New Limits of Provable Security and Applications to ElGamal Encryption

Sven Schäge [*]

Eindhoven University of Technology
s.schage@tue.nl

Abstract. We provide new results showing that ElGamal encryption cannot be proven CCA1-secure – a long-standing open problem in cryptography. Our result follows from a very broad, meta-reduction-based impossibility result on random self-reducible relations with efficiently re-randomizable witnesses. The techniques that we develop allow, for the first time, to provide impossibility results for very weak security notions where the challenger outputs fresh challenge statements at the end of the security game. This can be used to finally tackle encryption-type definitions that have remained elusive in the past. We show that our results have broad applicability by casting several known cryptographic setups as instances of random self-reducible and re-randomizable relations. These setups include general semi-homomorphic PKE and the large class of certified homomorphic one-way bijections. As a result, we also obtain new impossibility results for the IND-CCA1 security of the PKEs of Paillier and Damgård–Jurik, and many one-more inversion assumptions like the one-more DLOG or the one-more RSA assumption.

1 Introduction

The ElGamal public-key encryption (PKE) scheme from 1984 is, besides RSA, the most well-known asymmetric encryption system today and its importance for the development of cryptography can hardly be overestimated [22]. It is conceptually simple and, since it supports the use of elliptic curves, can provide very high efficiency. In the past, ElGamal encryption has served as a fruitful template for the development of new schemes that are based on novel mathematical structures like bilinear pairings or lattices and, via further modifications, paved the way for new powerful primitives like IBE and functional encryption. Most PKE systems that are used in practice are enhancements of the basic ElGamal scheme that additionally armor it from strong active attacks. The scheme has become so much part of the cryptographic canon, that introductory academic courses on cryptography that do not cover ElGamal are hardly imaginable.

Security Guarantees of ElGamal PKE. ElGamal is well-known to be IND-CPA secure (under the DDH assumption) which states that an attacker A who is given a challenge ciphertext c^* cannot distinguish if it encrypts message m_0^* or m_1^*, even if both messages have been chosen by A. Unfortunately, it turns out that without further modifications ElGamal PKE cannot fulfill the stronger, standard notion of IND-CCA2 security, where the attacker is also allowed to query a decryption oracle at any point in the security game (with the restriction that the challenge ciphertext may not be queried). Essentially this is due to the malleability of the ciphertexts in ElGamal encryption, making it possible to query a slight modification c'^* of c^* to the decryption oracle and use the answer to c'^* to find the message within c^*. This impossibility result is unconditional and there is no hope to circumvent it. However, there is another widespread but slightly weaker notion of security called lunchtime attacks or IND-CCA1 security. According to this notion, the attacker is allowed to also query the decryption oracle but only *before* she receives the challenge ciphertext. Now the above attack cannot be applied anymore since decryption queries cannot depend on c^*. For decades, the exact state of ElGamal's security against lunchtime attacks has remained unclear and, despite the theoretical and practical importance of this scheme, hardly any progress towards

[*] The author has partially been supported by the CONFIDENTIAL6G project that is co-funded by the European Union (grant agreement ID: 101096435).

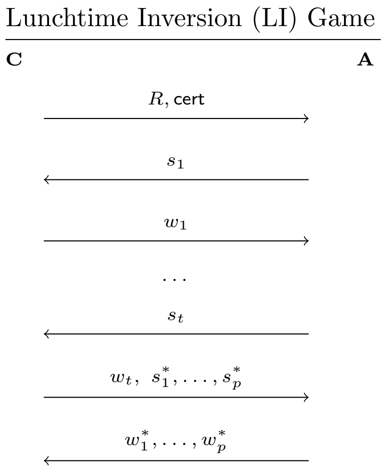

Fig. 1. Security game considered in this work. We consider a relation R with statements s and witnesses w that is used in a security game played between a challenger $\mathbf{C}$ and an attacker $\mathbf{A}$. The certificate cert proves that R is a re-randomizable and self-reducible relation. $\mathbf{A}$ has access to t adaptive inversion queries s_i and for each query receives back w_i such that $(s_i, w_i) \in R$. For every challenge statement s_i^*, $\mathbf{A}$ must output w_i^* such that $(s_i^*, w_i^*) \in R$.

that goal has been made. In this paper, we provide new, major contributions to the following long-standing open question.[1]

Can ElGamal PKE be shown IND-CCA1 secure?

We answer this question negatively in a very strong sense as detailed below.

Main Result. Our result on ElGamal encryption immediately follows from two fundamental results on random self-reducible and re-randomizable relations (RRRs) that are assumed to be hard to invert. In a nutshell, an RRR consists of a special relation R that is accompanied by a set of efficient algorithms for sampling, self-reducibility of statements, and re-randomizability of witnesses. A variant of RRRs that we call strong RRR additionally provides an efficient but indirect membership test. Although (strong) RRRs have a comparatively rich structure, we show that many cryptographic setups can be captured via the notion of RRRs. Remarkably, this includes general semi-homomorphic PKE (including ElGamal, Paillier, and Damgård–Jurik PKE) where statement/witness pairs correspond to ciphertext/plaintext pairs, and certified homomorphic one-way bijections (e.g., RSA, DLOG), where statement/witness pairs correspond to output/input pairs.

In our central result, we consider a security game (Figure 1) in which an attacker has to invert an RRR (i.e. find witnesses w^*) on a set of random challenge statements s^* (such that $(s^*, w^*) \in R$) that are specified at the *very end* of the security game. Deriving impossibility results for this challenging scenario has proven elusive in the past and our work is the first to provide an approach to this problem by exploiting the properties of RRRs. In a nutshell, our result shows that if the attacker is allowed to make $t+1$ inversion queries, we cannot have a security reduction that is based on a security assumption that allows up to t oracle calls (we call this a t-interactive complexity assumption or tICA for short). This is in some sense *optimal* as any security game with t oracle calls could easily be proven (tautologically) secure using security assumptions that allow t oracle calls — simply by assuming the security of the security game. We prove the following results.

Theorem 1 (First Main Result, Informal). *Let* RRR *be a RRR. Let* ICA *be a secure tICA. Then, there is no* simple[2] *Turing reduction that can reduce the lunchtime security of* RRR *with* $t+1$ *adaptive inversion queries to the security of* ICA.

[1] To provide some examples of explicit accounts on the importance of the question by well-known researchers, Yehuda Lindell calls the problem a "big open question" [32], whereas Helga Lipmaa terms it a "well-known open problem" [34].

[2] Simple reductions call the attacker only once and do not rewind.

Theorem 2 (Second Main Result, Informal). *Let* RRR *be a strong RRR. Let* ICA *be a secure tICA. Then, there is no (general) Turing reduction that, while creating up to u attacker instances, can reduce the lunchtime security of* RRR *with $t + u$ adaptive inversion queries to the security of* ICA.

One mere corollary of this result is that the number of calls to an inversion oracle granted to the attacker induces a complexity hierarchy of problems.[3] Transferred to semi-homomorphic PKE this says that its t-OW-CCA1 security (OW-CCA1 security when granted up to t decryption queries) cannot be based on its $(t-1)$-OW-CCA1 security. For $t = 1$ this already separates IND-CPA security from 1-OW-CCA1 security, i.e. OW-CCA1 security with only one-time access to a decryption oracle. However, we stress that our result on semi-homomorphic PKE is much more general than that. Since OW-CCA1 security is weaker than IND-CCA1 security, our result shows that semi-homomorphic PKE, and in particular ElGamal PKE, cannot be shown $(t+1)$-IND-CCA1 under *any* tICA under simple reductions. To obtain our results we have to overcome several challenges.

Challenge 1: Passing the Barrier towards Weaker Security Notions. We emphasize that, in general, the security game that we examine in our main impossibility result as depicted in Figure 1 considers an extremely weak security notion. However, for our purpose, this is of course advantageous as our impossibility result transfers to any security notion that is strictly stronger. Previous impossibility results that use related techniques already considered remarkably weak security notions (Figure 2). The weakest of them have in common that they formulate a so-called one-more inversion problem, where the attacker is given $t + 1$ challenge statements at the beginning of the security game together with subsequent t-time access to an inversion oracle [39]. However, our security notion is even weaker than that, and in fact, passing this barrier towards a weaker security notion has been the main obstacle to achieving further progress. In the notion that we consider, the attacker still has t-time access to an inversion oracle. However, the attacker's challenge statements are only decided on at the end of the security game by the challenger. In particular, all queries to the inversion oracle are thus independent of the final statement. For this challenging case, we show how to derive impossibility results based on the meta-reduction approach when dealing with efficiently random self-reducible and re-randomizable relations.

Proof Framework. Our result is obtained using the meta-reduction technique that was introduced by Boneh and Venkatesan [9]. The proof is by contradiction, where we start to simply assume that an appropriate PPT reduction B exists. According to the meta-reduction methodology, we first specify an ideal attacker that breaks the security game (with unbounded resources). Since the attacker A is successful, the reduction B must be able to use it to break some complexity assumption C with only a polynomial overhead in running time and an at most polynomial loss of success probability. In the second step, we show how to efficiently simulate the attacker towards the reduction. To this end, we define an efficient meta-reduction M that runs and controls the reduction algorithm. Crucially, we have to show that from B's perspective, M behaves indistinguishably from A. This shows that the combination of B and M breaks the underlying security assumption C efficiently, thus contradicting the starting assumption. From a more abstract angle, this approach shows that the reduction, if it exists, has already the power to break the security assumption itself.

Proof Idea. Our result holds for RRRs, relations that are random self-reducible and re-randomizable. On a high level, we intuitively exploit that any output of the reduction, and in particular any challenge statement, can be blinded and then relayed back to the reduction B as a query – without B noticing. More concretely, our notion of random self-reducibility guarantees that we can generate from any challenge statement s_i^* a derived statement $s_i'^*$ such that, crucially, the reduction has no way of recognizing that $s_i'^*$ was constructed from s_i^*. (Recall that to transfer the following discussion to the ElGamal setting, we simply can identify statements s with ciphertexts c and witnesses w with plaintexts m.) Next, the meta-reduction *rewinds* the reduction to some earlier point in the execution and sends $s_i'^*$ as an inversion query to B which in turn responds with the corresponding witness $w_i'^*$. Since the relation is random self-reducible, the meta-reduction can compute from $w_i'^*$ a challenge witness $w_i''^*$ for s_i^*. However, as $w_i'^*$ has been computed by B, the challenge witness $w_i''^*$

[3] For a more general discussion of both results let us for simplicity consider $u = 1$ in the following.

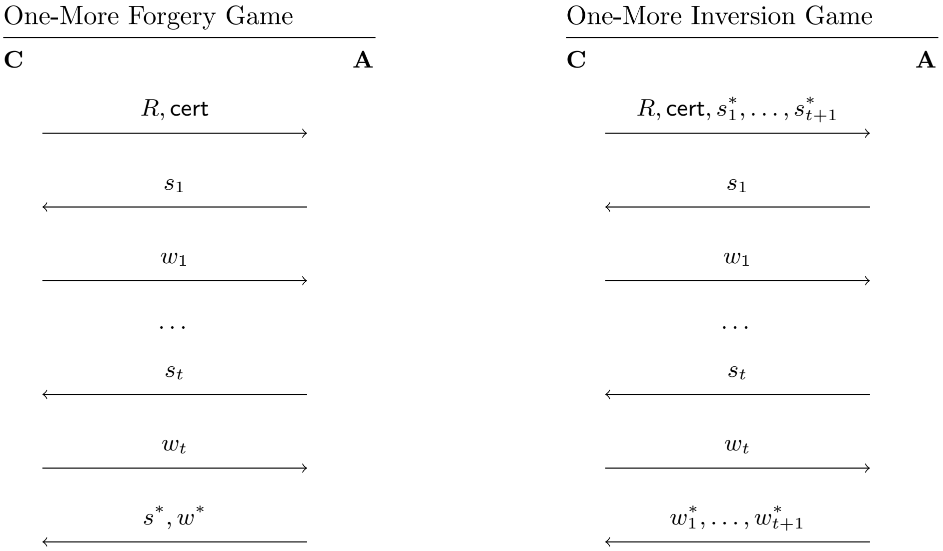

Fig. 2. Popular security notions that have previously been analyzed using meta-reductions. R is a relation with statements s and witnesses w that is used in a security game played between a challenger $\mathbf{C}$ and an attacker $\mathbf{A}$. cert proves that R has unique witnesses. $\mathbf{A}$ has access to t adaptive inversion queries s_i and for each query receives back w_i such that $(s_i, w_i) \in R$. For every challenge statement s_i^*, $\mathbf{A}$ must output w_i^* such that $(s_i^*, w_i^*) \in R$.

still might depend (in a way recognizable by the reduction) on $w_i'^*$. This is where we re-randomize $w_i''^*$ to obtain the final response w_i^* to s_i^*. We perform this last step since, similarly to before, the meta-reduction needs to make sure that the reduction is not able to recognize that the final output w_i^* has been computed from one of its responses. Here we exploit the properties of the re-randomization process, which state that for any witness pair, the output distributions of the re-randomization process are statistically close.

Certified Relations. So far, we have described how to ensure that the reduction cannot recognize any dependencies between the challenge statement/witness pair and the queried pairs of the relation. However, there is another important but rather subtle issue that we need to take care of. To closely model practice, we have included in our security model that the challenger sends the description of the relation to the attacker in the first step. Intuitively this may, for example, correspond to a public key in a cryptographic system along with the public system parameters. To make our technique work, the attacker needs to be able to efficiently verify that the so-specified relation is indeed an RRR. In practice, there might be several ways to ensure this. Most of the relations that we give as examples will allow to immediately verify the properties of the RRR. However, we may also consider relations where this is not possible and where instead relation descriptions are accompanied by appropriate (statistically or perfectly sound, non-interactive) proofs. Furthermore, we may consider trusted third parties that provide both parties with the relation such that the RRR properties are always fulfilled. This issue is indeed very subtle and in early works on related meta-reductions has not been addressed explicitly [31].

First Result: Exploiting RRRs. We present two main results. These results depend on whether the attacker may efficiently implement an (indirect) algorithm for deciding for given (s, w) if we have $(s, w) \in R$ or not. The first case assumes that no such algorithm exists. Our result for this case considers a more general relation type but achieves a weaker impossibility result. In particular, the impossibility result restricts the class of considered reductions considerably and only works for what is commonly called simple reductions where the reduction runs the attacker a single time without rewinding. Still, we believe that this rules out a very widespread type of reduction in cryptography. However, as we show, it becomes clear that if we allow the reduction to rewind the attacker even only once then this strategy becomes useless immediately. In a nutshell, the reason is that the reduction can use a simple strategy to distinguish the ideal attacker from the meta-reduction in case it can rewind the attacker, see Section 11. This strategy exploits that the meta-reduction does not know the witness corresponding to the modified statement that it sends to the reduction *after rewinding.* The ideal attacker, however, *always* knows all the witnesses to the queried statements.

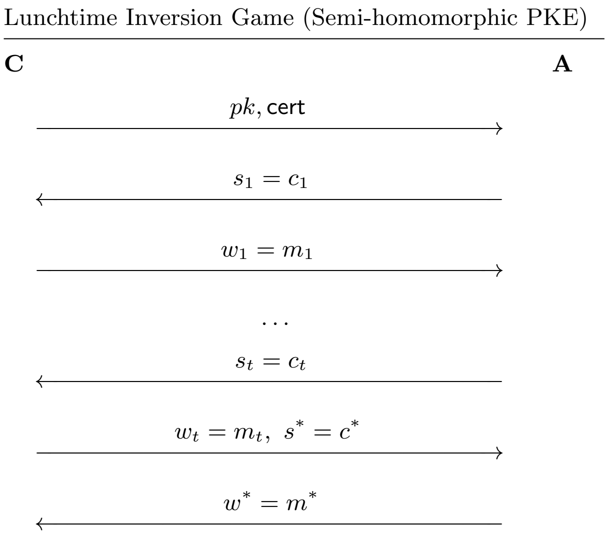

Fig. 3. Security definition when instantiating the first main result with RRRs based on semi-homomorphic PKE. Statements s correspond to ciphertexts c, while witnesses w correspond to plaintexts m. The resulting security game is equivalent to the OW-CCA1 security game for PKE.

Second Result: Exploiting Strong RRRs. In the second case, the RRR additionally admits an efficient algorithm called RSRTest that, given witness w, checks whether $(s, w) \in R$ where, importantly, s has in turn been derived from some given statement s^* using random coins r. We call such a relation a strong RRR. Intuitively, this algorithm can work as a weak form of membership test for values s that have been crafted in some specific way while having access to all the information that was used in the process of constructing s. (We stress that this does not constitute a direct, offline membership tester for R.)

Challenge 2: Dealing with Arbitrary Reductions. In contrast to before, our second main result rules out general (non-uniform) reductions. In this setting, we additionally have to deal with rewinding reductions that concurrently execute several instances of the attacker. In general, this is known to considerably complicate the analysis of impossibility results based on meta-reductions. And indeed, in our proof, we have to address several technical challenges.

One of these challenges is to ensure that a rewinding reduction cannot obtain more information from the meta-reduction than from the ideal attacker, for example by rewinding the attacker and answering a query once again, but this time with a distinct response. To this end, we extend recent techniques by [39] and [35] to guarantee that the attacker essentially behaves deterministically overall but independent of the concrete witnesses received as long as they are correct.

Another challenge is to avoid exponential blow-ups in the simulation time when dealing with rewinding reductions that execute several, u, attackers concurrently. To this end, we utilize some of the technical tools that were developed in [39] and which were in turn inspired by a line of research on resettable zero-knowledge [43,12,40,18,13]. However, our strategy fundamentally deviates from [39] and rather follows [35]. In particular, we do not describe a recursive rewinding strategy to deal with reductions that execute several attackers concurrently. Arguably, this makes our exposition more accessible since we do not have to formalize concepts like slots. At the same time, we obtain a result that is *concrete*. More precisely, let t' be the number of queries granted in the lunchtime inversion game and t be the number of queries in the security game of the interactive complexity assumption. Previous works like [39,47] require that the attacker makes $t' = \omega(\kappa + t + 1)$ queries for security parameter κ while requiring an upper bound on the number of overall queries M exchanged between the reduction and the attacker. Of course, bounding M implicitly also bounds u, the number of attacker instances created by the reduction. In contrast, our result solely requires an upper bound on the number of instances u created by the reduction while concretely assuming $t' = u + t$. For reductions that call a single attacker ($u = 1$), this now gives an optimal bound $t' = t + 1$, since for $t' \leq t$ we cannot hope to obtain a separation result from general interactive complexity assumptions with t queries.

Challenge 3: Casting Semi-Homomorphic PKE as a Strong RRR. To rule out general reductions via our second and main result, we also show how to cast ElGamal PKE as a strong RRR. Essentially this translates to an attacker that can check if the responses to inversion (decryption) queries are correct, although statement/witness pairs are in general not efficiently recognizable! This task is considerably more complicated than for mere RRRs and requires novel additional techniques. Solving this problem for general (semi-)homomorphic PKE is one of our main contributions.

Additional Encryption of MACed Plaintext. On a high level, we approach this problem by defining a strong RRR where statements consist of pairs of ciphertexts and witnesses of pairs of plaintexts. It is now possible to prepare pairs of ciphertexts such that the first one contains some message m while the second ciphertext contains a tag on m that is computed with a homomorphic MAC (under some freshly drawn MAC key) featuring statistical security (implemented as a pairwise-independent hash function). Since the MAC is homomorphic, this can be done even if the plaintext in the first ciphertext is unknown. Using this approach the attacker can check whether the reduction provides correct responses to the queries even for unknown ciphertexts. This is critical when relaying challenge statements back to the reduction in the presence of reductions that can rewind the attacker. Our final argument will show that when instantiating the lunchtime security game for RRRs with this strong RRR we end up in a very weak security game for (semi-homomorphic) PKE termed Paired OW-CCA1 that is (strictly) weaker than the OW-CCA1 security game. In particular, the main difference is that the Paired OW-CCA1 security game is less adaptive since pairs of ciphertext are queried to the decryption oracle. On the one hand, this makes our second impossibility result more general as we rely on a weaker security game. However, relying on strong RRRs also influences our bounds for PKE quantitatively, increasing the number of allowed decryption queries overall from $t + u$ to $2t + 2u$:

Corollary 9 (Semi-Homomorphic PKE under General Reductions). *Let* PKE *be a semi-homomorphic PKE. Let* ICA *be a tICA. Then, there is no (general) Turing reduction that, while creating at most u instances of the attacker, can reduce the Paired OW-CCA1 security of* PKE *with $2t + 2u$ decryption queries to the security of* ICA*.*

Broad Applicability: Certified Homomorphic One-Way Bijections and Semi-Homomorphic PKE. Our impossibility results are very general. The first application of our main theorems is to general semi-homomorphic PKE. This not only includes ElGamal PKE, but also the well-known public-key encryption system by Paillier and its generalization by Damgård–Jurik. To underline the broad applicability of our results further, we single out another practically important application that relies on certified homomorphic one-way bijections (CHOWB) like discrete exponentiation or the RSA permutation. It is not hard to show that these functions immediately give rise to suitable strong RRRs and our result applies. Since CHOWBs are so common in (asymmetric) cryptography, our result has a plethora of interesting applications (for example in the realm of one-more inversion assumptions, here improving on some existing impossibility results).

