1) \leftarrow G_2^1$ and $\vec{v}_2 = (\hat{g}_1, \hat{f}_1) \leftarrow \text{DDH}(g_1, f_1)$ and returns $((\mathbb{G}, \vec{v}_1, \vec{v}_2), x)$ where $x = \log_{g_1}(f_1)$.

That for any attacker A that $(t_A, \epsilon_A)$-breaks the NIWI-PoK security of this proof system there is an attacker B that $(t_B, \epsilon_B)$-breaks the SXDH-assumption in $\mathbb{G}$ with $t_A \approx t_B$ and $\epsilon_B \geq \epsilon_A$ is proven in [GS08]. We stress that if $\vec{v}_2 \in \text{DDH}(\vec{v}_1)$ then $(\vec{v}_1, \vec{v}_2)$ is a perfectly binding CRS whereas if $\vec{v}_2 \notin \text{DDH}(\vec{v}_1)$ then $(\vec{v}_1, \vec{v}_2)$ yields a perfectly hiding CRS both of which are computationally indistinguishable under the XDH_1 assumption in $\mathbb{G}$.

5.2 Description of the Scheme

The scheme is similar to the linearly homomorphic signature scheme from [LPJY14] which is provably secure in the standard model. However, its loss is linear in $\mu \cdot \ell$ while our scheme can be proven tightly secure. In detail, the scheme works as follows.

SIG.Setup$(1^\kappa)$. The setup algorithm, on input $1^\kappa$, works as follows:

1. Run $\mathbb{G} = (e, G_1, G_2, G_T, g_1, g_2, p) \leftarrow \text{GEN.asym}(1^\kappa)$.
2. Sample $f_1 \leftarrow \mathcal{G}_1$ and $k_z, k_r \leftarrow \mathcal{G}_2$ and set $\vec{v}_1 = (g_1, f_1)$.

3. Choose a hash-function $H : \{0,1\}^* \rightarrow \mathcal{G}_1$. The security analysis will view $H$ as a random oracle.

It returns $\Pi \leftarrow (\mathcal{G}, k_z, k_r, \vec{v}_1, H)$. The message space is $\mathcal{G}_1$.

**SIG.Gen(\Pi).** The key generation algorithm samples $\chi, \gamma \leftarrow \mathbb{Z}_p$ and computes $k = k_z^\chi k_r^\gamma \in \mathcal{G}_2$. The key is returned as $(vk, sk) \leftarrow (k, (\chi, \gamma))$.

**SIG.Sign(sk, m).** The sign algorithm first checks if $m$ has been already signed. If this is the case it recovers the bit $b_{vk,m}$ that was previously used to sign $m$. Else it samples $b_{vk,m} \leftarrow \{0,1\}$. Next, it proceeds as follows (recall that $m \in \mathcal{G}_1$).

1. Compute $z = m^{-\chi}$ and $r = m^{-\gamma}$.
2. Compute $\vec{v}_2 \in \mathcal{G}_1^2$ as
   \[
   \vec{v}_2 = (H(0||vk||m||b_{vk,m}), H(1||vk||m||b_{vk,m}))
   \]
   and set $\text{CRS} = (\vec{v}_1, \vec{v}_2)$.
3. Run the prove algorithm for relation $R$ (cf. Equation 5.2) and return
   \[
   \sigma \leftarrow \mathcal{S} \text{Prove(CRS, (k_z, k_r, k, m), (z, r))} \in \mathcal{G}_1^4 \times \mathcal{G}_2^2.
   \]

**SIG.Vfy(vk, m, \sigma).** The verification algorithm accepts iff

\[
\text{Vfy(CRS, (k_z, k_r, k, m), \sigma)}
\]

where $\text{CRS} = (\vec{v}_1, \vec{v}_2)$ and $\vec{v}_2 = (H(0||vk||m||0), H(1||vk||m||0))$ or $\vec{v}_2 = (H(0||vk||m||1), H(1||vk||m||1))$.