Scope and Interpretation. Our work provides fundamental impossibility results in the standard model under a very liberal notion of reduction where the only restrictions are i) that the reduction treats inefficient attackers in a black-box way (Turing reduction) and ii) that the number of queries allowed in the security assumption is strictly less than the number of queries in the security game. The latter condition seems necessary to rule out tautological results. The former condition states that our result does not exclude reductions that depend on the attacker. However, we stress that, although there has been considerable progress in recent years, it seems that current techniques for general non-black box reductions are still not able to tackle this problem without

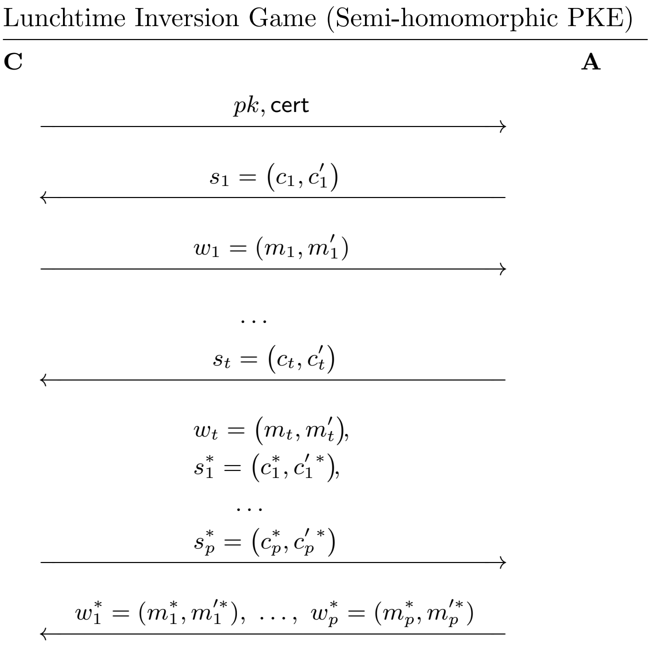
Lunchtime Inversion Game (Semi-homomorphic PKE)
C
A
pk, cert
$s_1 = (c_1, c'_1)$
$w_1 = (m_1, m'_1)$
...
$s_t = (c_t, c'_t)$
$w_t = (m_t, m'_t)$,
$s^*_1 = (c^*_1, {c'_1}^*)$,
...
$s^*_p = (c^*_p, {c'_p}^*)$
$w^*_1 = (m^*_1, m'^*_1), \ldots, w^*_p = (m^*_p, m'^*_p)$

Fig. 4. Security definition when instantiating the second main result with strong RRRs based on semi-homomorphic PKE. Statements are pairs of ciphertexts, witnesses are the corresponding plaintext pairs. The resulting security game is equivalent to the Paired OW-CCA1 security game for PKE.

major breakthroughs. At the same time, we note that there are several arguments for considering reductions that use attackers in a black-box sense as more natural — and in the context of security reductions even semantically more meaningful. One such argument that was for example invoked by Pass [39] is that such reductions will also work against attackers where parts of it are hard to be transformed into an explicit formal description, for example, if they consist of combinations of algorithms and humans. We emphasize that our result captures constructions that can rely on the underlying building block in a non-black-box manner (in contrast to, for example, most oracle separation techniques and idealized models like generic groups). Moreover, the reduction may also use the security assumptions in a non-black-box way. The only requirement that we have is that the reduction treats the attacker as black-box. In terms of the fine-grained hierarchy developed in [4], a refined version of [42], this corresponds to a so-called NBN reduction. We also emphasize that, as [4] have pointed out, if we only consider efficient attackers then knowledge of the code of the adversary does not lend additional power to the reduction (i.e. NBNa-type reductions exist iff NNNa-type do). So our result in fact only requires that *inefficient* attackers are treated in a black-box way by the reduction. This gives a somewhat complete picture. It is well-known that ElGamal can be proven CCA1 secure under some non-interactive assumption in idealized models like the algebraic group model (AGM), where the reduction also has access to the attacker's internal representation of group elements as shown in [26]. In the AGM, essentially *all* algorithms are assumed to (i) create new group elements X only via the application of the group operation to some fixed base B of group elements and (ii) always also output the internal representation of X with respect to B. We note that in ideal models like [26], the power to access the internal representation of *the attacker* specifically seems to be the main handle towards obtaining a security proof under non-interactive assumptions. Requiring access to the internal representation of the *reduction only* in contrast yields an impossibility result under non-interactive security assumptions [29]. We stress that [26] considers a specific and restricted attacker only that behaves according to (i) and (ii). For arbitrary attackers in contrast, it is still largely open how reductions can exploit non-black box access to attackers more generally to circumvent meta-reduction-based impossibility results.

Take-Home Message on ElGamal PKE. Conceptually, our arguments heavily exploit the random self-reducibility of ciphertexts in ElGamal PKE. In a sense, random self-reducibility can be viewed as a form of perfect malleability. In fact, it is so strong that no party, including the reduction, is able to recognize the original ciphertext given the derived one. Thus, one way to view our results on ElGamal is that:

While the malleability of the ciphertexts shows that ElGamal PKE is not IND-CCA2 secure, its "perfect" malleability shows its lack of provable IND-CCA1 security.

2 Related Work and Overview

ElGamal PKE [22] has been invented in 1984 and can be regarded as a non-interactive twist of the famous 1976 Diffie–Hellman protocol [19], where one of the exchanged public keys is made static. In particular, it is the first PKE scheme based on discrete-logarithm-type assumptions. Since then, many results have used or extended this scheme. Some important milestone results are the Cramer–Shoup encryption scheme that can be proven IND-CCA secure in the standard model [16], and the Boneh–Franklin scheme, the first identity-based encryption system in bilinear groups [8]. In the standard model, meta-reductions have been introduced by Boneh and Venkatesan [9] in 1998 and since then have proven a flexible and powerful technique for deriving strong impossibility results besides oracle separation. Previous works have used meta-reductions mostly to derive impossibility or efficiency results for concrete primitives like signatures schemes [21,20,37,27,44,5,24,25], and encryption systems [38]. Starting with the seminal work of Coron [15], many recent works like [31,30,3,35] use meta-reductions to prove bounds on the (non-exponential) security loss of cryptographic constructions for one-more-forgery type problems. Other results like [10,11] use meta-reductions to derive relationships among cryptographic one-more type problems. Probably closest to ours is the seminal work of Pass that showed a very general impossibility result on one-more inversion problems [39] with unique witnesses. As mentioned before, one of our main novelties is that by focusing on RRRs and a new simulation approach, we can provide results for a much weaker security notion that could not be covered by the previous techniques. In this way, our result is somewhat incomparable to [39] as it at the same time increases and decreases in generality albeit in distinct directions (since our security games are weaker but we have more restrictions on the relations by requiring them to be random self-reducible as well). Besides the already mentioned differences, our result provides several technical improvements that have not been considered in [39], like the broad treatment of re-randomizable relations via new and general notions. Finally, we point out that [39] does only hold for recognizable relations. Intuitively, this means that it requires the existence of Strong RRRs with explicit membership tests, to begin with. This makes it in general impossible to apply it to classical encryption-type definitions for probabilistic encryption where challenges correspond to ciphertexts and witnesses to plaintexts since the ability to recognize valid ciphertext/plaintext pairs could easily be used to simply break (even) IND-CPA security. Few papers have specifically worked on the impossibility of proving ElGamal PKE IND-CCA1 secure [46,34]. However, they all relate tICAs to ElGamal with t decryption queries. As stated, this may allow for tautological results:

- The result in [34] relates ElGamal PKE to DDH/CDH type assumptions. It shows a reduction from the security of ElGamal PKE with t decryption queries to a tICA. And indeed, the author himself regards this as a "tautology", "which is mainly useful to simplify further results".
- Similarly to this, [46] relates the OW-CCA1 security of ElGamal PKE with t decryption queries to a tICA called DTCDHA (or a combination of a weaker tICA called DTDLA and a knowledge type assumption).
- Again, the work in [1] offers similar results. Essentially, it shows that the IND-CCA1 security of semi-homorphic PKE schemes can be shown *equivalent* to an interactive assumption revolving around what they call Splitting Oracle-Assisted Subgroup Membership Problem. We note that this work does not make the number of queries explicit. However, the proof reveals that, as before, the CCA1 security of ElGamal PKE with t decryption queries is reduced to a tICA.

Finally, we stress that the meta-reduction framework has recently been extended in several directions. These extensions are typically quite generic such that they can often be applied to many works that use a similar meta-reduction technique. Some important stepping stones are the extension of the meta-reduction framework to rule out so-called memory-tight reduction as initiated in [2], or the application of the meta-reduction technique to non-uniform reductions [14].

Overview. After the introduction of some helpful notation in Section 3, we introduce semi-homomorphic public-key encryption and certified homomorphic one-way bijections in Section 4. Moreover, a series of progressively weaker security notions of PKE will be provided. The notions are characterized by an attacker that can make up to t' chosen-ciphertext queries. In Section 5, we turn our attention to the formal definition of so-callead RRR systems. Roughly, an RRR system consists of the description of several algorithms that operate on some RRR. Moreover, we introduce basic notions of correctness and soundness of these algorithms that capture precisely what properties we require from RRRs. Then, in Section 6, we provide concrete examples of important RRRs and strong RRRs. Importantly, we show how RRRs can be built from semi-homomorphic PKE and CHOWBs. In Section 7, we formally introduce our new, weak security notion for relations. It is the formalization of the security game given in Figure 1 (lunchtime inversion attacks with t adaptive queries). Next, in Section 8, we formally introduce our definition of t-interactive complexity assumptions, a security assumption where the attacker may interact with the challenger t times. Finally, with all these preparations, we are able to present our main results in the following sections. We essentially show that no RRR system can be proven secure with respect to the definition of lunchtime inversion attacks with t' adaptive inversion queries under a t-interactive complexity assumption if $t' > t$. Section 9 presents and proves our first main theorem. We provide interesting applications of this theorem in Section 10. Finally, in Section 11 we present our second main result. Important applications of our second main theorem can be found in Section 12. In Section 13, we sketch how our results can be transferred to non-uniform reductions. Finally, Section 14 discusses the practical value of our result for the construction of cryptographic systems. Moreover, it lists some of the general approaches that could be used to circumvent our impossibility results in cryptographic constructions.

For the benefit of the reader, we provide a standalone impossibility result of ElGamal PKE under simple reductions in Appendix B. This applies the arguments of our first main result in a simplified, concrete, and arguably more accessible setting.

3 Preliminaries

Relations. We use κ to denote the security parameter. Let S and W be sets and assume that there are efficient membership tests to check whether $s \in S$ and $w \in W$. We call S the set of statements and W the set of witnesses. Assume we have a relation $R \subseteq S \times W$. For any $s \in S$ let $W_s := \{w | (s,w) \in R\} \subseteq W$ be the *witness set* of s. Furthermore, we say that $s \in S$ has $t \in \mathbb{N}$ witnesses if $|W_s| = t$. If for a relation R we have that any statement $s \in S$ has at least one witness, we call it total. For simplicity, we will in the following only be interested in total relations. Moreover, we say that a relation is unique if for all statements s we have $|W_s| \leq 1$. We will use $[1;\ell]$ for $\ell \in \mathbb{N}$ to denote the set of natural numbers $1, \ldots, \ell$ and $[\ell]$ as a shorthand for $[1;\ell]$. In the following, we will always be interested in relations with superpolynomial statement set $|S|$.

Algorithms for Relations. We use $y = A(x;r)$ to denote a deterministic algorithm A that takes as input x and randomness r (for $r \in D = \{0,1\}^d$ with some polynomial $d = d(\kappa)$) to compute output y. Equally, we may make the random coins implicit, view A as a probabilistic algorithm, and simply write $y \leftarrow A(x)$ assuming that A is provided with uniformly random $r \in D$. In the following, we will for simplicity assume that every randomized algorithm uses random coins in $D = \{0,1\}^d$ for some appropriate polynomial d unless specified explicitly otherwise. If A is an algorithm and $(a_1, \ldots, a_\ell) \leftarrow A(x)$ for $\ell \in \mathbb{N}$ we use $[A(x)]_i$ for $i \in [\ell]$ to denote the ith output a_i. Similarly, we use $[A(x)]_{[i;j]}$ to denote the projection of the output of $A(x)$ on the coordinates i up to j. In the same way, we use $[B]_i$ for any tuple $B = (b_1, \ldots, b_\ell)$ to denote the projection on the ith component b_i and $[B]_{[i;j]}$ for the projection $(b_i, \ldots, b_j)$ on components i to j. In case $s \in S^k$ and $w \in W^k$ are tuples, we more generally write $(s,w) \in R$ to denote that for all $i \in [k]$ it holds that $([s]_i, [w]_i) \in R$. If D is a set we use $r \leftarrow D$ to denote that r is drawn uniformly at random from D. Let X be some random variable that is drawn according to some discrete distribution D' over some set. We use $H_\infty(X) = k$ if $\max_x \Pr[X = x] = 2^{-k}$. In this case we also say that distribution D has min-entropy $H_\infty(X) = k$.

Reductions. Let us now explain what type of reductions we consider. We mainly follow [39] here. For an alternative, more general, and very formal treatment we refer to [4] where the reductions that we focus on are called NBN reductions. Commonly, these types of reductions are simply called Turing reductions. In a nutshell, a black-box reduction for basing the security of a primitive X on the hardness of a primitive Y, is a probabilistic polynomial-time oracle machine B such that B^A "breaks" Y, whenever the oracle A (the attacker) "breaks" X. Typically a successful break is defined in a security game played between a challenger and an attacker with precise conditions under which the attacker wins. When the oracles are interactive, we only consider deterministic attacker oracles that may be given random coins as first input. If B gives the oracle A fresh random coins at the beginning of the security game, we also say that the reduction has *created an instance* (of A). Having explicit random coins given to a deterministic oracle allows general reductions to restart and "rewind" its oracle. Simple reductions, however, can only send these initial random coins to the oracle once. To model probabilistic reductions we assume that each reduction is given access to random coins $r_B \in D_B$. In the majority of this work, we concentrate on uniform algorithms and in particular uniform reductions. In Section 13 we sketch how our result could be transferred to non-uniform algorithms.

Success Probability after Rewinding. In our proofs we will make use of the well-known Splitting Lemma, see for example [41].

Lemma 1 (Splitting Lemma). *Let U and V be finite sets. Call $G \subseteq U \times V$ the set of good elements. For any element $(u, v) \in U \times V$ we define $k_u = |\{(u, v') \mid (u, v') \in G\}|$, i.e. k_u is the number of $v' \in V$ such that (u, v') is good. Suppose there is a lower bound on the number of good elements such that $|G| \geq \epsilon\ |U \times V|$. Define the set of super-good elements G' as a subset of G with $G' = \{(u, v) \in G \mid k_u \geq \epsilon/2\ |V|\}$. Then it holds that*

$$|G'| \geq \epsilon/2\ |U \times V|.$$

4 Notions for PKE and CHOWBs

In this section, we formally introduce semi-homomorphic public-key encryption together with a series of progressively weaker security notions. We will later use the notion of OW-CCA1 (Section 4.5) in a corollary of our first main result. We also introduce an even weaker notion called Paired OW-CCA1 security that will later be used when applying our second main result (Section 11) to semi-homomorphic PKE (Section 12).

4.1 Public-key encryption

A public-key encryption (PKE) system $\mathsf{PKE} = (\mathsf{PKEKGen}, \mathsf{PKEEnc}, \mathsf{PKEDec})$ consists of three algorithms.

1. $\mathsf{PKEKGen}(1^\kappa)$: the probabilistic key generator $\mathsf{PKEKGen}$ takes as input the security parameter in unary and outputs an asymmetric key pair with public key $pk \in \mathcal{PK}$ and secret key $sk \in \mathcal{SK}$.
2. $\mathsf{PKEEnc}(pk, m)$: the probabilistic encryption algorithm takes as input the public key $pk \in \mathcal{PK}$ and a message m from some message space $\mathcal{M}$ and outputs some ciphertext c in the ciphertext space $\mathcal{C}$.
3. $\mathsf{PKEDec}(sk, c)$: the deterministic decryption algorithm takes as input the secret key $sk \in \mathcal{SK}$ and a ciphertext c. It either outputs a message $m \in \mathcal{M}$ if $c \in \mathcal{C}$ or a dedicated error symbol $\perp$ if $c \notin \mathcal{C}$.

We say that PKE is perfectly correct if for all $(pk, sk) \leftarrow \mathsf{PKEKGen}(1^\kappa)$ we have for all $m \in \mathcal{M}$ that $\Pr[\mathsf{PKEDec}(sk, \mathsf{PKEEnc}(pk, m)) = m] = 1$. In the following, we will only be interested in perfectly correct PKE systems. We call a PKE system certified if i) there is an efficient algorithm that can check whether indeed $pk \in \mathcal{PK}$ and ii) it holds that $[\mathsf{PKEKGen}(1^\kappa; D)]_1 = \mathcal{PK}$, meaning that the set of public keys output by the key generator is efficiently recognizable.

4.2 Semi-Homomorphic PKE

We will particularly be interested in (semi-)homomorphic PKE. We say that the public key encryption scheme $\mathsf{PKE} = (\mathsf{PKEKGen}, \mathsf{PKEEnc}, \mathsf{PKEDec})$ is homomorphic if the following properties are fulfilled:

1. For all κ and all $(pk, sk) \leftarrow \mathsf{PKEKGen}(1^\kappa)$ it is possible to define (finite, cyclic) groups $(G, +)$ and $(H, \cdot)$ such that the set G is equal to the plaintext space $\mathcal{M}$ and H is equal to the ciphertext space $\mathcal{C}$. We require that membership in G and H can be efficiently tested.
2. $|G|$ grows exponentially in the security parameter.
3. There is an efficient algorithm that can draw $t \in \mathbb{N}$ uniformly from $t \leftarrow [|G|]$ with only statistically small error probability.
4. For any $m, m' \in G$ and any scalar $t \in [|G|]$ it holds that

$$\begin{aligned}\mathsf{PKEDec}(sk, \prod_{i=1}^{t} \mathsf{PKEEnc}(pk, m) \cdot \mathsf{PKEEnc}(pk, m')) &= \sum_{i=1}^{t} m + m' \\ &= tm + m'.\end{aligned}$$

5. We have that $\mathcal{C} = \mathsf{PKEEnc}(pk, \mathcal{M}; D)$, that is, for every pk and each ciphertext c in the ciphertext space there exists message m and randomness r such that $c = \mathsf{PKEEnc}(pk, m; r)$.
6. For uniformly random message $m \in \mathcal{M}$ and uniformly random $r \in D$, the value $\mathsf{PKEEnc}(pk, m; r)$ is distributed like a uniformly sampled ciphertext.

Property 3 allows us to not only model groups G of known order where we can directly draw $t \leftarrow [|G|]$ but also groups of unknown order like RSA (or Paillier) setups where randomly drawing from $[(N-1)/4]$ is statistically close to drawing from $[\phi(N)]$.

It is well-known that ElGamal PKE is a semi-homomorphic PKE. However, in the cryptographic literature, we can find several other well-known PKE systems that fulfill our definition of homomorphic PKE. Among them, there is the well-known PKE by Paillier [36] and its generalization by Damgård–Jurik [17] as well as the linear encryption scheme [7] and its generalizations [23]. For brevity, we refer to the original papers for the description of the other schemes. These works also indicate that all required properties are fulfilled if the public keys are set up as defined in the scheme's description. For these cryptosystems, it is also relatively easy to actually provide cryptographic proofs cert showing that the key material has been set up as given in the description. Such a proof can thus ultimately guarantee that all the above properties are fulfilled. For example, for Paillier encryption such a corresponding proof simply boils down to proving that $gcd(N, \phi(N)) = 1$ as shown in [33]. If such a proof can be computed and verified efficiently, we call the PKE scheme *certified.* It is important to stress that, since we require that plaintexts and ciphertexts are efficiently recognizable, our definition excludes PKE systems like the Cramer–Shoup scheme [16] or its IND-CCA1-secure lite version where real ciphertexts $(g_1^r, g_2^r, \ldots)$ are indistinguishable from "malformed" ciphertexts $(g_1^r, g_2^{r'}, \ldots)$ for $r' \neq r$ that the reduction can produce. Another interpretation of this condition is that we require that the validity of ciphertexts can be *publicly* verified.

4.3 ElGamal PKE

For concreteness, let us now also provide a description of ElGamal PKE.[4]

1. $\mathsf{PKEKGen}(1^\kappa)$: the algorithm computes a random prime q with polynomial, in κ, bitlength $|q|_2$ and a generator g of an order q group $\langle g \rangle = G$. Next, it computes $x \in \mathbb{Z}_q$ and outputs $pk = (g, h = g^x, q)$ and $sk = x$.
2. $\mathsf{PKEEnc}(pk, m)$: given message $m \in G$ this algorithm draws random $r \in \mathbb{Z}_q$ and outputs ciphertext $c = (c_1, c_2) = (g^r, m \cdot h^r) \in G^2$.
3. $\mathsf{PKEDec}(sk, c)$: this algorithm computes given $c = (c_1, c_2) \in G^2$ the plaintext as $m = c_2 / c_1^x$.

We note that we could also generate the values q, G, g beforehand in some global setup phase. All our results for ElGamal still hold then. The important fact for our results is that given all values available to the attacker, every ciphertext will have a unique plaintext.

[4] We also remark that all our results can easily be transferred to the ElGamal KEM where encapsulations consist of just g^r while the keys are defined as h^r. This is because all the homomorphic properties still hold in this scenario: using encapsulation $g^r \cdot g^{r'} = g^{r+r'}$ will result in key $h^r \cdot h^{r'} = h^{r+r'}$.

4.4 IND-CCA1

Let us first consider the classical notion for indistinguishability-based chosen-ciphertext security IND-CCA1 security (aka. lunchtime security) with t' queries.

To this end, consider the following security experiment between challenger C and attacker A.

1. The challenger C computes $(pk, sk) \leftarrow \mathsf{PKEKGen}(1^\kappa)$ and sends pk to the attacker A.
2. The attacker A may query up to t' decryption queries to C. Each query consists of a ciphertext c_i for $i \in [1; t']$.
3. The challenger C responds to each such query with $m_i = \mathsf{PKEDec}(sk, c_i)$.
4. Finally, the attacker outputs a message m^* to C.
5. The challenger draws a random message $\hat{m}^*$ of the same size as m^*. Next, it draws a random bit b and sends $c_b \leftarrow \mathsf{PKEEnc}(pk, \bar{m}_b)$ to A where $\bar{m}_0 = \hat{m}^*$ and $\bar{m}_1 = m^*$ (i.e. c_b contains m^* iff $b = 1$, otherwise $\hat{m}^*$).
6. A responds with bit b'.

Intuitively, the attacker is successful if it can determine b from c_b better than guessing. We say that A wins if $|\mathsf{Pr}[b = b'] - 1/2|$ is non-negligible.

4.5 A Weaker Security Notion: OW-CCA1

Let us now consider a security notion that is weaker than IND-CCA1 security called one-way CCA1 security (OW-CCA1). The difference is that the attacker now has to compute a plaintext from some ciphertext instead of distinguishing the real plaintext from a random one. This notion will be used in our first impossibility result. In Figure 3 and Figure 6, we provide an overview when instantiating this notion with semi-homomorphic PKE and ElGamal PKE in particular.

1. The challenger C computes $(pk, sk) \leftarrow \mathsf{PKEKGen}(1^\kappa)$ and sends pk to the attacker A.
2. The attacker A may query up to t' decryption queries to C. Each query consists of a ciphertext c_i for $i \in [1; t']$.
3. The challenger C responds to each such query with $m_i = \mathsf{PKEDec}(sk, c_i)$.
4. Finally, C draws a random message $\hat{m}^*$ and sends $c^* \leftarrow \mathsf{PKEEnc}(pk, \hat{m}^*)$ to A.
5. A responds with $m^* \in \mathcal{M}$.

The attacker wins if $\mathsf{Pr}[m^* = \hat{m}^*]$ is non-negligible.

It is easy to see that this security game is weaker than IND-CCA1 security. This is simply because the attacker now has to compute a randomly chosen plaintext from a given ciphertext. Now, if the attacker can compute from a given ciphertext the plaintext it can easily win in the standard security game by decrypting c^* (and comparing with m^*).

4.6 An Even Weaker Notion: Paired OW-CCA1

We will now consider a notion of PKE security that is yet considerably weaker. In particular, we require that the i) attacker makes decryption queries in pairs and ii) the attacker has to output $p \geq 1$ many challenge plaintexts. This notion will be used in our second impossibility result. In Figure 4, we provide an overview using ElGamal PKE.

1. The challenger C computes $(pk, sk) \leftarrow \mathsf{PKEKGen}(1^\kappa)$ and sends pk to the attacker A.
2. The attacker A may query up to $\lfloor t'/2 \rfloor$ decryption queries to C. Each query consists of two ciphertext c_i and c'_i for $i \in [1; \lfloor t'/2 \rfloor]$.
3. C responds to each such query with messages m_i, m'_i such that $m_i = \mathsf{PKEDec}(sk, c_i)$ and $m'_i = \mathsf{PKEDec}(sk, c'_i)$.
4. Finally, the challenger draws random messages $\hat{m}^*_1, \ldots, \hat{m}^*_p$ and sends the ciphertext values $c^*_1 \leftarrow \mathsf{PKEEnc}(pk, \hat{m}^*_1), \ldots, c^*_p \leftarrow \mathsf{PKEEnc}(pk, \hat{m}^*_p)$ to A.
5. A responds with $m^*_1, \ldots, m^*_p \in \mathcal{M}$.

The attacker wins if the probability is non-negligible that for all $i \in [p]$ we have $m^*_i = \hat{m}^*_i$. This game is weaker than OW-CCA1 security since we overall have the same number of ciphertexts (or one less) that the attacker may query for decryption but the queries are less adaptive. Moreover, the attacker has to decrypt more challenge ciphertexts. In particular, if attacker A breaks this security game, then it will surely break the previous one, OW-CCA1 security as well.

4.7 Certified Homomorphic One-Way Bijections

For each κ let F be a family of efficiently sampleable homomorphic one-way bijections such that $|F|$ grows exponentially in security parameter κ. We assume that the input and output space of each $f \in F$ is efficiently sampleable. Moreover, we assume that all the functions in F are *certified* so that there exists an efficient algorithm which given some f outputs 1 if $f \in F$ and 0 otherwise. We assume that each $f \in F$ maps elements from finite group $(A, \oplus)$ to group $(B = \mathsf{range}(f), \odot)$ of the same order. We thus have $f(x \oplus y) = f(x) \odot f(y)$ for all $x, y \in A$. We assume that all group operations are efficient. Finally, let us also assume that the evaluation of f is efficient for all $f \in F$. We will now consider a security notion in which the attacker makes a polynomial number, $t = t(\kappa)$, of inversion queries before it has to compute the pre-images of p challenge outputs (Figure 5).

1. The challenger draws $f \in F$ and sends f to the attacker A.
2. The attacker A may query up to t inversion queries to C. Each query y_i consists of a value in the image of f.
3. The challenger C responds to each such query with input x_i such that $y_i = f(x_i)$. If $i = t$, the challenger additionally sends random outputs $\hat{y}_1^*, \ldots, \hat{y}_p^*$ to A.
4. A responds with the corresponding inputs $x_1^*, \ldots, x_p^*$.

The attacker wins and breaks the security of F if the probability that for all $i \in [p]$ we have $y_i^* = f(x_i^*)$ is non-negligible. There are many well-known examples of CHOWBs in cryptography like the discrete exponentiation function or the (certified) RSA permutation. The above security notion formalizes a form of adaptive security for CHOWBs that is a very weak variant of a so-called one-more assumption where the challenge is given to the attacker only at the end of the security game.

5 Random Self-Reducible and Re-Randomizable Relations (RRRs)

In the following, we will define the class of relations that we are interested in. To this end, we formally specify several algorithms that work with some total relation R. After that, we will provide some detailed intuition for these algorithms and how we make use of them later. Additional intuition will be provided via the concrete examples of RRRs in Section 6.

5.1 Algorithms

To formally capture the intuition behind RRRs, we introduce the notion of RRR systems that comprise several efficient algorithms $\mathsf{RRR} = (\mathsf{RGen}, \mathsf{RSample}, \mathsf{RSubSample}, \mathsf{ReRand}, \mathsf{ReCheck}, \mathsf{RSRStatement}, \mathsf{RSRWitness}, \mathsf{RVerify})$. We parameterize our notion by some integer k which we call the *fan-out* of RRR. We will in the following assume that $k = 1$ unless explicitly specified otherwise. We also introduce the notion of strong RRR systems that have an additional efficient algorithm RSRTest: $\mathsf{RRR}' = (\mathsf{RGen}, \mathsf{RSample}, \mathsf{RSubSample}, \mathsf{ReRand}, \mathsf{ReCheck}, \mathsf{RSRStatement}, \mathsf{RSRWitness}, \mathsf{RSRTest}, \mathsf{RVerify})$.

- $\mathsf{RGen}(1^\kappa)$: The probabilistic *relation generator* RGen takes as input the security parameter κ and outputs the description of a relation $R \subseteq S \times W$ along with a proof $\mathsf{cert} \in U$. For simplicity, we assume that R implicitly also specifies the spaces S, W, U, and $S' \subseteq S$ with superpolynomial $|S'|$.
- $\mathsf{RSample}(R)$: The probabilistic *sampler* RSample outputs $(s, w) \in S \times W$ for uniformly random $s \in S$.
- $\mathsf{RSubSample}(R)$: The probabilistic *sub-sampler* RSubSample outputs $(s', w) \in R$, where $s' \in S'$ is drawn according to some distribution X with min-entropy $H_\infty(X) \geq \kappa$.
- $\mathsf{ReRand}(R, s, w)$: The probabilistic *re-randomizer* ReRand is given relation R and $(s, w) \in R$. It outputs a new witness $w' \in W$.
- $\mathsf{ReCheck}(R, s, w, w')$: The deterministic *checker* ReCheck is given relation R, $(s, w) \in R$, and $w' \in W$ as input. It outputs a bit c indicating whether $(s, w') \in R$.
- $\mathsf{RSRStatement}(R, s^*; r)$: The *statement transformer* RSRStatement is given as input relation R, a statement $s^* \in S$, and random coins r. The output is a vector of statements, $s \in S'^k$ for $k \in \mathbb{N}$, and the state $st = (s^*, r) \in S \times \{0,1\}^*$. We also say that s is the *transformed statement* of s^* (and r).

- $\mathsf{RSRWitness}(R, w, st)$: The deterministic *derivation algorithm* $\mathsf{RSRWitness}$ is given R, witness vector $w \in W^k$, and state $st = (s^*, r) \in S \times \{0,1\}^*$. It outputs a new witness $w^* \in W$. We also call w^* the *derivation* of w.
- $\mathsf{RSRTest}(R, w, k', st)$: The deterministic *tester* $\mathsf{RSRTest}$ is given R, a single witness $w \in W$, an index $k' \in [k]$, and a state $st = (s^*, r) \in S \times \{0,1\}^*$. The output is a bit t indicating whether the k'th component of the transformed statement of s^* and r has witness w.
- $\mathsf{RVerify}(R, \mathsf{cert})$: The deterministic *verifier* $\mathsf{RVerify}$ is given as input the description R and proof $\mathsf{cert} \in U$. The output is a bit v indicating whether the proof is correct.

If R is clear from the context, we might omit R as an input in our notation. If we do not specify an explicit value for cert, we implicitly assume that $\mathsf{cert} = \perp$ is set. Also, if $k = 1$ for some concrete RRR we may use $\mathsf{RSRTest}(R, w, st)$ short for $\mathsf{RSRTest}(R, w, 1, st)$.

Properties of R. Let us now, for some given relation R, define several properties that we might require from the algorithms of a (strong) RRR system.

P-1: Correctness of $\mathsf{RSample}$:

$$\forall r \in D : \mathsf{RSample}(R; r) \in R.$$

P-2: Correctness of $\mathsf{RSubSample}$:

$$\forall r \in D : \mathsf{RSubSample}(R; r) \in R.$$

P-3: Correctness of ReRand:

$$\forall (s, w) \in S \times W, r \in D : (s, w) \in R \Rightarrow (s, \mathsf{ReRand}(R, s, w; r)) \in R.$$

P-4: Witness indistinguishability of ReRand:

$$\forall s \in S, w, w', w'' \in W_s : \\ \mathsf{Pr}[w'' = \mathsf{ReRand}(R, s, w)] = \mathsf{Pr}[w'' = \mathsf{ReRand}(R, s, w')].$$

P-5: Correctness and soundness of $\mathsf{ReCheck}$:

$$\forall (s, w) \in R, w' \in W : \ \mathsf{ReCheck}(R, s, w, w') = 1 \Leftrightarrow (s, w') \in R.$$

P-6: Correctness of $(\mathsf{RSRStatement}, \mathsf{RSRWitness})$:

$$\forall s^* \in S, r \in D, w \in W^k : \\ ([\mathsf{RSRStatement}(R, s^*; r)]_{[k]}, w) \in R \\ \Rightarrow \mathsf{RSRWitness}(R, w, [\mathsf{RSRStatement}(R, s^*; r)]_{k+1}) \in W_{s^*}.$$

P-7: Statement indistinguishability of $\mathsf{RSRStatement}$:

$$\forall s^*, s' \in S : \\ [\mathsf{RSRStatement}(R, s^*)]_{[k]} \approx_s [\mathsf{RSRStatement}(R, s')]_{[k]}.$$

P-8: Indistinguishable sampleability of $\mathsf{RSubSample}$:

$$\forall s^* \in S, i \in [k] : \ [\mathsf{RSubSample}(R)]_1 \approx_s [\mathsf{RSRStatement}(R, s^*)]_i.$$

P-9: Statistical correctness and soundness of $(\mathsf{RSRStatement}, \mathsf{RSRTest})$:

$$\forall s^* \in S, w \in W, k' \in [k] \\ \mathsf{Pr}[(\mathsf{RSRStatement}(R, s^*; r)]_{k'}, w) \in R] \\ \approx_s \mathsf{Pr}[\mathsf{RSRTest}(R, w, k', [\mathsf{RSRStatement}(R, s^*; r)]_{k+1}) = 1], \\ \textit{where the probability is over } r \leftarrow D.$$

Let us finally define what a *certified* (strong) RRR system is.

Definition 1. *Let* RRR *be an RRR system. We say that* RRR *is a certified RRR system if properties P-10 to P-11 are fulfilled where:*

P-10: Correctness of RGen*:*

$$\forall r \in D : \mathsf{RGen}(1^\kappa; r) = (R, \mathsf{cert}) \Rightarrow \textit{P-1 to P-8 are fulfilled on input } R.$$

P-11: Soundness of $\mathsf{RVerify}$*:*

$$\forall R \subseteq S \times W, \mathsf{cert} \in U : \mathsf{RVerify}(R, \mathsf{cert}) = 1 \Leftrightarrow \textit{P-1 to P-8 hold on input } R.$$

Certified RRR systems will be utilized in our first result. In our second result, we need certified strong RRRs.

Definition 2. *Let* RRR' *be an RRR system. We say that* RRR' *is a certified strong RRR system if properties P-12 to P-13 are fulfilled where:*

P-12: Correctness of RGen*:*

$$\forall r \in D : \mathsf{RGen}(1^\kappa; r) = (R, \mathsf{cert}) \Rightarrow \textit{P-1 to P-9 are fulfilled on input } R.$$

P-13: Soundness of $\mathsf{RVerify}$*:*

$$\forall R \subseteq S \times W, \mathsf{cert} \in U : \mathsf{RVerify}(R, \mathsf{cert}) = 1 \Leftrightarrow \textit{P-1 to P-9 hold on input } R.$$

Intuition. Let us provide some intuition for these algorithms and the properties that we require. The pair of algorithms RGen and $\mathsf{RVerify}$, along with properties P-10 and P-11 (respectively P-12 and P-13) are used to capture that we can test if a given relation R is indeed a (strong) RRR. To this end, we have that RGen may output a corresponding proof cert alongside R that can be verified by $\mathsf{RVerify}$. We remark that this notion also captures situations where a dedicated proof is not necessary, such that the verifier can be convinced by performing some efficient computations on R. An example that will also underlie our analysis of ElGamal is when R specifies a generator g of cyclic group of prime order p. We assume that in these cases cert is set to $\mathsf{cert} = \perp$.

Our notion of re-randomizability of witnesses is captured via ReRand, $\mathsf{ReCheck}$, and P-4 to P-5. ReRand outputs when given $(s, w) \in R$ a new witness $\tilde{w} \in W_s$. At the same time, we guarantee a strong form of witness indistinguishability by requiring that the output distributions of $\mathsf{ReRand}(R, s, w)$ and $\mathsf{ReRand}(R, s, w')$, for any s and $w, w' \in W_s$, are equal. Finally, we have that the algorithm $\mathsf{ReCheck}$ can recognize, given arbitrary $(s, w) \in R$, if a new witness w' is also in W_s. As mentioned before, if R has unique witnesses the algorithm is trivial: it simply outputs 1 iff $w' = w$. In this case, we also say that the re-randomization of R is trivial.

To capture random self-reducibility we make use of $\mathsf{RSRStatement}$, $\mathsf{RSRWitness}$ along with P-6 and P-7. To model strong RRRs we also exploit $\mathsf{RSRTest}$ and require that P-9 is fulfilled. The algorithm $\mathsf{RSRStatement}$ simply constructs from statement s^* a derived vector of statements s. Similar to before, we have that the output of this algorithm does not reveal the input value s^*, capturing a strong form of statement indistinguishability. Moreover, we require that whenever we indeed obtain a witness w for s, we can use that to compute a witness w^* for the original statement s^*. Finally, for strong RRRs, we require that we have a weak membership test via $\mathsf{RSRTest}$ for derived s and arbitrary w. However, it requires as input the state st, which has been used to construct s from s^*. In contrast to $\mathsf{RSRWitness}$ that only works on a full vector $w' \in W^k$, we can run $\mathsf{RSRTest}$ with a single component $w \in W$ indexed by k'. Observe that nevertheless both algorithms are fed with the same state. In this way, we can use $\mathsf{RSRTest}$ to stepwisely test whether the components of $w' \in W^k$ are correct in the sense that a full vector w' can later indeed be used by $\mathsf{RSRWitness}$ to output witness w^*. We stress that property P-9 must hold for *all* possible witnesses w, including those that could be computed from the transformed statement $\mathsf{RSRStatement}(R, s^*; r)$. For most relations that we detail below, we have $k = 1$. However, for our final analysis of semi-homomorphic PKE, and in particular ElGamal PKE, we have that $k = 2$ (intuitively indicating that statements consist of pairs of ciphertexts).

Finally, we introduce two sampling algorithms $\mathsf{RSample}$ and $\mathsf{RSubSample}$. The first is used to generate relation pairs $(s, w) \in R$. The second is used to generate a new pair $(s', w') \in S' \times W$ such that $(s', w') \in R$. Importantly, the output distribution of this algorithm should be statistically close to that of the first k components of $\mathsf{RSRStatement}$ as encoded in P-8. So, using $\mathsf{RSubSample}$ we can output k distinct values that are distributed statistically close to the statements that are output by $\mathsf{RSRStatement}$. At the same time, the output s of $\mathsf{RSubSample}$ should have high min-entropy. This guarantees that we can generate hard instances of statements s such that it is difficult to compute w with $(s, w) \in R$ at all. In the algorithms that we consider we usually have that $\mathsf{RSubSample}(R)$ simply does the same as $\mathsf{RSample}(R)$ and we have that $S' = S$, but we also provide examples where this does not hold. In our security proof, the attacker will use $\mathsf{RSubSample}$ to generate pairs of the relation while the challenger may use either $\mathsf{RSubSample}$ or $\mathsf{RSample}$ to create statements.

6 Important RRRs

In the following, we will provide an important example of an RRR that is based on semi-homomorphic PKE. In Section 6.3 we show how we can easily obtain a strong RRR RRR_F from certified homomorphic one-way bijections when statement/witness pairs correspond to output/input pairs. We emphasize that all these RRRs are not built from concrete algebraic setups, like RSA groups, but from entire classes of building blocks ultimately leading to very broad impossibility results. For illustrative purposes we also present, in Appendix A, an example of an RRR with non-unique witnesses that is common in pairing-based cryptography. Moreover, the algorithms $\mathsf{RSample}$ and $\mathsf{RSubSample}$ of this RRR are distinct, showcasing the full range of cryptographic setups that are captured by RRRs.

6.1 RRRs from Semi-Homomorphic PKE

Let us now present an RRR, denoted $\mathsf{RRR}_{\mathrm{HomPKE}}$, that is derived from a semi-homomorphic PKE system PKE with associated groups $G = \mathcal{M}$ and $H = \mathcal{C}$. Intuitively, the statements correspond to ciphertexts while the witnesses are plaintexts. For simplicity, we assume that for each pk, PKE is certified via some cert, to begin with. To prove that all properties of RRRs are fulfilled, we exploit perfect correctness of the PKE such that for every ciphertext we have a unique plaintext.

- $\mathsf{RGen}(1^\kappa)$ calls $\mathsf{PKEKGen}(1^\kappa)$ to obtain secret key sk and public key pk. This specifies the relation of ciphertext/plaintext pairs.

$$\begin{aligned} R_{\mathrm{HomPKE}} &= \{(\ s, w = \mathsf{PKEDec}(sk, s)\ )\} \\ &= \{(\ \mathsf{PKEEnc}(pk, w; r), w\ )\}_{r \in D} \subseteq \mathcal{C} \times \mathcal{M}. \end{aligned}$$

  We have $S = S' = \mathcal{C}$ and $W = \mathcal{M}$. The output is R_{HomPKE}, i.e. pk, and cert.
- $\mathsf{RSample}(R_{\mathrm{HomPKE}})$ draws randomness $r \in D$ and random $w \in \mathcal{M}$ and outputs

$$(s, w) = (\ \mathsf{PKEEnc}(pk, w; r), w\ ).$$

- $\mathsf{RSubSample}(R_{\mathrm{HomPKE}})$ simply outputs $(s, w) \leftarrow \mathsf{RSample}(R_{\mathrm{HomPKE}})$.
- $\mathsf{ReRand}(s, w)$ outputs w.
- $\mathsf{ReCheck}(s, w, w')$ outputs 1 iff $w' = w$.
- $\mathsf{RSRStatement}(s^*; r)$ draws uniformly random $m \in \mathcal{M}$ and randomness $r' \in D$ to output

$$(s, st) = (\ (s^* \cdot \mathsf{PKEEnc}(pk, m; r'))\ , (s^*, r', m)\ )\ .$$

- $\mathsf{RSRWitness}(w, st)$ parses $st = (s^*, r', m)$ and outputs $w^* = w + (-m)$ where $-m$ is the inverse of m.
- $\mathsf{RVerify}(pk, \mathsf{cert})$ outputs 1 iff cert is a valid certificate for PKE.

Lemma 2. RRR_{HomPKE} *is a certified RRR.*

Proof. Observe that all algorithms are efficient, and due to perfect correctness for every statement there is a unique witness if R_{HomPKE} is certified. This makes re-randomization trivial. At the same time, $\mathsf{RVerify}$ perfectly tests whether a purported output of RGen is correct since PKE is certified. Finally, $\mathsf{RSRStatement}$ statistically hides s^* while $\mathsf{RSRWitness}$ perfectly recovers w^*. Since the output s of $\mathsf{RSRStatement}$ is uniformly random it is perfectly distributed like the output of $\mathsf{RSubSample}$. □

6.2 Strong RRRs from Semi-Homomorphic PKE

Let us now present an RRR, denoted $\mathsf{RRR}_{\mathrm{2HomPKE}}$, that is derived from a semi-homomorphic PKE system PKE. This transformation is one of our main contributions as it allows a proof under arbitrary reductions in the first place later on. We believe that the technique can be useful in many other contexts of cryptography as well. Intuitively, the statements now correspond to pairs of ciphertexts while the witnesses are pairs of plaintexts. Again, we assume that the PKE is certified. Moreover, we now rely on the fact that the group of plaintexts $|G| = |\mathcal{M}|$ has exponential size in the security parameter.

Before we begin, let us briefly provide some intuition for the novel techniques that we use to implement an online membership test. To provide some context, the technical challenge is to provide a technique such that the attacker can be sure that the reduction's responses are correct. At the most critical part of the proof, we need this to hold even if the attacker does not know the plaintext of one of the ciphertexts. The main idea is as follows: we make the attacker encrypt a message in the first ciphertext and the MAC tag of that message in the second one. The MAC is homomorphic and implemented via a two-wise independent hash function. Moreover, the MAC is a one-time MAC and its key material $u \in [|G|]$, $w' \in G$ is freshly drawn with each query. This gives statistical security. Crucially, exploiting the homomorphic properties, the attacker can use this technique to create from a given ciphertext s'^* encrypting w'^* a new ciphertext that encrypts the tag $uw'^* + w'$ *without knowledge of* w'^*. Of course, these two ciphertexts can also be blinded and, after rewinding, be sent to the reduction as a single query (recall that we are considering pairs of ciphertexts as statements). The other ciphertext in the challenge is treated in the same way. For each challenge ciphertext, the attacker so generates two ciphertexts, one for the blinded ciphertext and one for the MAC of this value, to extract a single challenge plaintext. We thus have fan-out $k = 2$.