Correctness of the signature scheme follows from the correctness of the proof system.

\[\text{Note that we could also let the signer evaluate a pseudo-random function on } m \text{ to determine } b. \text{ According to [KW03] another very simple solution is to determine } b \text{ by evaluating a hash function } H' \text{ on } m \text{ and } vk \text{ (which again will be viewed as a random oracle by the analysis). This way the signer does not need to maintain states.}\]


5.3 Proof of Security

Here, we show that there is a tight reduction from breaking the SXDH-assumption to breaking the unforgeability of the above signature scheme.

**Theorem 7.** For any attacker $A$ that $(t, \mu, \ell, \epsilon_{\text{SIG}})$-breaks the MU-EUF-CMA-C-security of the above signature scheme $\text{SIG}$ there is an attacker $B = (B_{\text{XDH}}, B_{\text{DP}})$ such that $B_{\text{XDH}} (t_{\text{XDH}}, \epsilon_{\text{XDH}})$-breaks the XDH-assumption in $G_1$ or $B_{\text{DP}} (t_{\text{DP}}, \epsilon_{\text{DP}})$-breaks the double pairing assumption in $G_2$ with $t \approx t_{\text{XDH}} \approx t_{\text{DP}}$ and

$$
\epsilon_{\text{SIG}} < \frac{\ell^2}{2} \cdot p + 2 \cdot \left( \epsilon_{\text{XDH}} + \epsilon_{\text{DP}} + \frac{\mu + 1}{p} \right).
$$

The analysis will view $H$ as a random oracle.

**Proof.** The proof is built on the following fact: Given only the public key, there are many possible secret keys and the actual values of $\chi$ and $\gamma$ are information theoretically hidden. However, given a message and a secret key the pair $(z, r)$ is determined. That is, a given secret key allows to compute exactly one pair that satisfies equation 5.1. At the same time, even if the secret key is available, any other tuple that satisfies equation 5.1 allows to solve an instance of the DP$_2$ problem. We argue that since the signer commits to $(z, r)$ via hiding commitments the actual values $(z, r)$ are information theoretically hidden from the view of $A$. Therefore the secret key is also hidden from the adversary. Now, the reduction will manipulate $H$ to produce binding commitment keys for (almost) any adversarially generated signature. From this, we can extract a DP solution with probability $1 - \frac{1}{p}$.

The proof proceeds in a sequence of games. As above, we denote by $\chi_i$ the event that $A$ outputs $(i^*, m^*, \sigma^*)$ such that $i^* \notin S^* \land m^* \notin S_{\text{Sig}} \land \text{Sig.Vfy}(vk^*, m^*, \sigma^*) = 1$ in Game $i$. We will abbreviate $vk_{i^*}$ by $vk^*$.

**Game 0.** This game is the real MU-EUF-CMA-C-security game. When issued a hash-query for the string $s$ the reduction $R$ first checks if $s$ has already been hashed. If this is the case it returns the previously computed value $H(s)$. Otherwise it samples $r$ uniformly at random from $G_1$ and sets and returns
$H(s) = r$. All other queries are answered according to the MU-EUF-CMA-C-security experiment. This perfectly simulates the challenger in the random-oracle model. Thus, we have:

$$\Pr[\chi_0] = \epsilon_{\text{SIG}}$$

**Game 1.** Let $Q_{vk\text{coll}}$ denote the following event:

$$Q_{vk\text{coll}} := \{ \exists (i, j) \in [\ell]^2 : i \neq j \land vk_i = vk_j \}$$