- $\mathsf{RGen}(1^\kappa)$ calls $\mathsf{PKEKGen}(1^\kappa)$ to obtain secret key sk and public key pk. This specifies the relation of ciphertext/plaintext pairs:

  $$\begin{aligned} R_{2\text{HomPKE},sk} &= \{(\ (s', s''),\ (\mathsf{PKEDec}(sk, s'), \mathsf{PKEDec}(sk, s''))\ )\} \\ &= \{(\ s,\ w\ )\} \subseteq \mathcal{C}^2 \times \mathcal{M}^2 \end{aligned}$$

  with $s = (s', s'')$ and $w = (\mathsf{PKEDec}(sk, s'), \mathsf{PKEDec}(sk, s''))$. We have $S = S' = \mathcal{C}^2$ and $W = \mathcal{M}^2$.
- $\mathsf{RSample}(R_{2\text{HomPKE},sk})$ draws randomness $r', r'' \in D$ and random $w', w'' \in \mathcal{M}$ and outputs

  $$((s', s''), (w', w''))$$

  where

  $$s' \leftarrow \mathsf{PKEEnc}(pk, w'; r') \text{ and } s'' \leftarrow \mathsf{PKEEnc}(pk, w''; r'').$$

  We set $r = (r', r'')$ to denote the overall randomness and $w = (w', w'')$ for the witness.
- $\mathsf{RSubSample}(R_{2\text{HomPKE},x})$ simply outputs $(s, w) \leftarrow \mathsf{RSample}(R_{2\text{HomPKE},x})$.
- $\mathsf{ReRand}(s, w)$ outputs w.
- $\mathsf{ReCheck}(s, w, w')$ outputs 1 iff $w' = w$.
- $\mathsf{RSRStatement}(s^*; r)$ takes as input a pair of ciphertexts $s^* = (s'^*, s''^*)$. From that it generates four ciphertexts that are organized as pairs $\hat{s} = (\hat{s}', \hat{s}'')$ and $\tilde{s} = (\tilde{s}', \tilde{s}'')$ along with some state information. First, the algorithm draws randomness $\hat{r}'$ and random witness $\hat{w}'$ and blinds[5] the ciphertext s'^* to

  $$\hat{s}' = s'^* \cdot \mathsf{PKEEnc}(pk, \hat{w}'; \hat{r}').$$

  Next, we draw randomness $\hat{r}''$ and random witness $\hat{w}''$ as well as random scalar $\hat{u} \leftarrow [|G|]$ (or statistically close to that as required by our definition of semi-homomorphic PKE). The pair $\hat{u}$ and $\hat{w}''$ correspond to a key of a statistically secure, homomorphic MAC that is implemented with a pairwise independent hash function. Then, we compute the encrypted MAC by setting

  $$\hat{s}'' = \prod_{i=1}^{\hat{u}} (\hat{s}'^*) \cdot \mathsf{PKEEnc}(pk, \hat{w}''; \hat{r}'') = (\hat{s}'^*)^{\hat{u}} \cdot \mathsf{PKEEnc}(pk, \hat{w}''; \hat{r}'')$$

  using standard multiplicative notation. We proceed similarly for the second pair of ciphertexts and obtain

  $$\tilde{s}' = s''^* \cdot \mathsf{PKEEnc}(pk, \tilde{w}'; \tilde{r}')$$

[5] This perfectly hides s'^*.

and

$$\tilde{s}'' = \prod_{i=1}^{\tilde{u}} (\tilde{s}'^{*}) \cdot \mathsf{PKEEnc}(pk, \tilde{w}''; \tilde{r}'') = (\tilde{s}'^{*})^{\tilde{u}} \cdot \mathsf{PKEEnc}(pk, \tilde{w}''; \tilde{r}'').$$

The output is $s = (\hat{s}, \tilde{s})$, where $\hat{s} = (\hat{s}', \hat{s}'')$ and $\tilde{s} = (\tilde{s}', \tilde{s}'')$, and the state consists of s^* and all the random values drawn throughout the game, i.e., $st = (s^*, (\hat{w}', \hat{r}', \hat{w}'', \hat{r}'', \hat{u}), (\tilde{w}', \tilde{r}', \tilde{w}'', \tilde{r}'', \tilde{u}))$.

- $\mathsf{RSRTest}(w, k', st)$ parses w as $w = (\hat{w}, \tilde{w})$ with $\hat{w} = (\hat{w}_1, \hat{w}_2)$ and $\tilde{w} = (\tilde{w}_1, \tilde{w}_2)$ and the state st as $st = (s^*, (\hat{w}', \hat{r}', \hat{w}'', \hat{r}'', \hat{u}), (\tilde{w}', \tilde{r}', \tilde{w}'', \tilde{r}'', \tilde{u}))$. The algorithm first recomputes $s = (\hat{s}, \tilde{s})$ using $\mathsf{RSRStatement}$ where $\hat{s} = (\hat{s}', \hat{s}'')$ and $\tilde{s} = (\tilde{s}', \tilde{s}'')$. Next, the algorithm essentially verifies the MACs on the plaintexts received, depending on $k' \in \{1, 2\}$. If $k' = 1$ the algorithm checks if

$$\sum_{i=1}^{\hat{u}} \hat{w}_1 + \hat{w}'' = \hat{u}\hat{w}_1 + \hat{w}'' = \hat{w}_2$$

using standard additive notation. If $k' = 2$ the algorithm checks if

$$\sum_{i=1}^{\tilde{u}} \tilde{w}_1 + \tilde{w}'' = \tilde{u}\tilde{w}_1 + \tilde{w}'' = \tilde{w}_2.$$

On failure, it outputs 0. On success, it outputs 1.

- $\mathsf{RSRWitness}(w, st)$ parses $st = (s^*, (\hat{w}', \hat{r}', \hat{w}'', \hat{r}'', \hat{u}), (\tilde{w}', \tilde{r}', \tilde{w}'', \tilde{r}'', \tilde{u}))$ and $w = (\hat{w}, \tilde{w})$ with $\hat{w} = (\hat{w}_1, \hat{w}_2)$ and $\tilde{w} = (\tilde{w}_1, \tilde{w}_2)$. It computes $w'^{*} = \hat{w}_1 + (-\hat{w}')$ where $-\hat{w}'$ is the inverse of $\hat{w}'$ and $w''^{*} = \tilde{w}_1 + (-\tilde{w}')$. This undoes the blinding in the plaintext space. The output is $w^* = (w'^{*}, w''^{*})$.
- $\mathsf{RVerify}(u)$ outputs 1 iff the key is certified.

Lemma 3. $\mathsf{RRR}_{2HomPKE}$ *is a strong certified RRR.*

Proof. Most properties immediately follow from the previous analysis. Observe that all algorithms are efficient, and due to perfect correctness for every statement there is a unique witness if R_{2HomPKE} is certified. This makes re-randomization trivial. At the same time, RVerify perfectly tests whether a purported output of RGen is correct since PKE is certified. Finally, RSRStatement statistically hides s^* while RSRWitness perfectly recovers w^*. Since the output s of RSRStatement is uniformly random it is distributed like the output of RSubSample. Conceptually, RSRStatement blinds each of the challenge ciphertexts. Next, it adds to each of the so obtained ciphertexts an encryption of a (MAC) tag on the plaintext. Due to the homomorphic properties, this does also work in case the plaintext of the challenge ciphertexts are not known. Moreover, since we use a two-wise independent hash function, we obtain statistical security guarantees. The algorithm RSRTest essentially checks whether the MAC is correct. Let us concentrate on the first set of computations that focus on the first ciphertext. The remaining computations and arguments are analogous. To start off, observe that $\hat{u}$ is a perfectly hidden scalar: the value is only used once and at the same time $\hat{s}''$ is also fully randomized by $\hat{w}''$ (and $\hat{r}''$). We now need to show that it is hard to find two incorrect plaintexts, that nevertheless give a positive MAC verification. We show that this only works with small probability $1/|G|$ that is defined by the entropy of $\hat{u}$. To see this, observe that any incorrect plaintext pair $\hat{v}_1, \hat{v}_2$ must fulfill the same MAC equations as the correct one so that we not only have $\sum_{i=1}^{\hat{u}} \hat{w}_1 + \hat{w}'' = \hat{w}_2$ but also $\sum_{i=1}^{\hat{u}} \hat{v}_1 + \hat{w}'' = \hat{v}_2$. However, this amounts to $\sum_{i=1}^{\hat{u}} (\hat{w}_1 - \hat{v}_1) = \hat{u}(\hat{w}_1 - \hat{v}_1) = \hat{w}_2 - \hat{v}_2$. Since $\hat{u}$ is perfectly hidden, the success probability to find such a pair is equal to $1/|G|$, which by our definition of semi-homomorphic PKE is statistically small. To show the other direction, observe that if the algorithm is given the correct plaintexts, then each term on the left and right will always be equal to zero (the neutral group element) independent of $\hat{u}$. □

6.3 Strong RRRs from Certified Homomorphic One-Way Bijections

Let F denote an efficiently sampleable family of certified homomorphic one-way bijections as defined in Section 4.7. We will now construct an RRR system RRR_F with fan-out $k = 1$. Intuitively, statements correspond to outputs of f while witnesses correspond to corresponding inputs.

- $\mathsf{RGen}(1^\kappa)$ draws random certified f and outputs
$$R_f = \{(f(z), z)\} \subseteq B \times A.$$
We have that $S = S' = B$ and $W = A$.
- $\mathsf{RSample}(R_f)$ outputs $(f(x), x)$ for uniformly random $x \in A$.
- $\mathsf{RSubSample}(R_f)$ simply outputs $\mathsf{RSample}(R_f)$.
- $\mathsf{ReRand}(s, w)$ outputs w.
- $\mathsf{ReCheck}(s, w, w')$ outputs 1 iff $w' = w$.
- $\mathsf{RSRStatement}(s^*; r)$ uses random coins $r \in A$ to output
$$(s, st) = (s^* \odot f(r), (s^*, r)).$$
- $\mathsf{RSRWitness}(w, st)$ parses $st = (s^*, r)$ and outputs $w \oplus r^{-1}$.
- $\mathsf{RSRTest}(w, st)$ parses $st = (s^*, r)$, recomputes $(s, st) = \mathsf{RSRStatement}(R, s^*; r)$, and outputs 1 iff $s = f(w)$.
- $\mathsf{RVerify}(u)$ outputs 1 if f is indeed certified and 0 otherwise.

Lemma 4. RRR_F *constitutes a certified strong RRR system.*

Proof. Since f is certified, P-12 and P-13 are trivially fulfilled. Since f is efficient, we have that P-1 and P-2 are fulfilled. Also, we obtain an efficient (offline) membership test showing P-9. It simply checks whether $s = f(w)$. Since we consider bijections, pre-images are unique and re-randomization is trivial, guaranteeing P-3, P-4, and P-5. Finally, as r is uniform in A so is $f(r)$ in B and thus $s^* \odot f(r)$ perfectly blinds s^* what shows P-7. Observe that this blinding can always be removed showing P-6. Lastly, since the blinding is perfect this also shows P-8. □

This class contains some of the most useful bijections in cryptography like the (certified) RSA permutation, discrete exponentiations, or the evaluation of a non-degenerate bilinear pairing.

7 A New Weak Security Notion for Relations

Let us now formalize the main security notion that we will consider in this work (as depicted in Figure 1). In the rest of the paper, we always assume that t, p are polynomials in the security parameter unless specified otherwise. We state the security game using some of the syntax of certified RRRs. However, we stress that the security game is very general and stating it does not require the existence of most of the algorithms that are associated to RRR systems. However, our main results will only apply if we indeed have access to all the algorithms of RRR systems.

Security Game: Lunchtime Inversion with t Adaptive Queries. The central security notion that we consider for $t \in \mathbb{N}$, $t \geq 1$ is very simple. It is formalized as a game played between two algorithms, a challenger C and attacker A.

1. The challenger C calls $(R, \mathsf{cert}) \leftarrow \mathsf{RGen}(1^\kappa)$. The output is sent to A.
2. The parties repeat the following steps for all $i \in [1; t]$.
   2.i.1 The attacker A outputs statement $s_i \in S$.
   2.i.2 The challenger C responds with uniformly random witness $w_i \in W_{s_i}$ such that $(s_i, w_i) \in R$. If $i = t$ the challenger also sends challenge statement $s_1^*, \ldots, s_p^* \in S$.
3. The attacker outputs the challenge witness $w_1^*, \ldots, w_p^* \in W$.

We say that the attacker wins if for all $i \in [p]$ we have $(s_i^*, w_i^*) \in R$. If in any execution of the security game algorithm C finishes Step 2.i.2, we say that C *accepts* the $s_1, \ldots, s_i$.

To generalize our security notion, we also define the case for $t = 0$ where the attacker is not allowed to query any statement for its corresponding witness.

Security Game: Lunchtime Inversion with $t = 0$ (No) Adaptive Queries.

1. The challenger C calls $(R, \mathsf{cert}) \leftarrow \mathsf{RGen}(1^\kappa)$. The output is sent to the attacker A along with p challenge statements $s_1^*, \ldots, s_p^* \in S$.
2. The attacker outputs the challenge witnesses $w_1^*, \ldots, w_p^* \in W$.

Again, we say that the attacker wins if for all $i \in [p]$ we have $(s_i^*, w_i^*) \in R$.

Definition 3. *We say that certified (strong) RRR system* RRR *is secure against (lunchtime) inversion (attacks) with t adaptive queries if no PPT attacker A can have non-negligible success probability to win in the above security game.*

8 Interactive Complexity Assumption

t-Interactive Complexity Assumptions. To model interactive complexity assumptions with t queries we will rely on a generalization of the formulation introduced in [3] that originally focuses on non-interactive assumptions only. It is slightly more general than the notion in [39] which uses fixed thresholds. Intuitively, the notion we use allows arbitrary algorithms to implement a trivial guessing strategy. A t-interactive complexity assumption (tICA) consists of $t+3$ Turing machines $\mathsf{ICA} = (\mathsf{ICAGen}, \mathsf{ICAVer}, \mathsf{ICATriv}, \mathsf{ICAQuery}_1, \ldots, \mathsf{ICAQuery}_t)$.

- The efficient probabilistic instance generator $\mathsf{ICAGen}(1^\kappa) \rightarrow (c, st_0)$ computes on input the security parameter in unary a problem instance c and initial state st_0.
- On input the i-th query q_i and state st_{i-1} for $i \in [1; t]$, the efficient algorithm $\mathsf{ICAQuery}_i$ outputs the i-th response p_i and state st_i. For convenience, we will also consider the set of all the $\mathsf{ICAQuery}_i$ as a stateful algorithm $\mathsf{ICAQuery}$ in the natural way.
- The efficient algorithm $\mathsf{ICATriv}$ is given a problem instance c and t-time oracle access to a stateful algorithm $\mathsf{ICAQuery}$. It outputs a candidate solution s. This algorithm implements a trivial attack.
- The verification algorithm ICAVer takes as input a problem instance c, the final state st_t, and a purported solution s and outputs a bit b. If $\mathsf{ICAVer}(c, st_t, s) = 1$ we say that s is a valid solution.

If ICAVer is efficient, we call ICA falsifiable.

Security Game for t-Interactive Complexity Assumptions (tICAs). Let us consider the following security experiment ICA_N^A involving attacker A.

1. The experiment runs $\mathsf{ICAGen}(1^\kappa) \rightarrow (c, st_0)$.
2. A is given c and oracle access to $\mathsf{ICAQuery}$ (run with random coins r_P) for up to t queries.
3. Finally, the attacker outputs a candidate solution s.
4. The experiment returns whatever $\mathsf{ICAVer}(c, st_t, s)$ returns.

Definition 4. *We say that A efficiently breaks tICA* ICA *if A runs in polynomial time and*

$$|\mathsf{Pr}[\mathsf{ICA}_{\mathsf{ICA}}^{A}(1^\kappa) \Rightarrow 1] - \mathsf{Pr}[\mathsf{ICA}_{\mathsf{ICA}}^{\mathsf{ICATriv}}(1^\kappa) \Rightarrow 1]| = \epsilon$$

is non-negligible where the probability is over the random coins consumed by the probabilistic algorithms $\mathsf{ICAGen}, \mathsf{ICATriv}, \mathsf{ICAQuery}$ *and A. We say that tICA* ICA *is secure if no probabilistic polynomial algorithm can break* ICA.

9 First Result: Impossibility of Simple Reductions for General RRR Systems

In this section, we will establish our first result. A standalone proof applied to ElGamal PKE is given in Appendix B. Before we begin, let us describe the restrictions that we make on the considered reductions.

9.1 Simple Reductions

Our first result will consider a restricted form of reduction algorithms that we call simple reductions. We stress that most of the reductions in cryptography seem to be of this type, including all reductions that are used for security proofs of ElGamal PKE and derived schemes. In particular, we will consider reductions B with the following properties.

- B treats the attacker in a black-box way, as in our general result.
- B only calls a single attacker.
- B does not rewind the attacker.

9.2 First Main Result

Now let us formally state our first result.

Theorem 1. *Let* RRR *be a certified RRR system with constant fan-out* k. *Let* p *be a constant that indicates the number of statements that the attacker must invert in the lunchtime-inversion game. Then there is no simple PPT reduction* B *that can reduce the security of the lunchtime-inversion game with* $t+1$ *adaptive queries to the security of any* t*-interactive complexity assumption* ICA*.*

9.3 Proof of Theorem 1

Let us provide an overview of the proof. At first, we specify an ideal (unbounded) attacker A for which the reduction has to work. Next, we present an efficient meta-reduction M that simulates the ideal attacker. Finally, we analyze the difference between the behavior of a reduction B in the two cases. We show that the reduction will not be able to tell the two settings apart. Thus, the reduction also has to work for the efficient M. The combination of B and M will therefore break the underlying tICA. The meta-reduction gains its power from rewinding the reduction.

In a nutshell, the proof relies on the fact that we can always find *a useful rewinding spot* after rewinding. The useful rewinding spot is an attacker query such that i) the reduction provides a (correct) response to that query and ii) the reduction does not query its tICA challenger before delivering the response. Intuitively, the meta-reduction will repeatedly try to hit a useful rewinding spot and send new queries that are derived from the challenge statements to the reduction. The proof exploits that in each of these runs, with non-negligible probability, the reduction will behave as in the first run if presented with a new query since new queries are distributed statistically close to queries of the first run. Let us now be more formal.

9.4 The Ideal Attacker A

1. The attacker A receives random coins r, and R, cert. It aborts in case $\mathsf{RVerify}(R, \mathsf{cert}) \neq 1$. Otherwise, it continues.
2. The parties repeat the following steps for all $i \in [1; t+1]$.
   2.i.1 The attacker A draws $(s_i, w_i) \leftarrow \mathsf{RSubSample}(R)$ and sends s_i to the challenger.
   2.i.2 The challenger responds with witness w'_i. If $i = t$, the challenger also sends challenge statements $s^*_1, \ldots, s^*_p \in S$. The attacker aborts in the case that $\mathsf{ReCheck}(s_i, w_i, w'_i) \neq 1$. Otherwise, it continues.
3. The attacker checks if $s^*_1, \ldots, s^*_p \in S$ and aborts otherwise. Next, the attacker uses its unbounded power to compute a challenge witness $w^*_\ell \in W$ for each $\ell \in [p]$. To this end it could for example compute $\mathsf{RSample}(R; r)$ for all possible random coins and generate the set $W_{s^*_\ell}$ for each $\ell \in [p]$. From this, it can draw an arbitrary witness $w'_\ell \in W_{s^*_\ell}$. Next, it re-randomizes witness w'_ℓ by computing $w^*_\ell \leftarrow \mathsf{ReRand}(s^*_\ell, w'_\ell)$. The final output consists of $w^*_1, \ldots, w^*_p$.

9.5 The Meta-Reduction M can Rewind Reduction B

We will now consider a meta-reduction M that acts as an attacker against the purported reduction B. In particular, we assume that M can store the full execution state state_i of B after B has sent some message and awaits a corresponding response. With these states, M can rewind B to a previous point in time by loading the corresponding execution states. Let us now specify how the meta-reduction simulates the ideal attacker.

9.6 The Simulated Attacker

0. The meta-reduction M receives the tICA instance c and relays it to B along with random coins $r_B \in D_B$.
1. The attacker M receives r, R, cert from B and aborts if $\mathsf{RVerify}(R, \mathsf{cert}) \neq 1$. Otherwise, it continues.
2. The parties repeat the following steps for all $i \in [1; t+1]$.
   - 2.i.1 First the attacker M stores B's execution state state_i. Next, the attacker M calls $(s_i, w_i) \leftarrow \mathsf{RSubSample}(R)$ and sends s_i to the reduction.
   - 2.i.2 The reduction responds with a witness $w'_i \in W$. If the reduction B outputs a query to its tICA challenger, this query is simply relayed by M to its tICA challenger. Likewise, all responses are relayed back to B. The attacker M aborts if $\mathsf{ReCheck}(s_i, w_i, w'_i) \neq 1$. Otherwise, it continues. If $i = t$ the reduction also sends challenge statements $s^*_1, \ldots, s^*_p \in S$ to M.

2' The attacker checks if $s^*_1, \ldots, s^*_p \in S$. On failure it aborts. Otherwise, it halts B and stores the current execution state state^* of B. Then, for each $\ell \in [p]$ the attacker M repeats the following loop:
   - 2'.ℓ The attacker M computes $(s'_\ell, st^*_\ell) \leftarrow \mathsf{RSRStatement}(R, s^*_\ell)$ with $s'_\ell = (s'_{\ell,1}, \ldots, s'_{\ell,k})$. Next, M iterates through all k components of s'_ℓ. To this end, it repeats the following loop for all $j \in [k]$:
     - 2'.$\ell.j$ The attacker M iterates through all possible states state_v. To this end, it repeats the following loop for all state indices $v \in [t+1]$:
       - 2'.$\ell.j.v$ The attacker M rewinds the reduction B back to the point before [2.v.1] by loading state_v. Next it sends $s'_{\ell,j}$ to B. If M receives a response $w'_{\ell,j} \in W$ and no external query to its tICA challenger has been made by the reduction in the time between sending $s'_{\ell,j}$ and receiving $w'_{\ell,j}$, M leaves this loop immediately (break[6]). Otherwise, it checks whether $v = t+1$. In case $v = t+1$ [7], attacker M jumps back to 2'.ℓ and repeats the entire computation calling $\mathsf{RSRStatement}(R, s^*_\ell)$ with fresh randomness.[8]

3. The meta-reduction computes the value $w'^*_\ell \leftarrow \mathsf{RSRWitness}(R, st^*_\ell, w'_\ell)$ for each $\ell \in [p]$ and from this $w^*_\ell \leftarrow \mathsf{ReRand}(R, s^*_\ell, w'^*_\ell)$ for all $\ell \in [p]$. Next it loads B's state state^*, and outputs the challenge witnesses $w^*_1, \ldots, w^*_p \in W$.
4. B responds with a solution to the tICA challenge.
5. Finally, meta-reduction M relays that solution to the tICA challenger.