In Game 1 $\mathcal{R}$ aborts (and $\mathcal{A}$ looses) if event $Q_{vk\text{coll}}$ occurs. Since $\chi$ and $\gamma$ are chosen uniformly at random by $\mathcal{R}$, public keys are distributed uniformly random over $G_2$ which implies $\Pr[Q_{vk\text{coll}}] = \frac{\ell \cdot (\ell - 1)}{2 \cdot p}$. Thus, we have

$$|\Pr[\chi_0] - \Pr[\chi_1]| \leq \frac{\ell^2}{2 \cdot p}$$

**Game 2.** Before we introduce the changes made in Game 2 let us fix some notation. Let $b_{vk,m}$ denote the bit that is (lazily) sampled by $\mathcal{R}$ during signing on $m$ under $vk$. In Game 2 $\mathcal{R}$ aborts if for the forgery $(i^*, m^*, \sigma^*)$ that is output by $\mathcal{A}$ it holds that $v_{2}^* = (H(0||vk^*||m^*||b_{vk^*,m^*}), H(1||vk^*||m^*||b_{vk^*,m^*}))$. In other words, $\mathcal{R}$ aborts (and $\mathcal{A}$ looses) if $\mathcal{A}$ chooses for the forgery the same bit $b_{vk^*,m^*}$ that $\mathcal{R}$ would have chosen itself to sign $m^*$ under $vk^*$. Since $\mathcal{R}$ chooses each bit uniformly at random the actual choice of $b_{vk^*,m^*}$ is information theoretically hidden from the view of $\mathcal{A}$ (recall that all $vk$ are distinct due to Game 1). Thus we have

$$\Pr[\chi_1] \leq 2 \cdot \Pr[\chi_2]$$

**Game 3.** In Game 3 the reduction proceeds similarly to Game 2 except for the following: $\mathcal{R}$ lazily programs the hash-oracle such that for every pair of $m$ and $vk$ we have that $(H(0||vk||m||1 - b_{vk,m}), H(1||vk||m||1 - b_{vk,m})) \in \text{DDH}(g_1, f_1)$. By the random self reducibility of $\text{XDH}$ we get:

$$|\Pr[\chi_2] - \Pr[\chi_3]| < \epsilon_{\text{XDH}}$$
5.3 Proof of Security

Game 4. This game is similar to Game 3 except that \( R \) aborts (and \( A \) looses), if for any sign query \((m, i)\) issued by \( A \) during the security experiment we have that \((H(0||vk_i||m||b_{vk_i,m}), H(1||vk_i||m||b_{vk_i,m})) \in \text{DDH}(g_1, f_1)\).

Since images of \( H \) are distributed uniformly over \( \mathcal{G} \) we have that

\[
\Pr[\chi_3] - \Pr[\chi_4] \leq \frac{\mu \cdot \ell}{p}
\]

Game 5. Game 5 proceeds exactly like Game 4 except for the following. \( R \) aborts if it cannot extract a satisfying assignment for equation 5.1 from \( \sigma^* \). Due to Game 3 we know that \((\hat{g}_1, \hat{f}_1) = (H(0||vk^*||m^*||1 - b_{vk^*,m^*}), H(1||vk^*||m^*||1 - b_{vk^*,m^*})) \in \text{DDH}(g_1, f_1)\). Therefore \((\vec{v}_1, \vec{v}_2) \) is in the (first) range of \( E_0(1^\kappa) \) and gives a perfectly binding CRS \( \text{GS08} \).

We show in detail how \( R \) proceeds. Let \((\vec{v}_1, \vec{v}_2) = ((g_1, f_1), (\hat{g}_1, \hat{f}_1))\) and let \( \tau = \log_{g_1}(f_1) \). Then

\[
C_z = \left( g_1^{\delta z,1} \cdot \hat{g}_1^{\delta z,2}, z \cdot f_1^{\delta z,1} \cdot \hat{f}_1^{\delta z,2} \right)
\]

is an ElGamal [ElG85] encryption of \( z \) that can be decrypted using \( \tau \). An analogous statement holds for \( C_r \). Thus, \( R \) is able to extract \((z^*, r^*)\) from the forged signature. Since \( \sigma^* \) is a valid signature for \( m^* \) we know that equation 5.3 is satisfied. Let \( C_{z^*} = (C_{z^*,1}, z^* \cdot (C_{z^*,1})^\tau) \) and \( C_{r^*} = (C_{r^*,1}, r^* \cdot (C_{r^*,1})^\tau) \). Then we can rewrite equation 5.3 as:

\[
e(1, k)^{-1} = e(C_{z^*,1}, k_z) \cdot e(C_{r^*,1}, k_r) \cdot e(g_1, \pi_1^*) \cdot e(\hat{g}_1, \pi_2^*)
\]

\[
\wedge e(m, k)^{-1} = e(z^* \cdot (C_{z^*,1})^\tau, k_z) \cdot e(r^* \cdot (C_{z^*,1})^\tau, k_r) \cdot e(g_1^\tau, \pi_1^*) \cdot e(\hat{g}_1^\tau, \pi_2^*)
\]
Now, the first equation is equal to 1. This implies for the second equation that:

\[
e(m, k)^{-1} = (e(C_{z,1}^*, k_z) \cdot e(C_{r,1}^*, k_r) \cdot e(g_1, \pi_1^*) \cdot e(\hat{g}_1, \pi_2^*))^T \cdot e(z^*, k_z) \cdot e(r^*, k_r) = 1^* \cdot e(z^*, k_z) \cdot e(r^*, k_r)
\]

Thus, we have:

\[
Pr[\chi_4] = Pr[\chi_5]
\]

**Game 6.** Game 6 proceeds exactly as Game 5 except for the following. The reduction aborts (and \(A\) loses) if for the forgery that \(A\) outputs it holds that the satisfying assignment of equation 5.1, \((z^*, r^*)\), that is extracted by \(R\) from \(\sigma^*\) is equal to \(((m^*)^{-\chi}, (m^*)^{-\gamma})\). Since for all sign queries \((m, i)\) issued by \(A\) it holds that \((H(0||vk_i ||m||b_{vk_i,m}), H(1||vk_i ||m||b_{vk_i,m})) \notin \text{DDH}(g_1, f_1)\) (which is due to Game 4) the signatures output by \(R\) are perfectly hiding proofs and do not leak any valuable information on \((z, r)\) that are used by \(R\) to compute the respective commitments. From the view of the adversary all \((z, r)\) that satisfy the respective equation 5.1 are equally likely. In particular the only information that the adversary obtains on \(\chi\) and \(\gamma\) comes from the public key. However the public key provides the adversary with one linear equation in two unknowns which has \(p\) possible solutions.

Though this follows directly from the properties of the GS proof system, we give an explicit proof here. Namely, each signature output by \(R\) provides \(A\) with the following set of linear equations (in the exponent):

\[
\begin{pmatrix}
-\log(k_z) & -\log(k_r) \\
\log(m) & \log(v_{1,1}) & \log(v_{2,1}) \\
\log(m) & \log(v_{1,2}) & \log(v_{2,2}) \\
-\log(k_z) & -\log(k_r)
\end{pmatrix}
\begin{pmatrix}
\chi \\
\gamma \\
\delta_{z,1} \\
\delta_{z,2} \\
\delta_{r,1} \\
\delta_{r,2}
\end{pmatrix}
= \begin{pmatrix}
\log(k) \\
\log(C_{z,1}) \\
\log(C_{z,2}) \\
\log(C_{r,1}) \\
\log(C_{r,2}) \\
\log(\pi_1) \\
\log(\pi_2)
\end{pmatrix}
\]

Now, let us denote the matrix on the lefthand side by \(A\). We denote the \(i\)-th row of \(A\) by \(A_i\). Now, let us consider the following equations:
5.4 Efficiency of our Scheme.

\[-\log(m) \cdot A_1 + \log(k_z) \cdot A_3 + \log(k_r) \cdot A_5 + \log(v_{1,1}) \cdot A_6 + \log(v_{2,1}) \cdot A_7 \]
\[-\log(k_z) \cdot A_2 + \log(k_r) \cdot A_4 + \log(v_{1,2}) \cdot A_6 + \log(v_{2,2}) \cdot A_7 \]

We observe that both equations evaluate to zero and conclude that \( \text{rank}(A) \leq 5 \) which implies that the adversary is actually given at most five equations in six variables. Thus we have:

\[|\Pr[\chi_5] - \Pr[\chi_6]| \leq \frac{1}{p} \]