9.7 Analysis

The proof relies on the following lemma.

Lemma 5. *The following conditions are fulfilled:*

1. M runs in expected polynomial time.
2. M always outputs $w^*_1, \ldots, w^*_p$ if it finishes its computations.
3. All values $w'_{\ell,j}$ received by M as responses to $s'_{\ell,j}$ in the rewinding process of M are correct with non-negligible probability, i.e. $(s'_{\ell,j}, w'_{\ell,j}) \in R$.
4. With probability statistically close to one, the reduction B can never distinguish M from A.
5. M makes at most t queries to the ICA challenger.

With Lemma 5 we are guaranteed that B, after expected polynomial time, outputs a solution to the tICA challenge when given $w^*_1, \ldots, w^*_p$ with non-negligible probability. This concludes the proof of Theorem 1. We note that previous results on meta-reductions like [39] also require the reduction to run in expected polynomial time. By an application of the Markov inequality, we can then truncate the execution of the machine while still guaranteeing an inverse polynomial success probability for infinitely many security parameters. It remains to prove Lemma 5.

[6] The attacker M continues at Step $2'.\ell.j'$ for $j' = j+1$.

[7] This means, that M has not received $w'_{\ell,j} \in W$ as a response to $s'_{\ell,j}$ in any of the states state_v or if the reduction has always made external queries.

[8] This implictly means that if M firts receives an external query as a response to $s'_{\ell,j}$ the loop just continues.

Proof. Consider a reduction B that breaks the underlying security assumption with probability ϵ when communicating with an attacker. The splitting lemma guarantees with non-negligible probability $\epsilon' = \epsilon/2$, that the randomness $r_B \in D_B$ used by the reduction B will make B accept at least a fraction of $\epsilon' = \epsilon/2$ of all the possible query tuple $(s_1, \ldots, s_{t+1}) \in S^{t+1}$. To see this set $U = D_B$ to be the randomness space for B and $V = S^{t+1}$ to be the space of all possible statement tuples and let G be all $(u, v) \in U \times V$ that make B break the tICA. By assumption we have that $|G|/|U \times V| \geq \epsilon$. In terms of the splitting lemma, the randomness r_B is thus a super-good element. In particular, for super-good randomness r_B, B will with probability ϵ' accept a randomly generated tuple $(s_1, \ldots, s_{t+1}) \in S^{t+1}$ that is computed by the ideal attacker A as $s_i = [\mathsf{RSubSample}(R)]_1$ for each $i \in [t+1]$. Each such tuple will make the reduction output the challenge statements and if the reduction B is provided corresponding witnesses $w_1^*, \ldots, w_p^*$ to the challenges $s_1^*, \ldots, s_p^*$ next, B will break the tICA. Observe that if B accepts $s_1, \ldots, s_{i+1}$ for any $i \in [t]$ it will trivially also accept $s_1, \ldots, s_i$. In the following, we will thus concentrate on a single, fixed randomness r_B that the reduction uses and assume it to be super-good. This accounts for at most an additional non-negligible decrease of the overall success probability by a factor of ϵ'.

Let us now focus on the difference in the behaviour of reduction B when B communicates with M instead of the ideal attacker A. A closer inspection reveals that the only case that the behavior may differ occurs if the reduction does abort in Step 2' but does not abort otherwise or if in Step 2' (or one of the substeps) the reduction delivers to an attacker query s a response w that is incorrect such that we have $(s, w) \notin R$. Otherwise, the simulation is perfect. Let us thus now analyze this event in more detail. We begin with a helpful observation. We observe that for each tuple $(s_1, \ldots, s_{t+1}) \in S^{t+1}$ that the reduction accepts there is always one s_i with $i \in [t+1]$ such that i) the reduction does not make a query to its tICA challenger before delivering the response w_i and ii) the reduction has provided a correct response in the first run. The first condition is guaranteed simply because the number $t+1$ of queries allowed in the lunch-time inversion game is larger than the number of queries t allowed in the communication with the tICA challenger. The second condition is true since the meta-reduction (in the first run) and the ideal attacker, both have access to the witnesses of the statements they send. Thus they can always use $\mathsf{ReCheck}$ to verify the responses and any successful reduction is bound to deliver correct witnesses w_i for every $i \in [t]$ with high probability.

First, observe that if the meta-reduction finishes, it will by construction always output witnesses $w_1^*, \ldots, w_p^*$.

Let us next show that the meta-reduction runs in expected polynomial time. First, observe that all operations performed in step 2'.$\ell.j.v$ are efficient. To argue that the entire meta-reduction is efficient, and since the number of each loop iteration is polynomially bounded ($\ell \in [p], j \in [k], v \in [t+1]$) we must now specifically show that the jumps in Step 2'.$\ell.j.v$ back to 2'.ℓ will not make the overall runtime super-polynomial. We thus have to analyse how likely such jumps are. To this end, compute the probability $\Pr[E_{\ell,j,v}]$ of the event $E_{\ell,j,v}$ that after sending $s'_{\ell,j}$ in state_v, M has received back witness $w'_{\ell,j} \in W$ while B has not made an external query. Recall that B will with probability ϵ' accept a random tuple of statements before the attacker outputs a forgery. This in particular means that to *any single* statement s_v of such a vector, the reduction will respond with a witness w_v with at least probability ϵ'. Recall that $s'_{\ell,j}$ is distributed as s_v (except for statistically small probability δ). So, from the viewpoint of the reduction, the overall tuple of statements received so far $(s_1, \ldots, s_{v-1}, s'_{l,j})$ is distributed like a tuple of an ideal attacker except with probability δ. Thus, the reduction will respond to $s'_{\ell,j}$ with witness $w'_{\ell,j}$ with at least probability $\epsilon' - \delta$ in state state_v. Now, since we must always have that at least one of the $t+1$ statements that are delivered to B in an accepting tuple will not invoke external queries to compute the response, we have that $\Pr[E_{\ell,j,v}] \geq (\epsilon' - \delta)/(t+1)$ for *at least one* of the possible states state_v with $v \in [t+1]$. Next, compute the probability $\Pr[E_{\ell,j}]$ of the event $E_{\ell,j}$ that for *any* of the states state_v with $v \in [t+1]$, reduction R has responded to query $s'_{\ell,j}$ with $w'_{\ell,j} \in W$ without making an external query. For this event we have that $\Pr[E_{\ell,j}] = \Pr[E_{\ell,j,1} \vee \ldots \vee E_{\ell,j,t+1}] \geq (\epsilon' - \delta)/(t+1)$. This means that after an expected number of $x = O(1/\Pr[E_{\ell,j}])$ iterations per $j \in [k]$ the meta-reduction will not jump back to $2'.\ell$. Let us finally compute the probability $\Pr[E_\ell]$ of the event E_ℓ that the reduction responds with $w'_{\ell,j}$ to all the $s'_{\ell,j}$ with $j \in [k]$ (for fixed ℓ) without making any external queries. We have that $\Pr[E_\ell] = \Pr[E_{\ell,1} \wedge \ldots \wedge E_{\ell,k}]$. Now, since any derived value $s'_{\ell,j}$ is distributed statistically close to any of the s_v (and thus statistically close to a value that is independent of any other $s'_{\ell',j'}$) we get that $\Pr[E_{\ell,i+1}|E_{\ell,i}] \geq (\epsilon' - \delta)/(t+1)$ for all $i = 1, \ldots, k-1$. Now, we have probability $\Pr[E_\ell] = \Pr[E_{\ell,1}] \cdot \prod_{i=1}^{k-1} \Pr[E_{\ell,i+1}|E_{\ell,i}] \geq ((\epsilon' - \delta)/(t+1))^k$. For any sufficiently large security parameter, we have that $\epsilon' \geq \delta$ since δ is statistically small and ϵ' is non-negligible. Thus $\Pr[E_\ell] \geq (\epsilon'/(2t+2))^k$ and since k is constant, $\Pr[E_\ell]$ is non-negligible. Thus after at most a polynomial number of iterations $O(1/\Pr[E_\ell])$ the meta-reduction will not jump back and finish the loop for a single $\ell \in [p]$. Repeating this process for all ℓ and relying on the fact that all operations in step $2'.\ell.j.v$ are indeed efficient, we now obtain that after an expected polynomial number of iterations the meta-reduction obtains all p k-tuples $w'_\ell = (w'_{\ell,1}, \ldots, w'_{\ell,k})$ for $\ell \in [p]$. This shows that M is indeed efficient.

Now let us consider the probability that all the witnesses received in the rewinding process are indeed correct in the sense that $(s'_{\ell,j}, w'_{\ell,j}) \in R$. This is necessary because since R is not a strong RRR, the meta-reduction cannot test whether the outputs are indeed correct. We have that the reduction responds with a correct witness $w'_{\ell,j}$ per statement $s'_{\ell,j}$ with probability at least $\epsilon' - \delta$. This is because i) in the first run, each statement s_i will make the reduction respond with w_i with probability at least ϵ' and ii) the $s'_{\ell,j}$ are distributed like the s_i except with probability δ. So after receiving a total of $p \cdot k$ witnesses $s'_{\ell,j}$ the probability that all of them are correct is at least $(\epsilon' - \delta)^{p \cdot k}$ which is non-negligible in case $p \cdot k$ is constant. (Jumping ahead, in our second main proof, we can, by the properties of the Strong RRR, immediately test whether the responses of the reduction are indeed correct. This reduces the bound to $(\epsilon' - \delta)^k$. We are thus able to have a polynomial p at this point. However, we need to still require k to be constant since the algorithm $\mathsf{RSRStatement}$ always outputs k statements at a time. If only a single one of them is incorrect we have to jump back.)

Let us finally analyze the distribution of values produced by the meta-reduction and compare them to the ideal attacker. First observe that according to the view of the reduction B, B will always see at most a single transformed statement $s'_{\ell,j}$, each time as part of a tuple of statements of the form $s = (s_1, \ldots, s_i, s'_{\ell,j})$. Since only the last statement is not distributed like the output of the ideal attacker, the distribution of s is identical to the ideal attacker except for statistically small probability δ. In other words, B will respond as for an ideal attacker except with probability δ. Now if the values $w'_{\ell,j}$ received by M are all correct, M can finally compute $w'^*_\ell \leftarrow \mathsf{RSRWitness}(R, st, w'_\ell)$ and ultimately, after re-randomization, the meta-reduction outputs all the re-randomized $w_1^*, \ldots, w_p^*$. Due to property P-4, each of these values is distributed like the ideal attacker. Thus, the reduction will not tell the tuple $(w_1^*, \ldots, w_p^*)$ apart from the witnesses output by the ideal attacker outputs.

At last, observe that, after the rewindings, B's view will be such that overall it has received a tuple of statements $(s_1, \ldots, s_{t+1})$. Since the $s_1, \ldots, s_{t+1}$ are exactly distributed as the ideal attacker, the reduction will never be able to distinguish the meta-reduction from the ideal attacker except for probability ζ. Moreover, the meta-reduction has never made more than t queries to the challenger of the tICA as it is merely relaying queries from the reduction to the tICA challenger in Step 2. By assumption, this will only account for up to t external queries. In particular, no additional query will be relayed to the tICA challenger after a rewinding in Step 2'.

To sum up, we have that i) the runtime of M is polynomially bounded, ii) that it will always output a witness tuple $(w_1^*, \ldots, w_p^*)$, iii) that the witnesses received by M from B in the rewinding process are correct with non-negligible probability, and iv) that the reduction B can never distinguish M from A with probability statistically close to one. Finally, we have v) that M makes only at most t queries to the ICA challenger altogether. Thus B outputs after expected polynomial time a solution to the tICA challenge when given w^* with non-negligible probability.

□

9.8 No Need for the Forking Lemma

It might be tempting to consider an analysis of the error probability that relies on the Forking Lemma [41]. However, it turns out that this is not necessary and the Splitting Lemma suffices. Arguments based on the Forking Lemma often need to guarantee that the queries sent at the rewinding spot before and after the actual rewinding are distinct. This accounts for an additional term in the analysis that depends on the size of the set of all possible queries. In particular, a small set of queries is prohibitive for a security proof. In our analysis, however, this is not necessary. This is because in our analysis we could theoretically have that the two queries (statements) that are sent at the rewinding spot, one before and one after the actual rewinding, *may be equal.* This is essentially because the second query is constructed by deriving it from a challenge statement. So a single answer w to statement s could theoretically help to not only compute a witness for s but also for other statements s^* that, after blinding (for self-reducibility), happen to result in s.

10 Applications of First Main Theorem

We now present several interesting applications of our first theorem. To apply it to semi-homomorphic PKE, we can simply concretely instantiate the RRR system in the theorem by RRR where $k = 1$ (Figure 3). We immediately end up in the OW-CCA1 security game for PKE. At the same time, this establishes a hierarchy of security games.

Corollary 1. *There is no simple PPT reduction that can reduce the OW-CCA1 security of a certified semi-homomorphic PKE with* $t + 1$*-decryption queries to the security of any* t*-interactive complexity assumption.*

Corollary 2. *There is no simple PPT reduction that can reduce the OW-CCA1 security of any certified semi-homomorphic PKE with* $t + 1$ *decryption queries to the OW-CCA1 security of the same semi-homomorphic PKE with* t *decryption queries.*

Exploiting that CHOWBs can easily be cast as RRRs (Sectionx 6.3), we can also apply our first result to CHOWBs (Figure 5).

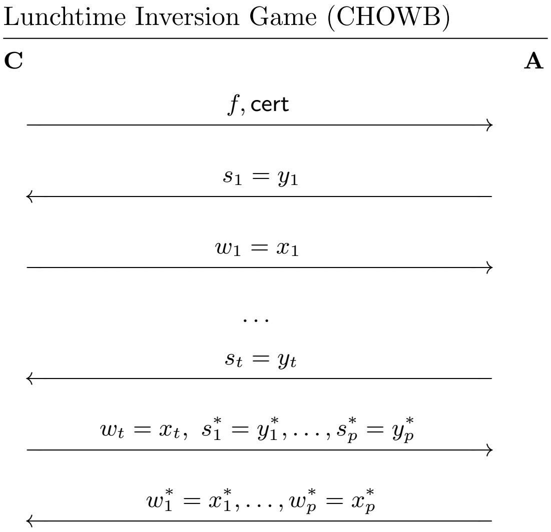
Lunchtime Inversion Game (CHOWB)
C
A
f, cert
$s_1 = y_1$
$w_1 = x_1$
...
$s_t = y_t$
$w_t = x_t,\ s_1^* = y_1^*, \ldots, s_p^* = y_p^*$
$w_1^* = x_1^*, \ldots, w_p^* = x_p^*$

Fig. 5. Security definition when instantiating the second main result with strong RRRs based on CHOWB $f \in F$. Statements s correspond to output values y, while witnesses w correspond to inputs x of f. The resulting security game is equivalent to the security game for CHOWBs.

Corollary 3. *Let* F *denote an efficiently sampleable family of certified homomorphic one-way bijections. Then we have that there is no simple PPT reduction that can reduce the security of* F *with* $t + 1$*-inversion queries to the security of any* t*-interactive complexity assumption.*

Corollary 4. *Let* F *denote an efficiently sampleable family of certified homomorphic one-way bijections. Then we have that there is no simple PPT reduction that can reduce the security of* F *with* $t + 1$*-inversion queries to the security of* F *with any* t*-inversion queries.*

Even more concretely, we finally obtain a result on ElGamal PKE (Figure 6). This in particular shows that ElGamal cannot be shown IND-CCA1 secure via simple reductions. Appendix B gives a standalone proof of this result.

Corollary 5. *There is no simple PPT reduction that can reduce the OW-CCA1 security of ElGamal PKE with $t+1$ decryption queries to the security of any t-interactive complexity assumption.*

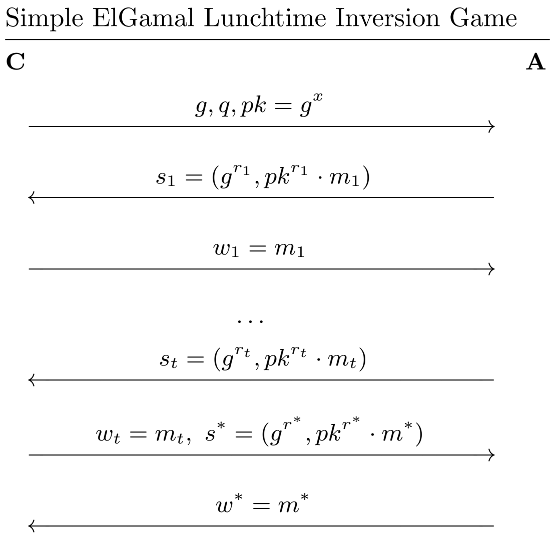
Simple ElGamal Lunchtime Inversion Game
C
A
$g, q, pk = g^x$
$s_1 = (g^{r_1}, pk^{r_1} \cdot m_1)$
$w_1 = m_1$
...
$s_t = (g^{r_t}, pk^{r_t} \cdot m_t)$
$w_t = m_t,\ s^* = (g^{r^*}, pk^{r^*} \cdot m^*)$
$w^* = m^*$

Fig. 6. Security definition considered in our first result. Statements are ElGamal ciphertexts, witnesses are the corresponding plaintexts. The public parameters define a cyclic group of prime order q with generator g and a public key pk. This guarantees that for each ciphertext, there is a unique plaintext.

As a last example, we may apply our result to the discrete exponentiation function which in prime order groups yields a certified homomorphic one-way bijection, i.e. a concrete instance of R_f.

Corollary 6. *There is no simple PPT reduction that can reduce the one-more DLOG assumption with access to t' DLOG-queries to the security of any t-interactive complexity assumption where $t' = t + 1$.*

The one-more DLOG assumption and one-more variants of other CHOWBs have for example been analyzed in [6]. This result is a quantitative improvement over [39] who showed an asymptotic bound of $t' = \omega(\kappa + t + 1)$. However, in this result, we consider only simple reductions.

11 Second Main Result

In this section, we describe our second main result. We believe that it is instructive to first explain how our first result breaks down in the presence of rewinding reductions.

11.1 On the Necessity of Non-Rewinding Reductions

For our first result, it is important that the reduction cannot rewind the attacker. Otherwise, it could (in Step 2.i.2) first try to send some incorrect w' to M such that $(s, w') \notin R$ when answering an inversion query for s. Only if the attacker aborts, the reduction would rewind M and retry to respond to the query s with a correct w. However, since M can *in a rewound run* not notice if w' is correct while the ideal attacker can always notice this, the reduction can easily tell A and M apart. This is a fundamental problem when dealing with RRRs.

11.2 Tackling Rewinding Reductions

In our second main result, we will thus extend and refine the above idea to deal with general reductions. In particular, we need to take care of rewinding reductions. At the same time we take care of reductions that execute several, u, attackers concurrently. To this end, we make use of novel and known techniques. However, what is crucial is that because of the generic problem that we described we now have to rely on strong RRR systems when dealing with rewinding reductions. This has important consequences. First, we will end up showing an impossibility result for the weaker notion of Paired OW-CCA1 security (Section 4.6) when applying our result to strong RRRs based on semi-homomorphic PKE. This makes our result seemingly stronger than the last one since this notion is weaker than OW-CCA1 security and thus IND-CCA1 security. However, crucially, the result now holds only if the reduction makes at most half the number of queries $\lfloor t/2 \rfloor$ to its $\lfloor t/2 \rfloor$ICA challenger where t is the number of queries by the attacker. This also makes our result somewhat weaker. Finally, we will consider reductions that make at most arbitrary u instances overall.

Theorem 2. *Let* RRR *be a certified strong RRR system with constant fan-out* k*. Let* p *be a polynomial. There is no PPT reduction* B *that can reduce the security of the lunchtime inversion game with* $t' \geq (u + t)$ *adaptive queries to the security of any tICA while creating at most* u *instances of the attacker overall.*

Intutition: Lazy Simulation. Let us provide some intuition for the simulation strategy of our final meta-reduction. Assume the reduction B creates u instances of the attacker A. Intuitively, we employ a careful generalization of the previous proof strategy that is applied to each simulated attacker, one after the other. The meta-reduction M simulates all of these runs. Since it has to deal with rewinding reductions, the proof makes sure that the ideal attacker and the meta-reduction behave deterministically from the point of view of the reduction. Thus rewindings do not help the reduction to obtain additional information. Corresponding to the lunchtime inversion game (Figure 1, Section 7), in the first phase of each of the simulated attackers, the meta-reduction only queries the reduction on $t + 1$ random statements. At the end of each of these phases, the meta-reduction has received back the corresponding witnesses to its random queries and also the set of challenge statements $s_1^*, \ldots, s_p^*$. Let A' be the instance of the simulated attacker that receives the challenge statements *first*. This starts the second phase of the simulation of A' which is similar to the proof of the first main theorem. First, the meta-reduction stores the state of the reduction at this point to later come back to it. Next, M repeatedly rewinds B back to hit a useful rewinding spot with a derived statement. However, in this proof a useful rewinding spot fulfills three conditions. It is an index i, such that in the time span between query s_i from A' and corresponding response w_i sent to A', the reduction has i) *not* made any external query to the interactive complexity assumption, ii) has responded with a correct response (this can now be tested), and iii) not required any of the $u - 1$ other instances of the (simulated) attacker to output a forgery. As in the first proof, i) ensures that the outside communication between B and its challenger is not disturbed. The third condition is crucial to avoid that simulation runtimes can grow exponentially when the communication with the attacker is nested by the reduction *after* rewinding [39]. Since there are only t queries from the reduction to the complexity assumption overall and since we only have at most u instances (and at most $u - 1$ that are different from A') there will always be at least one useful rewinding spot if $t' \geq u + t$ by the pigeon-hole principle. Now, the reduction, as in the first proof, proceeds according to a lazy strategy that tries to extract *all* p challenge witnesses by exploiting that M can repeatedly send new query s_i to B and that, with high probability, B behaves as in the first run. As a first step, it randomizes the first challenge statement s_1^* (exploiting random self-reducibility) and sends the result as a new query s_i' at index i to B. Since the distribution of the two queries s_i, s_i' are statistically close, the reduction will with high probability behave as after the first query and output a witness without calling the challenger of the complexity challenge. On failure, M simply generates another randomization of s_1^* and tries again. Observe that since we now rely on strong RRRs, the meta-reduction can, like the ideal attacker, *always* check if the responses from the reduction are correct. On success, M can exploit the response to s_i' and the properties of the RRR (random self-reducibility and re-randomizability) to obtain a solution w_1^* to the first challenge statement. After that, M repeats the process with all of its other challenge statements, one after the other. Since the meta-reduction can now *immediately* check if a witness w_i^* is indeed correct (exploiting the indirect membership test of strong RRRs at the end of Step 2'.ℓ.j in the meta-reduction of Appendix B.5), the meta-reduction can proceed in a step-by-step manner and only continue if the previous witness was successfully found. In this way, the meta-reduction can slowly but surely accumulate all p witnesses. Next, the meta-reduction returns to the original execution state and, after re-randomization, outputs the solutions to the challenge queries. By the properties of the RRR, the reduction will accept these values with high probability. The simulation continues until the next simulated attacker receives its challenge statements and the process of extracting solutions to the challenge statements is repeated. This increases the runtime of the meta-reduction at most by an additional factor of u. Finally, if M has computed responses to all challenge statements, B will output the solution to the tICA.