**Lemma 5.1.** \( \Pr[\chi_6] < \epsilon_{\text{DP}_2} \).

*Proof.* We will show that any forgery output by the adversary in Game 6 allows \( \mathcal{B}_{\text{DP}} \) to solve a given instance of the \( \text{DP}_2 \) assumption. To this end, assume that \( \mathcal{A} \) outputs a valid signature \( \sigma^* \) for \( m^* \) that was not signed before under \( v_k \). By Game 5 we know that from \( \sigma^* \) we can extract \( (z^*, r^*) \) such that 

\[ 1 = e(k_z, z^*) \cdot e(k_r, r^*) \cdot (k^*, m^*) \]

Moreover due to Game 6 we know that \( (z^*, r^*) \neq (z, r) = ((m^*)^{-\chi}, (m^*)^{-\gamma}) \). In addition to that, we know that \( (z, r) \) also satisfies equation 5.1. Now, \( (\frac{z}{z^*}, \frac{r}{r^*}) \neq (1, 1) \) yields a solution to the \( \text{DP}_2 \) instance \( (k_z, k_r) \in G_2 \):

\[ e\left(\frac{z}{z^*}, k_z\right) \cdot e\left(\frac{r}{r^*}, k_r\right) = e(z, k_z) \cdot e(r, k_r) \cdot e(z^*, k_z)^{-1} \cdot e(r^*, k_r)^{-1} = e(z, k_z) \cdot e(r, k_r) \cdot e(m^*, k^*)^{1-1} \cdot e(z^*, k_z)^{-1} \cdot e(r^*, k_r)^{-1} = 1 \]

where the last equation is due to the fact that both, \( (z, r) \) and \( (z^*, r^*) \), satisfy equation 5.1. This completes our proof.

We stress that the reduction is able to reveal the secret key corresponding to a public key in every single game throughout the proof and is nevertheless able to extract a solution to a hard problem from a forgery. We do not even have to re-randomize publicly available values. That is, we can use \( k_z \) and \( k_r \), as well as \( v_1 \) from \( \Pi \) for all users. 

\[ \square \]
5 Practical Tightly Secure Signatures in the ROM

<table>
<thead>
<tr>
<th></th>
<th>$\ell = 2^{16}$</th>
<th></th>
<th>$\ell = 2^{45}$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$</td>
<td>vk</td>
<td>$</td>
<td>$</td>
</tr>
<tr>
<td>BLS</td>
<td>$\approx 1510$</td>
<td>$\approx 256$</td>
<td>$\approx 4900$</td>
<td>$\approx 310$</td>
</tr>
<tr>
<td>RSA PSS</td>
<td>$&gt; 1792$</td>
<td>$\approx 1792$</td>
<td>$&gt; 3150$</td>
<td>$\approx 3150$</td>
</tr>
<tr>
<td>Ours</td>
<td>$1248$</td>
<td>$3156$</td>
<td>$1248$</td>
<td>$3156$</td>
</tr>
</tbody>
</table>

Figure 5.1: Comparison of our random oracle based scheme and random oracle signature schemes from the literature.

5.4 Efficiency of our Scheme.

Figure 5.1 compares our signature scheme to random oracle signature schemes from the literature when 80 bits of security are required.

**BLS:** The BLS signature scheme [BLS01] is a Full Domain Hash signature scheme that supports very short signatures. A tight reduction was first ruled out by Coron [Cor02]. Our result from Chapter 3 confirms this result, namely, that any reduction from breaking the CDH-assumption to breaking the EUF-CMA-security of the BLS signature scheme must lose a factor of $\mu$, the number of sign-queries.

**RSA PSS:** RSA PSS was proposed in [BR96]. A tight reduction to the RSA-assumption was given by Coron [Cor02].

We compare public key size and signature size in bits when parameters are selected to obtain 80 bits of security in a theoretically sound way (i.e., parameter selection considers the security loss) following Ecrypt recommendations [ECR12], cf. Section 2.3.2. Following [BR96] we assume $\mu = 2^{30}$ sign-queries per public key. The BLS and our scheme do also require common public parameters. These are omitted in our comparison since they have to be stored only once by each user.

We observe that if the number of users is $2^{16}$ then the signature size of our scheme is roughly twice the size of an RSA PSS signature and 10 times the size of a BLS signature. Our scheme outperforms RSA PSS in both, public key size and signature size, if the number of public keys is about $2^{45}$ which is a very
5.4 Efficiency of our Scheme.

<table>
<thead>
<tr>
<th>Security Parameter</th>
<th>Signature length [bits]</th>
<th>Signing [ms]</th>
<th>Verifying [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>$\approx 2640$</td>
<td>$\approx 16$</td>
<td>$\approx 211$</td>
</tr>
<tr>
<td>128</td>
<td>$\approx 4160$</td>
<td>$\approx 36$</td>
<td>$\approx 447$</td>
</tr>
<tr>
<td>256</td>
<td>$\approx 8257$</td>
<td>$\approx 173$</td>
<td>$\approx 1980$</td>
</tr>
</tbody>
</table>

Figure 5.2: Efficiency Evaluation of our Signature Scheme.

large number. However, even in this case, BLS signatures are shorter than our signatures. Therefore, for most of today’s practical applications our scheme is no better than known solutions. Nonetheless, due to the loss of the generic reduction we find it interesting in its own right to construct a signature scheme with tight MU-EUF-CMA-C security.

**Implementation**  We have implemented our signature scheme in Java on an Intel i7-2760QM CPU 2.40 GHz over Baretto-Naehrig Curves [BN06, PSNB11] using the library bnpairings [PBJ12] of the authors of [PSNB11]. The evaluation can be found in Figure 5.2. For each security level, we have generated 100 keys and signed the message “test” once with each key to verify the signature. The table shows the time it took in milliseconds to sign the message and to verify the signature and the signature length in bits, averaging over these instances. Using point compression [BN06], our signatures are slightly shorter than estimated in Figure 5.1.
Conclusion

In this thesis we considered tight reductions in cryptography. Chapter 3 dealt with properties of an implementation of a cryptographic primitive that rule out a tight reduction. We considered security experiments that induce some relation over statements and witnesses and that may be broken by the adversary if it is able to compute a “fresh” witness. To rule out a tight security proof we required the relation to be efficiently computable during the security experiment and witnesses to be efficiently re-randomizable. An interesting open problem is whether these conditions are also necessary to rule out tight reductions.

We have discussed how known implementations of primitives that come along with an (almost) tight security proof circumvent our result. Here, we focused on the re-randomization property. However, it may also be possible to circumvent our result when the security experiment of the considered primitive does not allow efficient verification, e.g., when the considered primitive is a pseudorandom function or a message authentication code. For PRFs a possibility result is the classic Naor-Rheingold PRF [NR97]. Further research could analyze this property in the light of our new work. Moreover, it would be interesting to apply our result to further primitives.

In Chapters 4 and 5 we showed how to construct signature schemes that are tightly secure in the multi user setting with respect to both, sign-queries and number of users. We proposed a tightness preserving compiler that is provably secure in the standard model, as well as a quite efficient solution in the ROM. While the ROM solution supports signatures, public keys and parameters of constant size, it is currently not clear if this can be achieved using our standard model compiler. Namely, one drawback of our compiler is that, when it is implemented with signature schemes that are tightly secure in the multi user setting without corruptions [HJ12, CW13] we either have large pub-
lic parameters or large signatures. An interesting open problem is to construct a signature scheme that is tightly secure in the multi user setting with corruptions from standard assumptions and that supports constant size signatures, public keys and parameters and that is provably secure in the standard model.
Bibliography


Bibliography


Bibliography


[HKS15] Dennis Hofheinz, Jessica Koch, and Christoph Striecks. Identity-based encryption with (almost) tight security in the multi-instance, multi-ciphertext setting, 2015. (Cited on page 4)