Let us now begin with the formal proof. The proof closely follows the proof for the first main theorem.

11.3 The Ideal Attacker

Assume the attacker A_{ideal} is given access to a random oracle $h(\cdot)$. In the lunchtime inversion security game with challenger C, the ideal attacker A_{ideal} works as follows. To model that the reduction can rewind the attacker at will, we model A_{ideal} as a deterministic next message function:

1. *Verifying the initialization values.* If C feeds A_{ideal} with relation R, a proof cert, and random coins r, A_{ideal} runs $\mathsf{RVerify}(R, \mathsf{cert}) = b$. If $b = 0$, the attacker aborts. Otherwise, A_{ideal} deterministically computes the first statement. To this end, it computes randomness $r_1 = h(r, R, 1)$ and $(s_1, w_1) = \mathsf{RSubSample}(R; r_1)$ and outputs s_1 to C.
2. *Verifying the initialization values and the witnesses.* If C sends to A_{ideal} a message consisting of values $r', R', \mathsf{cert}', w'_1, \ldots, w'_{i'}$ for $i' \in [t']$, the attacker first verifies all values in that view. To this end, A_{ideal} at first *verifies the initialization values* as before. It thus computes $\mathsf{RVerify}(R', \mathsf{cert}') = b$, and if $b = 0$, A_{ideal} aborts. Otherwise A_{ideal} *verifies the witnesses* $w'_1, \ldots, w'_{i'}$. To this end A_{ideal} will compute $r_i = h(r, R, i)$ for all $i \in [i']$ and then $(s_i, w_i) = \mathsf{RSubSample}(R; r_i)$. Next, it will check for all $i \in [i']$ that $\mathsf{ReCheck}(R, s_i, w_i, w'_i) = 1$. On failure A aborts. After this verification, A_{ideal} computes a new statement. It computes $(s_{i'+1}, w_{i'+1}) \leftarrow \mathsf{RSubSample}(R; r_{i'+1})$ with $r_{i'+1} = h(r, R, i' + 1)$ and outputs $s_{i'+1}$ to C.[9]
3. *Verifying the initialization values, the witnesses, and computing challenge witnesses.* If the message $r', R', \mathsf{cert}', w'_1, \ldots, w'_{t'}, s^*_1, \ldots, s^*_p$ is sent by C to A_{ideal}, A_{ideal} first verifies the initialization values r', R', cert' as before. On failure, it aborts the simulation of A. Next, A_{ideal} verifies the witnesses w'_i for $i \in [t']$ as before. On failure A_{ideal} aborts. Finally, it tests whether we have $s^*_1, \ldots, s^*_p \in S$. If this does not hold, the ideal attacker aborts. Next, A_{ideal} uses its unbounded power to compute $w^*_1, \ldots, w^*_{t'}$ such that $(s^*_i, w^*_i) \in R$ for all $i \in [t']$. The ideal attacker can do this by repeatedly running $\mathsf{RSample}(R; r'')$ for all possible random values $r'' \in D$ and finding among the outputs, for all i, the lexicographically smallest pair $(s^*_i, w'^*_i) \in R$. Next, the ideal attacker can compute $r^* = h(r, R, t' + 1)$ (we also use $r_{t'+1} = r^*$) and output $w^*_i \leftarrow \mathsf{ReRand}(R, s^*_i, w'^*_i; r^*)$.

This ends the description of A_{ideal}. Observe that A_{ideal} wins with probability 1. Also, A_{ideal} is entirely deterministic.

Invariant of Each Instance of the Ideal Attacker. Observe that for each instance of the ideal attacker we always have that for a single query $(s_i, w_i) \leftarrow \mathsf{RSubSample}(R; r_i)$ it holds that

1. the reduction outputs to A a correct witness w'_i as a response to s_i. By property P-5 it holds that $\mathsf{ReCheck}(R, s_i, w_i, w'_i) = 1$ if and only if $(s_i, w_i) \in R$.
2. in the time between receiving s_i and responding w'_i, B has not made a query to its tICA challenger.
3. in the time between receiving s_i and responding w_i, B has not sent a challenge statement to any of the other $u - 1$ instances.

The last two properties simply hold by the pidgeonhole principle because there are overall $t + u$ queries made by the attacker but only at most $u - 1$ other instances of A and at most t queries to the tICA. So for at least one query the latter two properties are fulfilled. The first property is always fulfilled since the ideal attacker will simply abort if $\mathsf{ReCheck}(R, s_i, w_i, w'_i) \neq 1$.

[9] Observe that this process guarantees that the ideal attacker will only output a new statement if it has received correct witnesses for all the previous statements. In this way, we carefully make the derivation of $s_{i'+1}$ depend on the correctness of the witnesses provided by the reduction but not on the witnesses themselves.

11.4 The Meta-Reduction M Rewinds B and Behaves Deterministically

We will now consider a meta-reduction M that executes the purported reduction B. We use $A[z]$ to denote the zth instance of A that is simulated by B with $z \in u$. We will keep most of the variables that have been used in the first proof. For any variable x that belongs to the communication between B and $A[z]$ we write $x[z]$. As before, we assume that M can store the full execution state $\mathsf{state}_i[z]$ of B immediately after $A[z]$ has sent message $s_i[z]$ and awaits a corresponding response. With these states, M can rewind B to a previous point in time by loading the corresponding execution states. Correspondingly $\mathsf{state}^*[z]$ is the state of B when B has just output $s_1^*[z], \ldots, s_p^*[z]$ to $A[z]$.

We will also use that since the meta-reduction controls the reduction it can easily simulate the deterministic behavior of the ideal attacker via appropriate bookkeeping. For the sake of simplicity, we refrain from explaining this in detail. We can imagine a table that essentially simulates what the random oracle does for the ideal attacker. The table could for example contain all the randomness $r_i[z]$ and $r^*[z]$ that M uses to compute $s_i[z]$. Another way to view this, is that M precomputes all the $r_i[z]$ for $i \in [t+1]$ and $z \in [u]$ as random values in D and uses them consistently.

Let us now specify how the meta-reduction proceeds. Intuitively it behaves honestly as long as the reduction does not require any of the $A[z]$ to deliver a witness tuple $w_1^*[z], \ldots, w_p^*[z]$ to challenge $s_1^*[z], \ldots, s_p^*[z]$. We now have to address that the reduction may, at any time, create new instances of the attacker. Moreover, the messages may arbitrarily be interleaved. In the following, let z' be a variable counting the current number of instances created and initialize it to $z' = 0$. By assumption, we always have $z' \leq u$.

11.5 The Simulation of u Attackers

0. The attacker M receives the tICA instance c and relays it to B along with random coins $r_B \in D_B$.
1. If at any point, M receives r, R, cert from B it verifies the initialization values by computing $\mathsf{RVerify}(R, \mathsf{cert})$. It aborts if $\mathsf{RVerify}(R, \mathsf{cert}) \neq 1$. Otherwise, it increments z' setting $z' \leftarrow z' + 1$ and starts to simulate a new, z'th instance $A[z']$ of the attacker. To this end, M first sets $(r[z'], R[z'], \mathsf{cert}[z']) = (r, R, \mathsf{cert})$. Next, M stores B's execution state $\mathsf{state}_1[z']$, calls $(s[z']_1, w[z']_1) \leftarrow \mathsf{RSubSample}(R[z'])$, and outputs $s_1[z']$.
2. If B sends to $A[z]$ ($z \in [z']$) a message $r'[z], R'[z], \mathsf{cert}'[z], w_1'[z], \ldots, w_{i'}'[z]$ for $i' \in [t']$, $A[z]$ first *verifies the initialization values* as the ideal attacker. To this end, instance $A[z]$ computes $\mathsf{RVerify}(R'[z], \mathsf{cert}'[z]) = b$, and if $b = 0$, it aborts. Otherwise $A[z]$ *verifies the witnesses* $w_1'[z], \ldots, w_{i'}'[z]$. To this end, it computes $(s_i[z], w_i[z]) = \mathsf{RSubSample}(R[z]; r_i[z])$ for each i. Next, it will check for all $i \in [i']$ that $\mathsf{ReCheck}(R[z], s_i[z], w_i[z], w_i'[z]) = 1$. On failure $A[z]$ aborts. After this verification, $A[z]$ computes a new statement. This is done by computing $(s_{i'+1}[z], w_{i'+1}[z]) \leftarrow \mathsf{RSubSample}(R; r_{i'+1}[z])$. Now M stores B's execution state as $\mathsf{state}_{i'+1}[z]$ and outputs $s_{i'+1}[z]$ to B.
3. *Verifying the initialization values, the witnesses, and computing challenge witnesses.* If a message consisting of $r'[z], R'[z], \mathsf{cert}'[z], w_1'[z], \ldots, w_{t'}'[z], s_1^*[z], \ldots, s_p^*[z]$ is sent by B to $A[z]$ for ($z \in [z']$), $A[z]$ first verifies the initialization values $r'[z], R'[z], \mathsf{cert}'[z]$ as before. On failure, B aborts the simulation of $A[z]$. Next, $A[z]$ verifies the witnesses $w_i'[z]$ for $i \in [t']$ as before. On failure, $A[z]$ aborts. Finally, it tests whether we have $s_1^*[z], \ldots, s_p^*[z] \in S$. If this does not hold, $A[z]$ aborts. Now, $A[z]$ looks up if the values $w_1^*[z], \ldots, w_p^*[z]$ have in the past already been computed. If so, it outputs them to the reduction. Otherwise, it calls algorithm $(w_1^*[z], \ldots, w_p^*[z]) \leftarrow \mathsf{Extract}(z)$ and outputs $w_1^*[z], \ldots, w_p^*[z]$.
4. If at any point, M receives a query of the reduction that is directed at its tICA challenger, M just relays it to its own tICA challenger and sends the response of its own tICA challenger back to B.

This finishes the description of the meta-reduction. Observe that besides $\mathsf{Extract}$ all algorithms are efficient. It remains to specify algorithm $\mathsf{Extract}$.

11.6 The Extraction Algorithm

Essentially while focusing on a single z, $\mathsf{Extract}$ closely follows the rewinding strategy of the meta-reduction of the first proof. The only change is that useful rewinding spots are defined differently and that the proof now supports polynomial p and constant k — in contrast to demanding constant $k{\cdot}p$ only. A useful rewinding spot is now characterized by the behavior of reduction B when receiving $s'_{\ell,j}[z]$ in state $\mathsf{state}_v[z]$ s.t. $(s'_{\ell,1}[z], \dots, s'_{\ell,k}[z], st^*_\ell[z]) = \mathsf{RSRStatement}(R[z], s^*[z]; r_v[z])$. We call $s'_{\ell,j}[z]$ when it is sent to the reduction in state $\mathsf{state}_v[z]$ a useful rewinding spot if

1. the reduction outputs to $A[z]$ a witness $w'_{\ell,j}[z]$ with $\mathsf{RSRTest}(R[z], w'_{\ell,j}[z], j, st^*_\ell[z]) = 1$ as a response.
2. in the time between receiving $s'_{\ell,j}[z]$ and responding $w'_{\ell,j}[z]$ has not made a query to its tICA challenger, and
3. in the time between receiving $s'_{\ell,j}[z]$ and responding $w'_{\ell,j}[z]$ has not sent a challenge statement to any of the other $z' - 1$ instances.

In our arguments, we will intuitively use the fact that these conditions are fulfilled with non-neglible probability. The second and third requirements hold with non-negligible probability because i) M attempts to send $s'_{\ell,j}[z]$ in any possible state and ii) $s'_{\ell,j}[z]$ is distributed like the s_i of the ideal attacker except for a statistically small error. We thus have that any query $s'_{\ell,j}[z]$ of $A[z]$ has the same probability of making B fulfill the last two properties, as one of the queries s_i of the ideal attacker. The first property holds because property P-9 holds for strong RRRs. It essentially states that the test performed via $\mathsf{RSRTest}(R[z], w'_{\ell,j}[z], j, st^*_\ell[z]) = 1$ is statistically close to the test $\mathsf{ReCheck}(R, s_i, w_i, w'_i) = 1$ performed by the ideal attacker. For any $z \in [z']$, and $z' \leq u$ we must thus always be able to find a useful rewinding spot with non-negligible probability. Let us now describe algorithm $\mathsf{Extract}$.

$\mathsf{Extract}(z)$:

```
1’ $M$ stores the execution state $\mathsf{state}^*[z]$ of $B$.
2’ The meta-reduction repeats the following loop for each $\ell \in [p]$:
    2’.$\ell$ The meta-reduction computes $(s'_\ell[z], st^*_\ell[z]) \leftarrow \mathsf{RSRStatement}(R[z], s^*_\ell[z])$ with $s'_\ell[z] = (s'_{\ell,1}[z], \dots, s'_{\ell,k}[z])$. Next, the meta-reduction iterates through all $k$ components of $s'_\ell[z]$. To this end, it repeats the following loop for all $j \in [k]$:
        2’.$\ell.j$ The meta-reduction iterates through all possible states $\mathsf{state}_v[z]$. To this end, it repeats the following loop for all state indices $v \in [t+1]$:
            2’.$\ell.j.v$ The meta-reduction rewinds the reduction $B$ back to the point before [2.$v$.1] by loading $\mathsf{state}_v[z]$. Next it sends $s'_{\ell,j}[z]$ to $B$. If $M$ receives a response $w'_{\ell,j}[z] \in W$, $M$ checks i) if no external query to its $t$ICA challenger has been made by the reduction in the time between sending $s'_{\ell,j}[z]$ and receiving $w'_{\ell,j}[z]$, ii) whether no challenge $s^*_1[z''], \dots, s^*_p[z'']$ has been sent to any other instance $A[z'']$ ($z'' \neq z$) by the reduction in the time between sending $s'_{\ell,j}[z]$ and receiving $w'_{\ell,j}[z]$, and iii) if $w'_{\ell,j}[z]$ looks like a correct witness for statement $s'_{\ell,j}[z]$ with respect to $R[z]$ by testing $\mathsf{RSRTest}(R, w'_{\ell,j}[z], j, st^*_\ell[z]) = 1$. If all of these conditions are fulfilled, $M$ leaves the loop immediately (break[10]). Otherwise, it checks whether $v = t+1$. On success[11], the meta-reduction jumps back to 2’.$\ell$ and repeats the entire computation calling $\mathsf{RSRStatement}(R[z], s^*_\ell[z])$ with fresh randomness.
3’. The meta-reduction computes the value $w'^*_\ell[z] \leftarrow \mathsf{RSRWitness}(R[z], st^*_\ell[z], w'_\ell[z])$ for each $\ell \in [p]$ and from this $w^*_\ell[z] \leftarrow \mathsf{ReRand}(R[z], s^*_\ell[z], w'^*_\ell[z]; r^*[z])$ for all $\ell \in [p]$. Next it loads $B$’s state $\mathsf{state}^*[z]$, and returns the challenge witnesses $w^*_1[z], \dots, w^*_p[z] \in W$.
```

[10] The meta-reduction continues at Step $2'.\ell.j'$ for $j' = j + 1$.

[11] This means, that the meta-reduction has not received $w'_{\ell,j}[z] \in W$ as a response to $s'_{\ell,j}[z]$ in any of the states $\mathsf{state}_v[z]$ or that the reduction has always made external queries or that it has always queried challenge statements to other instances.

11.7 Analysis

The proof is very close to the previous proof of our first main result.

Consider a reduction B that breaks the underlying security assumption with probability ϵ when communicating with an attacker. First observe that any reduction B that will never see a tuple $(w_1^*[z], \ldots, w_{t+1}^*[z])$ for any of the $z \in [z']$ can easily be used to break the tICA since it will essentially only receive randomly generated statements. So in the following we assume that for at least one $z'' \in [z']$ the reduction will accept $s_1[z''], \ldots, s_{t+1}[z'']$. For any other $z \in [z'] - z''$ we assume that the reduction accepts arbitrary $e[z] \in [t+1]$ queries. For convenience, we also set $e[z''] = t+1$. Now, using this notation we have that the reduction will in general accept statements $s_1[z], \ldots, s_{e[z]}[z]$ for each $A[z]$. Now let us consider the set S' of all possible tuples that the reduction could potentially accept before breaking the tICA where

$$S' = \left\{ \left(\begin{array}{c} (s_1[1], \ldots, s_{e[1]}[1]), \\ \ldots, \\ (s_1[z'], \ldots, s_{e[z']}[z']) \end{array} \right) \middle| \begin{array}{l} \forall z \in [z'] : s_{i,z} \in S, e[z] \in [t+1] \\ \wedge \max_{\hat{z} \in [z']}\{e[\hat{z}]\} = t+1 \end{array} \right\}.$$

For convenience, we call S' the set of all possible *configurations*. Assume B has success probability ϵ when communicating with the ideal attacker. The splitting lemma guarantees with non-negligible probability $\epsilon' = \epsilon/2$, that the randomness $r_B \in D_B$ used by the reduction B will make B accept at least a fraction of $\epsilon' = \epsilon/2$ of all the possible query tuples $s' \in S'$. To see this, set $U = D_B$ to be the randomness space for B and $V = S'$ to be the space of all possible statement tuples and let G be all $(u, v) \in U \times V$ that make B break the tICA. By assumption we have that $|G|/|U \times V| \geq \epsilon$. In terms of the splitting lemma, the randomness r_B is thus a super-good element. In particular, for super-good randomness r_B, B will with probability ϵ' accept a random $s' \in S'$. In the following, we will concentrate on a single, fixed randomness r_B that the reduction uses and assume it to be super-good. This accounts for at most an additional non-negligible decrease of the overall success probability by a factor of ϵ'. For convenience, let us prepare some additional terminology.

We say that B *accepts* configuration

$$s' = \big((s_1[1], \ldots, s_{e[1]}[1]), \ldots, (s_1[z'], \ldots, s_{e[z']}[z'])\big) \in S'$$

if for each $z \in [z']$, B accepts $(s_1[z], \ldots, s_{e[z]}[z])$ when communicating with $A[z]$. Observe that if B accepts

$$s' = \big((s_1[1], \ldots, s_{e[1]}[1]), \ldots, (s_1[z'], \ldots, s_{e[z']}[z'])\big) \in S'$$

it trivially also accepts

$$s'' = \big((s_1[1], \ldots, s_{e'[1]}[1]), \ldots, (s_1[z'], \ldots, s_{e'[z']}[z'])\big) \in S'$$

if $e'[z] \leq e[z]$ for all $z \in [z']$ since each tuple $(s_1[z], \ldots, s_{e'[z]}[z])$ is a prefix of $(s_1[z], \ldots, s_{e'[z]}[z])$. In this case, we also say that $s'' \in S'$ is a prefix of $s' \in S'$. Moreover, we say that configuration $s' \in S'$ is maximal in $\mathsf{state}_v[z]$ if i) B accepts s' and ii) any other $s'' \in S'$ accepted by B in state $\mathsf{state}_v[z]$ is a prefix of s'. For tuple $s = (s_1[z], \ldots, s_i[z]) \in S^i$ we define the size of s as $|s| = i$. For configuration

$$s' = \big((s_1[1], \ldots, s_{e[1]}[1]), \ldots, (s_1[z'], \ldots, s_{e[z']}[z'])\big) \in S'$$

we also define the size of s' as $|s'| = \sum_{i=1}^{z'} e[z]$ and the size of its zth component as $|s'|_z = e[z]$ for all $z \in [z']$. In our proof, we will rely on the fact that after each rewinding, $\mathsf{Extract}$ only slightly changes the configuration that B sees by appending a single value to some accepted configuration. Let us be more precise. Let s' be the maximal configuration that B accepts in $\mathsf{state}_v[z]$. Then in any step of $\mathsf{Extract}(z)$, the statements that B sees will form a configuration $s'' \in S'$ such that for all components $z'' \neq z$ we have i) $[s']_{z''} = [s']_{z''}$ (i.e. their projections to the z''th component are equal) ii) $[s']_z$ is a prefix of $[s'']_z$, and iii) $|s'|_z + 1 = |s''|_z$. This intuitively says that in $\mathsf{state}_v[z]$, $\mathsf{Extract}(z)$ will only send a single new statement $s'_{\ell,j}[z]$ to B that will form the tuple $s_1[z], \ldots, s_v[z], s'_{\ell,j}[z]$ in the communication with $A[z]$. The accepted statements in the communication with all other $A[z'']$ for $z'' \neq z$ remain the same. As a result, the failure probability of this event can be bounded as in the first proof by simply bounding how much $s'_{\ell,j}[z]$ differs from a statement that an ideal attacker would send.

Next, let us analyze the difference in the behaviour of the reduction B, when B communicates with M instead of the ideal attacker. A closer inspection reveals that the only case that the behavior may differ occurs if the reduction does abort in the run of $\mathsf{Extract}[z]$ but does not abort otherwise. Apart from that, the simulation is perfect.

First, observe that if $\mathsf{Extract}(z)$ finishes, it will always output p witnesses $w_1^*[z], \ldots, w_p^*[z]$.