[HW09] Susan Hohenberger and Brent Waters. Short and stateless signatures from the RSA assumption. In


[KM15] Neal Koblitz and Alfred Menezes. The random or-
cacle model: A twenty-year retrospective. Cryptol-

[KPW13] Hugo Krawczyk, Kenneth G. Paterson, and Hoeteck Wee. On the security of the TLS protocol: A system-
atic analysis. In Ran Canetti and Juan A. Garay, ed-
itors, CRYPTO 2013, Part I, volume 8042 of LNCS,

[KSS13] Florian Kohlar, Sven Schge, and Jrg Schwenk. On the security of tls-dh and tls-rsa in the stan-

[KW03] Jonathan Katz and Nan Wang. Efficiency improve-
ments for signature schemes with tight security re-
ductions. In Sushil Jajodia, Vijayalakshmi Atluri, and Trent Jaeger, editors, ACM CCS 03, pages 155–
164. ACM Press, October 2003. (Cited on pages 5, 6, 7, 8, 61, 75, 76, and 80)

[KY03] Jonathan Katz and Moti Yung. Scalable proto-
cols for authenticated group key exchange. In Dan Boneh, editor, CRYPTO 2003, volume 2729 of LNCS,

[LJYP14] Benoit Libert, Marc Joye, Moti Yung, and Thomas Peters. Concise multi-challenge cca-secure encryp-
tion and signatures with almost tight security, 2014. (Cited on page 4)

[LPJY14] Benoit Libert, Thomas Peters, Marc Joye, and Moti Yung. Non-malleability from malleability: Simulation-sound quasi-adaptive nizk proofs and cca2-secure encryption from homomorphic signa-
tures. In Phong Nguyen and Elisabeth Oswald,


[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Richard E. Ladner and Cynthia Dwork, editors, 40th ACM STOC,
Bibliography


Christoph Bader

Geburtsdatum: 23.06.1987
Geburtsort: Bielefeld

Ausbildung

2006 Abitur, Niklas Luhmann Gymnasium, Oerlinghausen.

Anstellungen

2012 - wissenschaftlicher Mitarbeiter, heute Lehrstuhl für Netz- und Datensicherheit, Ruhr-Universität Bochum.

✉ christoph.bader@rub.de
Wissenschaftliche Konferenzveröffentlichungen

New Modular Compilers for Authenticated Key Exchange, with Yong Li, Sven Schäge, Zheng Yang and Jörg Schwenk, in Proceedings of the 12th International Conference, ACNS 2014.


Tightly Secure Authenticated Key Exchange, with Dennis Hofheinz, Tibor Jager, Eike Kiltz and Yong Li, in Proceedings of the 12th International Conference, TCC 2015.

Sonstige wissenschaftliche Veröffentlichungen


Konferenzteilnahmen & Vorträge

Juni 2014 Teilnahme an der Konferenz ACNS 2014 in Lausanne, Schweiz,
Vortrag: New Modular Compilers for Authenticated Key Exchange.

September 2014 Teilnahme an der Konferenz ESORICS 2014 in Breslau, Polen.

Oktober 2014 Teilnahme an der Konferenz CANS 2014 in Heraklion, Griechenland,

März 2015 Teilnahme an der Konferenz TCC 2015 in Warschau, Polen,
Vortrag: Tightly Secure Authenticated Key Exchange.

April 2015 Teilnahme am Workshop on Cryptography 2015 in Bochum,
Vortrag: Sufficient Conditions for the Impossibility of Tight Reductions in Cryptography.

Lehrtätigkeiten

Ruhr-Universität Bochum
Netz sicherheit, Computernetze und Kryptographische Protokolle,
Hochschule Albstadt
Sigmaringen
Kryptographie,
Wintersemester 2014,
Gastdozent.

Hochschule Bonn-Rhein-Sieg
Fortgeschrittene Angriffstechniken im Web,
Sommersemester 2015,
Gastdozent.

✉ christoph.bader@rub.de