Next, observe that except for the algorithms in $\mathsf{Extract}$ all operations in the description of the meta-reduction are efficient. Moreover, except for the back jump in Step 2'.$l.j.v$ of $\mathsf{Extract}$, all operations in $\mathsf{Extract}$ are efficient as well. Let us now ensure that this jump is taken at most an expected polynomial number of times. Once we have established this, the entire meta-reduction will have expected polynomial runtime. To this end, we proceed as in our first proof. Let us now focus on a run of $\mathsf{Extract}(z)$ on some arbitrary but fixed input z. First, compute the probability $\Pr[E_{\ell,j,v}]$ of the event $E_{\ell,j,v}$ that after sending $s'_{\ell,j}[z]$ in $\mathsf{state}_v[z]$, M has received back witness $w'_{\ell,j}[z] \in W$ while B has not made an external query. Recall that B will with probability ϵ' accept a random configuration $s' \in S'$. Since $\mathsf{Extract}$ is called on input z, M has already received $s_1^*[z], \ldots, s_p^*[z]$. This in particular means that B accept $[s']_z = (s_1[z], \ldots, s_{t+1}[z])$ when communicating with $A[z]$. This in particular means that to *any single* statement $s_v[z]$ of the tuple $(s_1[z], \ldots, s_{t+1}[z])$, the reduction will respond with a witness $w_v[z]$ to $A[z]$ with at least probability ϵ'. Recall that $s'_{l,j}[z]$ is distributed as $s_v[z]$ (except for statistically small probability δ). So, from the viewpoint of the reduction, the overall series of statements received so far $s_1[z], \ldots, s_{v-1}[z], s'_{l,j}[z]$ from $A[z]$ is distributed like the series of an ideal attacker except with probability δ. Thus, the reduction will respond to $s'_{\ell,j}[z]$ with witness $w'_{\ell,j}[z]$ with at least probability $\epsilon' - \delta$ in state $\mathsf{state}_v[z]$. Now, since we must always have that at least one of the $t+u$ statements that are delivered to B will neither invoke external queries to compute the response nor make B send a challenge statement to some other instance, we have that $\Pr[E_{\ell,j,v}] \geq (\epsilon' - \delta)/(t+u)$ for *at least one* $v \in [t+1]$. Next, compute the probability $\Pr[E_{\ell,j}]$ of the event $E_{\ell,j}$ that for *any* of the states $\mathsf{state}_v[z]$ with $v \in [t+1]$, reduction R has responded to query $s'_{\ell,j}[z]$ with $w'_{\ell,j}[z] \in W$ without making an external query. For this event we have that $\Pr[E_{\ell,j}] = \Pr[E_{\ell,j,1} \vee \ldots \vee E_{\ell,j,t+1}] \geq (\epsilon' - \delta)/(t+u)$. This means that after an expected number of $x = O(1/\Pr[E_{\ell,j}])$ iterations the meta-reduction will not jump back to 2'.ℓ. Let us finally compute the probability $\Pr[E_\ell]$ of the event E_ℓ that the reduction responds with $w'_{\ell,j}[z]$ to all the $s'_{\ell,j}[z]$ with $j \in [k]$ (for fixed ℓ) without making any external queries. We have that $\Pr[E_\ell] = \Pr[E_{\ell,1} \wedge \ldots \wedge E_{\ell,k}]$. Now, since any derived value $s'_{\ell,j}[z]$ is distributed statistically close to any of the $s_v[z]$ (and thus statistically close to a value that is independent of any other $s'_{\ell',j'}[z]$) we get that $\Pr[E_{\ell,i+1}|E_{\ell,i}] \geq (\epsilon' - \delta)/(t+u)$ for all $i = 1, \ldots, k-1$. Now, we have probability $\Pr[E_\ell] = \Pr[E_{\ell,1}] \cdot \prod_{i=1}^{k-1} \Pr[E_{\ell,i+1}|E_{\ell,i}] \geq ((\epsilon' - \delta)/(t+u))^k$. For any sufficiently large security parameter, we have that $\epsilon' \geq \delta$ since δ is statistically small and ϵ' is non-negligible. Thus $\Pr[E_\ell] \geq (\epsilon'/(2t+2u))^k$ and since k is constant, $\Pr[E_\ell]$ is non-negligible. Thus after at most a polynomial number of iterations $y = O(1/\Pr[E_\ell])$ the meta-reduction will finish the loop for each of the $\ell \in [p]$. Repeating this process for all ℓ while using that all other operations of M are indeed efficient, we now obtain that after an expected polynomial number of iterations $\mathsf{Extract}$ computes all p k-tuples $w'_\ell[z] = (w'_{\ell,1}[z], \ldots, w'_{\ell,k}[z])$ for $\ell \in [p]$. Now with the values received in this way, M can finally compute $w'_\ell[z]^* \leftarrow \mathsf{RSRWitness}(R[z], st[z], w'_\ell[z])$ and ultimately, after re-randomization, the meta-reduction outputs all the re-randomized $w_1^*[z], \ldots, w_p^*[z]$.

Since all arguments are essentially independent of the concrete choice of z, this shows that after expected polynomially time, $\mathsf{Extract}$ will always output the $w_1^*[z], \ldots, w_p^*[z]$ for any z. As B is a PPT algorithm we have that u is polynomial. Since $\mathsf{Extract}$ will be called at most $z' \leq u$ times, the meta-reduction runs in expected polynomial runtime overall.

Let us next show that with non-negligible probability, all $w'_{\ell,j}[z]$ are correct in the sense that $(s'_{\ell,j}[z], w'_{\ell,j}[z]) \in R[z]$. Observe that we have to consider at most a polynomial number $y = p \cdot k \cdot u$ of values $w'_{\ell,j}[z]$ that the meta-reduction will ultimately use to compute all the $w_\ell^*[z]$ for $\ell \in [p]$ and $z \in [z']$. Due to property P-9, each of these values is distributed like the values of the ideal attacker except with statistically small error γ. With probability $\zeta = (1-\gamma)^y$ we will thus have that all the $w'_{\ell,j}[z]$ are correct. For convenience let $\gamma' = \gamma \cdot y$ and observe that for sufficiently large security parameter we have that γ' is statistically small as well. But since $(1-\gamma)^y \geq (1 - y \cdot \gamma) = 1 - \gamma'$ due to Bernoulli's inequality, we have that ζ is statistically close to 1 as well.

Let us finally analyze the distribution of values produced by the meta-reduction and compare them to the ideal attacker. First observe that B will always see at most one single transformed statement $s'_{\ell,j}[z]$ at a time. Each time, this will only occur after a rewinding and as part of a tuple of statements $(s_1[z],\ldots,s_i[z],s'_{\ell,j}[z])$ that B receives. Since only the last statement is not distributed like the output of the ideal attacker, these distributions have statistical distance of at most δ. So with probability $1-\delta$ each $(s_1[z],\ldots,s_i[z],s'_{\ell,j}[z])$ looks distributed like the values from the ideal attacker. Since M runs in expected polynomial-time, it needs a polynomial number of rewindings overall y to compute all the $w^*[z]$. We now have that all the rewindings get accepted by B with probability $\zeta' = (1-\delta)^y \geq 1-y\delta$ that is statistically close to one. Now if the values received in this way are all correct, M can finally compute $w'^*_\ell[z] \leftarrow \mathsf{RSRWitness}(R[z], st[z], w'_\ell[z])$ and ultimately, after re-randomization, the meta-reduction outputs all the re-randomized $w^*_1[z],\ldots,w^*_p[z]$. Due to property P-4, the distribution of each of these values is distributed like the ideal attacker.

At last, observe that according to the view of the reduction B, B accepts a tuple of statements $(s_1[z]\ldots,s_{t+1}[z])$ for at least one $z \in [z']$. Since the $s_1[z],\ldots,s_{t+1}[z]$ are exactly distributed as the statements sent by the ideal attacker, the reduction will not be able to distinguish the meta-reduction from the ideal attacker in this case with probability. So overall, the distributions of values received by M and the ideal attacker are always identical, except with statistically small error δ. Finally, the meta-reduction has never made more than t queries to the challenger of the tICA as it is merely relaying queries from the reduction to the tICA challenger and back again.

To sum up, we have that i) the runtime of M is polynomially bounded, ii) that $\mathsf{Extract}(z)$ will always output a witness tuple $(w^*_1[z],\ldots,w^*_p[z])$, iii) that all the witnesses $w'_{\ell,j}[z]$ output by the reduction are correct with probability statistically close to 1, and iv) that the reduction B can never distinguish M from the ideal attacker with probability statistically close to one. Thus B must behave as if it communicates with the ideal attacker and finally output a solution to the tICA challenge.

Now with the values received in this way, M can compute $w'_\ell[z]^* \leftarrow \mathsf{RSRWitness}(R[z], st[z], w'_\ell[z])$ and ultimately, after re-randomization, the meta-reduction outputs all the re-randomized witnesses $w^*_1[z],\ldots,w^*_p[z]$. Due to property P-4, the distribution of each of these values is identical to what the ideal adversary produces. Thus, with non-negligible probability the reduction will accept $w^*[z]$ and output a solution to the tICA challenge.

12 Applications of Second Main Theorem

We now present several immediate applications of our second main theorem.

Corollary 7. *There is no PPT reduction that, while creating at most u instances, that can reduce the Paired OW-CCA1 security of a certified semi-homomorphic PKE with $t+u$-decryption queries for pairs of ciphertexts (Figure 4) to the security of any t-interactive complexity assumption.*

Corollary 8. *There is no PPT reduction that, while creating at most u instances, can reduce the IND-CCA1 security of a certified semi-homomorphic PKE with $2t+2u$-decryption queries overall (Figure 4) to the security of any t-interactive complexity assumption.*

Corollary 9. *There is no PPT reduction that, while creating at most u instances, can reduce the Paired OW-CCA1 security of any certified semi-homomorphic PKE with $t+u$-decryption queries to the Paired OW-CCA1 security of the same semi-homomorphic PKE with t-decryption queries.*

Corollary 10. *Let F denote an efficiently sampleable family of certified homomorphic one-way bijections. Then we have that there is no PPT reduction that, while creating at most u instances, can reduce the security of F with $t+u$-inversion queries to the security of any t-interactive complexity assumption.*

Corollary 11. *Let F denote an efficiently sampleable family of certified homomorphic one-way bijections. Then we have that there is no PPT reduction, while creating at most u instances, that can reduce the security of F with $t+u$-inversion queries to the security of F with any t-inversion queries (Figure 5).*

Since ElGamal is semi-homomorphic, we can immediately obtain strong separation results for ElGamal PKE.

Corollary 12. *There is no PPT reduction that, while creating at most u instances, can reduce the Paired OW-CCA1 security of ElGamal PKE with $t + u$ adaptive queries for pairs of ciphertexts to the security of some tICA (Figure 7).*

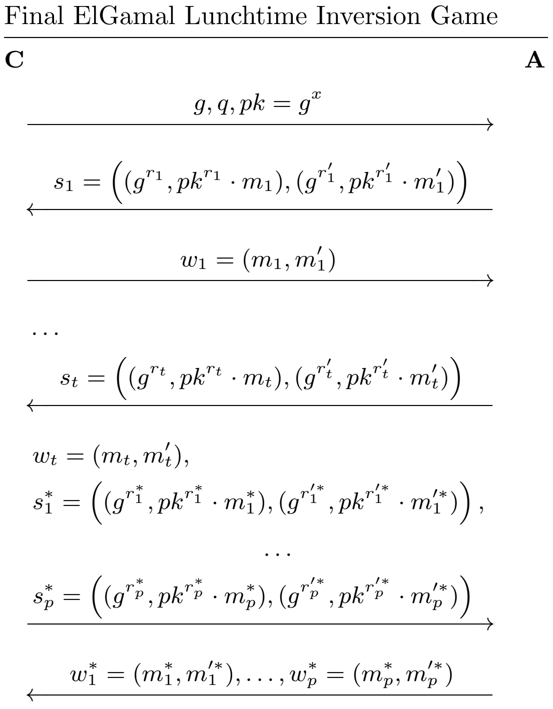
Final ElGamal Lunchtime Inversion Game
C
A
$g, q, pk = g^x$
$s_1 = \Big((g^{r_1}, pk^{r_1} \cdot m_1), (g^{r_1'}, pk^{r_1'} \cdot m_1')\Big)$
$w_1 = (m_1, m_1')$
$\dots$
$s_t = \Big((g^{r_t}, pk^{r_t} \cdot m_t), (g^{r_t'}, pk^{r_t'} \cdot m_t')\Big)$
$w_t = (m_t, m_t')$,
$s_1^* = \Big((g^{r_1^*}, pk^{r_1^*} \cdot m_1^*), (g^{r_1'^*}, pk^{r_1'^*} \cdot m_1'^*)\Big)$,
$\dots$
$s_p^* = \Big((g^{r_p^*}, pk^{r_p^*} \cdot m_p^*), (g^{r_p'^*}, pk^{r_p'^*} \cdot m_p'^*)\Big)$
$w_1^* = (m_1^*, m_1'^*), \dots, w_p^* = (m_p^*, m_p'^*)$

Fig. 7. Statements are pairs of ciphertexts. Witnesses are the corresponding plaintext pairs.

Corollary 13. *There is no PPT reduction that, while creating at most u instances, can reduce the IND-CCA1 security of a homomorphic PKE with $2t + 2u$ decryption queries overall to the security of some tICA.*

13 Extension to Non-Uniform Algorithms

So far, we have assumed that all algorithms are uniform. We are confident that our second main result can also be transferred to non-uniform algorithms if we restrict our attention to falsifiable assumptions. In particular, we can use the same methodology that was employed in [14] to show how to obtain non-uniform results for [39]. In a nutshell, this means that we can substitute all PPT algorithms by poly-sized circuits. Qualitatively we may now also rule out non-uniform reductions, however, we also need the meta-reduction to be non-uniform. This is theoretically incomparable to our main result. Let us provide some intuition of the essential challenges towards obtaining a non-uniform variant of our result. Unfortunately, the algorithm M from the non-uniform proof cannot be applied since, when dealing with a non-uniform reduction, M cannot guarantee that its statements are distributed independently random from B's point of view. However, this is necessary for the employed strategy where M tries to rewind queries and send in some new but equally distributed statements. In particular, we need that the distributions of the original and modified statement are equal (or statistically close) from B's point of view to argue that the behavior of M does not deviate from A too much. Otherwise, B could recognize that it is communicating with M and not A. To this end, we want that each message sent by M as a statement has full entropy like the messages computed by A. Intuitively, when we deal with a non-uniform reduction we run into the problem that B is given an advice string that may depend on A. So, there might be some potential queries for which we are not guaranteed full entropy anymore. Fortunately, [14] show while relying on a result of Unruh [45], that the number of problematic queries, that do in fact noticeably depend on the advice string, is polynomially bounded. All other possible queries still have high enough entropy. To deal with the problematic queries then we can simply assume that M is also non-uniform and that M receives as an advice string (which in turn depends on B) all of the problematic queries together with the responses by A at startup. In this way, M can also perfectly simulate the problematic queries.

14 Practical Impact and Ways to Circumvent our Results

Besides its theoretical importance, we consider our result to be of considerable value for the design of cryptographic systems. It can be used as a criterion to decide which draft designs are impossible to prove secure at very early stages of the design process. As for any impossibility result, it is advantageous that our result holds for very weak security games. Often the starting points of new cryptographic systems are simpler building blocks with weak security properties. Next, these building blocks are step-wisely improved to resist stronger attacks in stronger security models. It is thus beneficial that our impossibility result already manifests itself for weak notions (and before further effort has been spent to improve the basic designs).

From a more abstract point of view, it has been shown in many previous works that relations with unique witnesses, or more generally re-randomizable witnesses, can allow for meta-reduction-based impossibility results and often constructions must circumvent these results. Our results further support this. At the same time, they show that when basing cryptographic designs on random self-reducible relations we have to be even more careful since impossibility results hold more broadly. This is particularly interesting in light of the usefulness of security assumptions that are based on RRRs for developing tight security reductions.

Let us consider some general ways to circumvent our and previous results. One reliable way to do this is to use relations with at least two witnesses and no efficient re-randomization procedure. A strategy that can be based on this uses proofs involving so-called partitioning arguments. The idea is to let the reduction secretly partition the space of witnesses for each statement into two sets. The first set contains witnesses that the reduction can efficiently produce. They will be used in the query phase. However, the second set contains witnesses that the reduction cannot produce herself but which she can use to extract a solution to the underlying complexity assumption. Of course, this technique crucially relies on the fact that the two witness sets are indistinguishable to the attacker so that with good probability the attacker will indeed output a witness from the second set.

Other techniques that might help to circumvent our results rely on the programmability of the random oracle model as exemplified in [28]. In a nutshell, one way to view the uniqueness requirement is that it makes each statement a perfectly binding commitment to its witness. As such, by publishing statements, the reduction fixes critical witnesses at an early stage. However, using the programmability of the random oracle we may delay the specification of the witnesses to some later point in time by essentially implementing commitments that are non-committing for the reduction and can be opened on the fly.

Finally, our result does not exclude reductions that depend on the attacker. However, we stress that, although there has been considerable progress in recent years, it seems that for arbitrary attackers current techniques for general non-black box reductions are still not able to tackle this problem without major breakthroughs.

References

1. Armknecht, F., Katzenbeisser, S., Peter, A.: Group homomorphic encryption: characterizations, impossibility results, and applications. Des. Codes Cryptogr. **67**(2), 209–232 (2013). https://doi.org/10.1007/S10623-011-9601-2, https://doi.org/10.1007/s10623-011-9601-2
2. Auerbach, B., Cash, D., Fersch, M., Kiltz, E.: Memory-tight reductions. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part I. LNCS, vol. 10401, pp. 101–132. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 20–24, 2017). https://doi.org/10.1007/978-3-319-63688-7_4

3. Bader, C., Jager, T., Li, Y., Schäge, S.: On the impossibility of tight cryptographic reductions. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 273–304. Springer, Heidelberg, Germany, Vienna, Austria (May 8–12, 2016). https://doi.org/10.1007/978-3-662-49896-5_10
4. Baecher, P., Brzuska, C., Fischlin, M.: Notions of black-box reductions, revisited. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 296–315. Springer, Heidelberg, Germany, Bengalore, India (Dec 1–5, 2013). https://doi.org/10.1007/978-3-642-42033-7_16
5. Baldimtsi, F., Lysyanskaya, A.: On the security of one-witness blind signature schemes. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part II. LNCS, vol. 8270, pp. 82–99. Springer, Heidelberg, Germany, Bengalore, India (Dec 1–5, 2013). https://doi.org/10.1007/978-3-642-42045-0_5
6. Bellare, M., Namprempre, C., Pointcheval, D., Semanko, M.: The power of RSA inversion oracles and the security of Chaum's RSA-based blind signature scheme. In: Syverson, P.F. (ed.) FC 2001. LNCS, vol. 2339, pp. 319–338. Springer, Heidelberg, Germany, Grand Cayman, British West Indies (Feb 19–22, 2002)
7. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 15–19, 2004). https://doi.org/10.1007/978-3-540-28628-8_3
8. Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2001). https://doi.org/10.1007/3-540-44647-8_13
9. Boneh, D., Venkatesan, R.: Breaking RSA may not be equivalent to factoring. In: Nyberg, K. (ed.) EUROCRYPT'98. LNCS, vol. 1403, pp. 59–71. Springer, Heidelberg, Germany, Espoo, Finland (May 31 – Jun 4, 1998). https://doi.org/10.1007/BFb0054117
10. Bresson, E., Monnerat, J., Vergnaud, D.: Separation results on the "one-more" computational problems. In: Malkin, T. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 71–87. Springer, Heidelberg, Germany, San Francisco, CA, USA (Apr 7–11, 2008). https://doi.org/10.1007/978-3-540-79263-5_5
11. Brown, D.R.L.: Irreducibility to the one-more evaluation problems: More may be less. Cryptology ePrint Archive, Report 2007/435 (2007), http://eprint.iacr.org/
12. Canetti, R., Goldreich, O., Goldwasser, S., Micali, S.: Resettable zero-knowledge (extended abstract). In: 32nd ACM STOC. pp. 235–244. ACM Press, Portland, OR, USA (May 21–23, 2000). https://doi.org/10.1145/335305.335334
13. Canetti, R., Lin, H., Pass, R.: Adaptive hardness and composable security in the plain model from standard assumptions. In: 51st FOCS. pp. 541–550. IEEE Computer Society Press, Las Vegas, NV, USA (Oct 23–26, 2010). https://doi.org/10.1109/FOCS.2010.86
14. Chung, K., Lin, H., Mahmoody, M., Pass, R.: On the power of nonuniformity in proofs of security. In: Kleinberg, R.D. (ed.) ITCS '13, Berkeley, CA, USA, January 9-12, 2013. pp. 389–400. ACM (2013). https://doi.org/10.1145/2422436.2422480
15. Coron, J.S.: Optimal security proofs for PSS and other signature schemes. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 272–287. Springer, Heidelberg, Germany, Amsterdam, The Netherlands (Apr 28 – May 2, 2002). https://doi.org/10.1007/3-540-46035-7_18
16. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO'98. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 23–27, 1998). https://doi.org/10.1007/BFb0055717
17. Damgård, I., Jurik, M.: A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In: Kim, K. (ed.) PKC 2001. LNCS, vol. 1992, pp. 119–136. Springer, Heidelberg, Germany, Cheju Island, South Korea (Feb 13–15, 2001). https://doi.org/10.1007/3-540-44586-2_9
18. Deng, Y., Goyal, V., Sahai, A.: Resolving the simultaneous resettability conjecture and a new non-black-box simulation strategy. In: 50th FOCS. pp. 251–260. IEEE Computer Society Press, Atlanta, GA, USA (Oct 25–27, 2009). https://doi.org/10.1109/FOCS.2009.59
19. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Trans. Inf. Theory **22**(6), 644–654 (1976). https://doi.org/10.1109/TIT.1976.1055638
20. Dodis, Y., Oliveira, R., Pietrzak, K.: On the generic insecurity of the full domain hash. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 449–466. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2005). https://doi.org/10.1007/11535218_27
21. Dodis, Y., Reyzin, L.: On the power of claw-free permutations. In: Cimato, S., Galdi, C., Persiano, G. (eds.) SCN 02. LNCS, vol. 2576, pp. 55–73. Springer, Heidelberg, Germany, Amalfi, Italy (Sep 12–13, 2003). https://doi.org/10.1007/3-540-36413-7_5
22. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO'84. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 1984)

23. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.: An algebraic framework for Diffie-Hellman assumptions. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 129–147. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 18–22, 2013). https://doi.org/10.1007/978-3-642-40084-1_8
24. Fischlin, M., Fleischhacker, N.: Limitations of the meta-reduction technique: The case of Schnorr signatures. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 444–460. Springer, Heidelberg, Germany, Athens, Greece (May 26–30, 2013). https://doi.org/10.1007/978-3-642-38348-9_27
25. Fleischhacker, N., Jager, T., Schröder, D.: On tight security proofs for Schnorr signatures. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part I. LNCS, vol. 8873, pp. 512–531. Springer, Heidelberg, Germany, Kaoshiung, Taiwan, R.O.C. (Dec 7–11, 2014). https://doi.org/10.1007/978-3-662-45611-8_27
26. Fuchsbauer, G., Kiltz, E., Loss, J.: The algebraic group model and its applications. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 33–62. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2018). https://doi.org/10.1007/978-3-319-96881-0_2
27. Garg, S., Bhaskar, R., Lokam, S.V.: Improved bounds on security reductions for discrete log based signatures. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 93–107. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 17–21, 2008). https://doi.org/10.1007/978-3-540-85174-5_6
28. Guo, F., Chen, R., Susilo, W., Lai, J., Yang, G., Mu, Y.: Optimal security reductions for unique signatures: Bypassing impossibilities with a counterexample. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 517–547. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 20–24, 2017). https://doi.org/10.1007/978-3-319-63715-0_18
29. Hanaoka, G., Matsuda, T., Schuldt, J.C.N.: On the impossibility of constructing efficient key encapsulation and programmable hash functions in prime order groups. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 812–831. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2012). https://doi.org/10.1007/978-3-642-32009-5_47
30. Hofheinz, D., Jager, T., Knapp, E.: Waters signatures with optimal security reduction. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 66–83. Springer, Heidelberg, Germany, Darmstadt, Germany (May 21–23, 2012). https://doi.org/10.1007/978-3-642-30057-8_5
31. Kakvi, S.A., Kiltz, E.: Optimal security proofs for full domain hash, revisited. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 537–553. Springer, Heidelberg, Germany, Cambridge, UK (Apr 15–19, 2012). https://doi.org/10.1007/978-3-642-29011-4_32
32. Lindell, Y.: Is ElGamal IND-CCA1 Secure? – Answer. https://crypto.stackexchange.com/questions/26867/is-elgamal-ind-cca1 (2015), [Online; accessed 03-April-2022]
33. Lindell, Y.: Fast secure two-party ECDSA signing. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 613–644. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 20–24, 2017). https://doi.org/10.1007/978-3-319-63715-0_21
34. Lipmaa, H.: On the CCA1-security of Elgamal and Damgård's Elgamal. Cryptology ePrint Archive, Report 2008/234 (2008), https://ia.cr/2008/234
35. Morgan, A., Pass, R.: On the security loss of unique signatures. In: Beimel, A., Dziembowski, S. (eds.) TCC 2018, Part I. LNCS, vol. 11239, pp. 507–536. Springer, Heidelberg, Germany, Panaji, India (Nov 11–14, 2018). https://doi.org/10.1007/978-3-030-03807-6_19
36. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT'99. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg, Germany, Prague, Czech Republic (May 2–6, 1999). https://doi.org/10.1007/3-540-48910-X_16
37. Paillier, P., Vergnaud, D.: Discrete-log-based signatures may not be equivalent to discrete log. In: Roy, B.K. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 1–20. Springer, Heidelberg, Germany, Chennai, India (Dec 4–8, 2005). https://doi.org/10.1007/11593447_1
38. Paillier, P., Villar, J.L.: Trading one-wayness against chosen-ciphertext security in factoring-based encryption. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 252–266. Springer, Heidelberg, Germany, Shanghai, China (Dec 3–7, 2006). https://doi.org/10.1007/11935230_17
39. Pass, R.: Limits of provable security from standard assumptions. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd ACM STOC. pp. 109–118. ACM Press, San Jose, CA, USA (Jun 6–8, 2011). https://doi.org/10.1145/1993636.1993652
40. Pass, R., Venkitasubramaniam, M.: On constant-round concurrent zero-knowledge. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 553–570. Springer, Heidelberg, Germany, San Francisco, CA, USA (Mar 19–21, 2008). https://doi.org/10.1007/978-3-540-78524-8_30

41. Pointcheval, D., Stern, J.: Security proofs for signature schemes. In: Maurer, U.M. (ed.) EUROCRYPT'96. LNCS, vol. 1070, pp. 387–398. Springer, Heidelberg, Germany, Saragossa, Spain (May 12–16, 1996). https://doi.org/10.1007/3-540-68339-9_33
42. Reingold, O., Trevisan, L., Vadhan, S.P.: Notions of reducibility between cryptographic primitives. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 1–20. Springer, Heidelberg, Germany, Cambridge, MA, USA (Feb 19–21, 2004). https://doi.org/10.1007/978-3-540-24638-1_1
43. Richardson, R., Kilian, J.: On the concurrent composition of zero-knowledge proofs. In: Stern, J. (ed.) EUROCRYPT'99. LNCS, vol. 1592, pp. 415–431. Springer, Heidelberg, Germany, Prague, Czech Republic (May 2–6, 1999). https://doi.org/10.1007/3-540-48910-X_29
44. Seurin, Y.: On the exact security of Schnorr-type signatures in the random oracle model. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 554–571. Springer, Heidelberg, Germany, Cambridge, UK (Apr 15–19, 2012). https://doi.org/10.1007/978-3-642-29011-4_33
45. Unruh, D.: Random oracles and auxiliary input. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 205–223. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2007). https://doi.org/10.1007/978-3-540-74143-5_12
46. Wu, J., Stinson, D.R.: On the security of the ElGamal encryption scheme and Damgård's variant. IACR Cryptol. ePrint Arch. p. 200 (2008), http://eprint.iacr.org/2008/200
47. Zhang, J., Zhang, Z., Chen, Y., Guo, Y., Zhang, Z.: Black-box separations for one-more (static) CDH and its generalization. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part II. LNCS, vol. 8874, pp. 366–385. Springer, Heidelberg, Germany, Kaoshiung, Taiwan, R.O.C. (Dec 7–11, 2014). https://doi.org/10.1007/978-3-662-45608-8_20

A RRR with Non-Unique Witnesses

For illustration, we will now provide an example of an RRR system where we have $S' \neq S$ and non-unique witnesses. In particular, the difference between S and S' will be that all elements in S' have a representation with parity bit 1 (the XOR of all the bits of its binary representation is 1) whereas for S this is not required. Our RRR system works in the common setting of symmetric bilinear pairings. Consider the following RRR system RRR_e with fan-out $k = 1$:

- $\mathsf{RGen}(1^\kappa)$ computes two groups G, G_T of order q (with polynomial bitlength $|q|_2$) that are equipped with an efficient, bilinear, and non-degenerate map $e : G \times G \to G_T$ and two generators g, h of G. It outputs a description of

$$R_e = \{(e(x_1, g) \cdot e(x_2, h), (x_1, x_2)) | (x_1, x_2) \in G\}$$

  alongside g, h, and e. We have $W = G^2$, $S = G_T$, and $S' \subsetneq S$ such that any $s \in S'$ has parity bit 1.
- $\mathsf{RSample}(R_e)$ draws uniformly random $w_1, w_2 \in G$ and outputs $s = e(w_1, g) \cdot e(w_2, h)$ and $w = (w_1, w_2)$.
- $\mathsf{RSubSample}(R_e)$ draws random group elements $w_1, w_2 \in G$ until $s' = e(w_1, g) \cdot e(w_2, h)$ has parity bit 1. Then it outputs s' and $w = (w_1, w_2)$.
- $\mathsf{ReRand}(s, w)$ draws $r \in \mathbb{Z}_q$ and output $w' = (w_1 h^r, w_2 g^{-r})$.
- $\mathsf{ReCheck}(s, w, w')$ where $w' = (w'_1, w'_2)$ output 1 if $e(w'_1, g) \cdot e(w'_2, h) = s$ and otherwise 0.
- $\mathsf{RSRStatement}(s^*)$ repeatedly draws random $r \in \mathbb{Z}_q$ $(r \neq 0)$ until the representation of $(s^*)^r$ has parity bit 1 to compute and output $(s, st) = ((s^*)^r, (s^*, r))$.
- $\mathsf{RSRWitness}(w, st)$ parses st as $st = (s^*, r)$ and w as $w = (w_1, w_2)$ and output $w^* = (w_1^{1/r}, w_2^{1/r})$.
- $\mathsf{RSRTest}(w, st)$ parses st as $st = (s^*, r)$ and w as $w = (w_1, w_2)$, computes $w^* = (w_1^*, w_2^*) = (w_1^{1/r}, w_2^{1/r})$ and outputs 1 if $e(w_1^*, g) \cdot e(w_2^*, h) = s^*$ and otherwise 0.
- $\mathsf{RVerify}(u)$ outputs 1 iff q is prime and of appropriate size and g, h are indeed generators of G and e is a degenerate bilinear pairing as required.

Lemma 6. RRR_e *is a certified strong RRR system.*

Proof. First observe that $\mathsf{RVerify}(u)$ can efficiently test all important properties such that P-13 and P-14 are fulfilled. Also, observe that by choosing r in ReRand we uniformly draw a new witness w' for the same statement independent of the input witness w. Now observe that $e(w'_1, g) \cdot e(w'_2, h) = s$

for $w' = (w'_1, w'_2) = (w_1 h^r, w_2 g^{-r})$ always simplifies to $e(w_1, g) \cdot e(w_2, h) = s$. These facts imply that the re-randomization algorithm is correct, that it provides witness indistinguishability, and that ReCheck is correct and sound showing P-3, P-4, P-5. Observe that the output $s = (s^*)^r$ of RSRStatement blinds s^* perfectly. This shows P-7. Moreover, given a witness $w = ((w_1^*), (w_2^*))$ for s such that $s = (s^*)^r = e(w_1, g) \cdot e(w_2, h)$ we can easily see that $w^* = ((w_1)^{1/r}, (w_2)^{1/r})$ is a witness for s^*. This shows P-6 and P-9. Finally, since e is efficient, we also have P-1 and P-2. P-8 follows since the output of RSRStatement is uniform over the elements in S with parity bit 1 in the first k coordinates, exactly like the output of RSubSample. □

B Impossibility of Simple Reductions for OW-CCA1-Security of ElGamal PKE

To see our arguments in a more concrete setting, we provide a stand-alone proof for the ElGamal PKE scheme. We concentrate on the first main result. Recall the definition of ElGamal PKE scheme in Section 4.3 and consider the security notion of OW-CCA1 security that is strictly weaker than IND-CCA1 (lunchtime) security. The resulting security game can be found in Figure 6.

B.1 Application of First Main Result

Now let us formally state an application of our first main result when applied to ElGamal PKE.

Theorem 3. *Let* ICA *be a secure tICA. Then, there is no* simple *reduction that can reduce the lunchtime security of ElGamal PKE with* $t + 1$ *adaptive decryption queries to the security of* ICA.

B.2 Proof of Theorem 3

In the proof, we concentrate on an impossibility result under OW-CCA1 security. This immediately implies an impossibility result against IND-CCA1 attacks since the OW-CCA1 notion is weaker than IND-CCA1.

We proceed exactly as in the general proof but are able to make considerable simplifications. We specify an ideal (unbounded) attacker A for which the reduction has to work. Next, we present an efficient meta-reduction M that simulates the ideal attacker. Finally, we analyze the difference between the behavior of reduction B in the two cases. We show that the reduction will not be able to tell the two settings apart. Thus, the reduction also has to work for the efficient M. The combination of B and M will therefore break the underlying tICA. The meta-reduction gains its power from rewinding the reduction.

In a nutshell, the proof relies on the fact that we can always find *a useful rewinding spot* after rewinding. A useful rewinding spot is a decryption query c' that is sent after the reduction has been rewound to some previous state such that i) the reduction provides a (correct) plaintext m' to that query and ii) the reduction does not query its tICA challenger before delivering the response. Intuitively, the meta-reduction will repeatedly try to hit a useful rewinding spot and send new decryption queries c' to B that are derived from the challenge ciphertext to the reduction until it receives back message m'. The proof exploits that in each of these runs, with non-negligible probability, the reduction will behave as in the first run if presented with a new decryption query since new queries are exactly distributed as in the first run. Let us now be more formal.

B.3 The Ideal Attacker A

1. The attacker A receives random coins r and pk. The attacker M receives $pk = (g, h, q)$ and random coins r from B. If g does not generate a group G of prime order q or if $h \notin G$ the attacker aborts. Otherwise, it continues.
2. The parties repeat the following steps for all $i \in [1; t + 1]$.

   2.i.1 The attacker A computes a ciphertext/plaintext pair by drawing random message $m_i \in G$ and computing $c_i \leftarrow$ PKE.Enc(pk, m_i). Next, it sends c_i to the challenger.

2.i.2 The challenger responds with message m'_i. The attacker aborts in the case that $m_i \neq m'_i$. Otherwise, it continues. If $i = t$, the challenger also sends challenge ciphertext $c^* \in C$. If $c^* \notin C$ the attacker aborts. Otherwise, it continues.

3. The attacker uses its unbounded power to compute the plaintext $m^* \in G$ in c^*. To this end, it could for example compute all possible messages $m \in G$ and all possible ciphertexts $c \leftarrow \mathsf{PKE.Enc}(pk, m)$ for all possible random coins. In this way, it can find m^* such that for some random coins we have $c^* \leftarrow \mathsf{PKE.Enc}(pk, m^*)$. The value m^* is the final output of the attacker.

B.4 The Meta-Reduction *M* can Rewind Reduction *B*

We will now consider a meta-reduction M that executes the purported reduction B. In particular, we assume that M can store the full execution state state_i of B after B has sent some message and awaits a corresponding response. With these states, M can rewind B to a previous point in time by loading the corresponding execution states. Let us now specify how the meta-reduction simulates the ideal attacker.

B.5 The Simulated Attacker

0. The attacker M receives the tICA instance c and relays it to B along with random coins $r_B \in D_B$.
1. The attacker M receives $pk = (g, h, q)$ and random coins r from B. If g does not generate a group G of prime order q or if $h \notin G$, M aborts. Otherwise, it continues.
2. The parties repeat the following steps for all $i \in [1; t + 1]$.

   2.i.1 First the attacker M stores B's execution state state_i. The attacker M computes a ciphertext/plaintext pair by drawing random message $m_i \in G$ and computing $c_i \leftarrow \mathsf{PKE.Enc}(pk, m_i)$. Next, it sends c_i to the reduction.

   2.i.2 The reduction responds with a message $m'_i \in G$. If the reduction B outputs a query to its tICA challenger, this query is simply relayed by M to its tICA challenger. Likewise, all responses are relayed back to B. The attacker M aborts if $m_i \neq m'_i$. Otherwise, it continues. If $i = t$ the reduction also sends challenge statement c^* to M. If $c^* \notin C$ the meta-reduction aborts. Otherwise, it continues.

2' Once the attacker has received the challenge statement and checked if $c^* \in C$, it halts B and stores the current execution state state^* of B. Let $c^* = (c^*_1, c^*_2)$ and denote the plaintext stored in c^* as m^* such that for some randomness r^* we have $c^* = \mathsf{PKE.Enc}(pk, m^*; r^*) = (g^{r^*}, pk^{r^*} \cdot m^*)$. Then, the meta-reduction computes a derived statement $c' = (c'_1, c'_2)$ by drawing uniformly random $m'' \in G$ and random $r'' \in \mathbb{Z}_q$ and computing

$$c' = (c^*_1 \cdot g^{r''}, c^*_2 \cdot pk^{r''} \cdot m'') = (g^{r^*+r''}, pk^{r^*+r''} \cdot m^* \cdot m'').$$

Observe that with this choice c' is distributed like an encryption of a random group element $m' = m^* \cdot m''$ with fresh randomness. In particular, c' is independent of c^*. The meta-reduction iterates through all possible states state_v. To this end, it repeats the following loop for all state indices $v \in [t + 1]$:

   2'.v The meta-reduction rewinds the reduction B back to the point before [2.v.1] by loading state_v. Next, it sends c' to B. If the reduction outputs a message $m' \in G$ while not making a query to its tICA challenger, the meta-reduction leaves this loop immediately (break[12]). Otherwise, it checks whether $v = t + 1$ indicating that it has tried to send c' in all possible states. On success, the meta-reduction jumps back to 2' and repeats the entire computation with fresh randomness for r'' and m''.

3. M computes $m^* = m'/m''$, loads B's state state^*, and outputs the message m^*.
4. B responds with a solution to the tICA challenge.
5. Finally, M relays that solution to the tICA challenger.

[12] The meta-reduction continues at Step 2'.v' for $v' = v + 1$.

B.6 Analysis

Consider a reduction B that breaks the underlying security assumption with probability ϵ when communicating with an attacker. The splitting lemma guarantees with non-negligible probability $\epsilon' = \epsilon/2$, that the randomness $r_B \in D_B$ used by the reduction B will make B accept at least a fraction of $\epsilon' = \epsilon/2$ of all the possible query tuples $(c_1, \ldots, c_{t+1}) \in C^{t+1}$. To see this set $U = D_B$ to be the randomness space for B and $V = C^{t+1}$ to be the space of all possible statement tuples and let G be all $(u, v) \in U \times V$ that make B break the tICA. By assumption we have that $|G|/|U \times V| \geq \epsilon$. In terms of the splitting lemma, the randomness r_B is thus a super-good element. In particular, for super-good randomness r_B, B will with probability ϵ' accept a randomly generated tuple $(c_1, \ldots, c_{t+1}) \in C^{t+1}$ that is computed by the ideal attacker A via an encryption $c_i \leftarrow \mathsf{PKE.Enc}(pk, m_i)$ for each $i \in [t+1]$. Each such tuple will make the reduction output the challenge ciphertext next and if the reduction B is provided corresponding message m^*, B will break the tICA. Observe that if B accepts $c_1, \ldots, c_{i+1}$ for any $i \in [t]$ it will trivially also accept $c_1, \ldots, c_i$. In the following, we will thus concentrate on a single, fixed randomness r_B that the reduction uses and assume it to be super-good. This accounts for at most an additional non-negligible decrease of the overall success probability by a factor of ϵ'.

Let us analyze the difference between the execution of a reduction with an ideal attacker and the meta-reduction M from B's point of view. A closer inspection reveals that the only case that the behavior may differ occurs if the reduction does abort in Step 2' but does not abort otherwise or if in Step 2' the reduction delivers to the attacker a response m' such that for c' we have $\mathsf{PKE.Dec}(sk, c') \neq m'$. Otherwise, the simulation is perfect. Let us thus now analyze this event in more detail.

We start with a useful observation. We observe that for each tuple $(c_1, \ldots, c_{t+1}) \in C^{t+1}$ that the reduction accepts there is always one c_i $i \in [t+1]$ such that i) the reduction does not make a query to its tICA challenger before delivering the response and ii) the reduction has provided a correct message in the first run.

The first condition is guaranteed simply because the number of queries allowed in the security game, $t+1$, is larger than the number of queries allowed in the communication with the tICA challenger. The second condition is true since the meta-reduction (in the first run) and the ideal attacker, both know the messages within the ciphertexts queries they send. Thus they can always verify the responses and any successful reduction is bound to deliver correct messages m_i for every $i \in [t]$. Next, observe that all the values given to B as query c_i (including those after rewindings) have the same distribution. So the statements given to the reduction before and after the rewinding cannot be told apart.

Next observe that if the meta-reduction finishes its computations in Step 2', it will always compute m'.

Let us now show that the meta-reduction runs in expected polynomial time. First, observe that all operations performed in step 2'.v are efficient. To argue that the entire meta-reduction is efficient, we must now specifically show that the jumps in Step 2'.v back to 2' will not make the overall runtime super-polynomial. To this end, we have to analyze how likely such jumps are. To this end, compute the probability $\Pr[E_v]$ of the event E_v that after sending c' in state_v, M has received back witness m' while B has not made an external query. Recall that B accepts a random tuple of ciphertexts $(c_1, \ldots, c_{t+1})$ (before it outputting challenge ciphertext c^*) with probability at least ϵ'. This in particular means that to *any single* ciphertext c_v with $v \in [t+1]$, the reduction will respond with a witness m_v with at least probability ϵ'. Recall that c' is distributed as c_v. So, from the viewpoint of the reduction, the overall tuple of statements received so far $(c_1, \ldots, c_{v-1}, c')$ is distributed like the values of an ideal attacker. Thus, the reduction will respond to c' with witness m' with at least probability ϵ' in state state_v. Now, since we must always have that at least one of the $t+1$ ciphertexts that are delivered to B in an accepting tuple will not invoke external queries to compute the response, we have that $\Pr[E_v] \geq \epsilon'/(t+1)$ for *at least one* of the possible states state_v with $v \in [t+1]$. Next, compute the probability $\Pr[E]$ of the event E that for *any* of the states state_v with $v \in [t+1]$, reduction B has responded to query c' with m' without making an external query. For this event we have that $\Pr[E] = \Pr[E_1 \vee \ldots \vee E_{t+1}] \geq \epsilon'/(t+1)$. This means that after an expected number of $x = O(1/\Pr[E])$ iterations the meta-reduction will not jump back to $2'.\ell$.

Now let us now bound the probability that the message m' received by M in the rewinding process is indeed the correct plaintext encrypted in c' such that there is randomness $r' \in \mathbb{Z}_q$ with

$$c' = \left(g^{r'}, pk^{r'} m'\right) .$$

This bounding process is necessary because the meta-reduction cannot actually test whether the output m' is indeed correct. We have that the reduction responds with a *correct* witness m' per ciphertext c' with probability at least ϵ'. This is because i) in the first run, each ciphertext c_i will make the reduction respond with m_i with probability at least ϵ' and ii) the c' is distributed exactly like the c_i. So the probability that m' is correct is at least ϵ'.

Let us finally analyse the overall distribution of values produced by the meta-reduction and compare them to the ideal attacker. This is simple. First observe that in each rewinding attempt, B will at most see a single transformed ciphertext c', each time as the last component of a tuple of received ciphertexts $(c_1, \ldots, c_i, c')$. All of these values are distributed exactly like the values of the ideal attacker. Now, if the values received in this way are all correct, M can finally compute m' and ultimately m^*. Since there is only a single plaintext per ciphertext (for fixed pk) we have that m^* is also exactly distributed like the output of the ideal attacker. Thus, the probability of telling the ideal attacker and the meta-reduction apart is zero.

To sum up, we have that i) the runtime of M is polynomially bounded, ii) that it will always output m^*, iii) that the value m' received in the rewinding process is correct with non-negligible probability, and vi) that the reduction B cannot distinguish M from A. Moreover, we have that v) by the setup, M makes at most t queries to the ICA challenger. Thus B outputs after expected polynomial time a solution to the tICA challenge when given m^* with non-negligible probability.