Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 374 (2007) 229–260
Theoretical Computer Science
www.elsevier.com/locate/tcs

Hybrid commitments and their applications to zero-knowledge proof systems[☆]

Dario Catalano[a], Ivan Visconti[b,*]

[a] Dip. di Matematica e Informatica, Università di Catania, Viale Andrea Doria 6, 95126 Catania, Italy
[b] Dip. di Informatica ed Appl., Università di Salerno, Via Ponte Don Melillo, 84084 Fisciano, SA, Italy

Received 4 October 2005; received in revised form 30 November 2006; accepted 15 January 2007

Communicated by A. Fiat

Abstract

We introduce the notion of hybrid trapdoor commitment schemes. Intuitively a hybrid trapdoor commitment scheme is a primitive which can be either an unconditionally binding commitment scheme or a trapdoor commitment scheme depending on the distribution of commitment parameters. Moreover, such two possible distributions are computationally indistinguishable. Hybrid trapdoor commitments are related but different with respect to *mixed* commitments (introduced by Damgård and Nielsen at Crypto 2002). In particular hybrid trapdoor commitments can either be polynomially trapdoor commitments or unconditionally binding commitments, while mixed commitments can be either trapdoor commitments or extractable commitments. In this paper we show that strong notions (e.g., simulation sound, multi-trapdoor) of hybrid trapdoor commitments admit constructions based on the sole assumption that one-way functions exist as well as efficient constructions based on standard number-theoretic assumptions. To further stress the difference between hybrid and mixed commitments, we remark here that mixed commitments seem to require stronger theoretical assumptions (and the known number-theoretic constructions are less efficient). Our main result, is to show how to construct concurrent and simulation-sound zero-knowledge proof systems (in contrast to the arguments recently presented in [I. Damgård, Efficient concurrent zero-knowledge in the auxiliary string model, in: Advances in Cryptology — Eurocrypt'00, in: Lecture Notes in Computer Science, vol. 1807, Springer-Verlag, 2000, pp. 418–430; P. MacKenzie, K. Yang, On simulation-sound trapdoor commitments, in: Advances in Cryptology — Eurocrypt'04, in: Lecture Notes in Computer Science, vol. 3027, Springer-Verlag, 2004, pp. 382–400; R. Gennaro, Multi-trapdoor commitments and their applications to proofs of knowledge secure under concurrent man-in-the-middle attacks, in: Advances in Cryptology — Crypto'04, in: Lecture Notes in Computer Science, vol. 3152, Springer-Verlag, 2004, pp. 220–236]) in the common reference string model. We crucially use hybrid trapdoor commitments since we present general constructions based on the sole assumption that one-way functions exist and very efficient constructions based on number-theoretic assumptions.

Keywords: Commitment schemes; Zero-knowledge proofs

[☆] A preliminary version of this paper appears in [D. Catalano, I. Visconti, Hybrid trapdoor commitments and their applications, in: 32nd International Colloquium on Automata, Languages, and Programming, ICALP 05, in: Lecture Notes in Computer Science, vol. 3580, Springer-Verlag, 2005, pp. 298–310. [20].]

[*] Corresponding author.
E-mail addresses: catalano@dmi.unict.it (D. Catalano), visconti@dia.unisa.it (I. Visconti).

0304-3975/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2007.01.007

1. Introduction

Commitment schemes are arguably among the most important and useful primitives in cryptography. Intuitively a commitment scheme can be seen as the digital equivalent of a sealed envelope. If a party A wants to commit to some message m she just puts it into the sealed envelope, so that whenever A wants to reveal the message, she opens the envelope. Clearly, such a mechanism can be useful only if it meets some basic requirements. First of all the digital envelope should *hide* the message: no party other than A should be able to learn m from the commitment (this is often referred in the literature as the hiding property). Second, the digital envelope should be *binding*, meaning with this that A can not change her mind about m, and by checking the opening of the commitment one can verify that the obtained value is actually the one A had in mind originally (this is often referred to as the binding property). These two properties make commitments very useful in a wide range of cryptographic applications such as zero-knowledge protocols, multi-party computation, digital auctions and electronic commerce.

A *commitment scheme* is a primitive to generate and open commitments. More precisely a commitment scheme is a two-phase protocol between two probabilistic polynomial time algorithms `sender` and `receiver`. In a first stage (called the *commitment* phase) `sender` commits to a message m using some appropriate function `Com` which takes as input m and some auxiliary value r and produces as output a value c. The value c is sent to `receiver` as a commitment on m. In the second stage (called the *decommitment* phase) `sender` "convinces" `receiver` that c is actually a valid commitment on m (if `receiver` is not convinced, it just outputs some special string). A commitment scheme with this form is called non-interactive since both stages require only one message from `sender` to `receiver`. The requirements that we make on a commitment scheme are the following ones. First, if both `sender` and `receiver` behave honestly, then at the end of the decommitment phase `receiver` is convinced that `sender` had committed to bit m with probability 1. This is often referred to as the *correctness* requirement. Second a cheating `receiver` can not guess m with respect to any other legal message m' with probability significantly better than $1/2$. This is the so-called *hiding* property. Finally, a cheating `sender` should be able to open a commitment (i.e., to decommit) with both m and $m' \neq m$ only with very small (i.e., negligible) probability (this is the *binding* property). Each of the last two properties (i.e., hiding and binding) can be satisfied unconditionally or relatively to a computational assumption. In our context (i.e., where only two parties are involved) this immediately implies that one can not hope to build a commitment scheme where both the hiding and the binding properties hold unconditionally. Unconditionally binding commitment schemes have been constructed under the sole assumption that one-way functions exist [57] and in that construction an initial message of the receiver is required. It is known how to construct non-interactive unconditionally binding commitment schemes by using any one-to-one one-way function [6]. In [4], the authors show how to construct non-interactive commitments based on one-way functions and derandomization techniques. Constant-round unconditionally hiding commitment schemes have been constructed under the assumption that collections of claw-free functions [46] or collision resistant hash functions [51] exist. Recently in [50] it is shown how to construct unconditionally hiding commitment schemes from any regular one-way function, but unfortunately, these schemes are not constant round.

Since commitment schemes are very useful primitives they are often used as building blocks to construct larger protocols. In this sense it is often the case that the two basic requirements described above turn out to be insufficient. For this reason commitment schemes with additional properties have been proposed. Here we highlight some of these constructions, others, more directly related to our results, will be discussed in the next section.

A *trapdoor commitment scheme* (sometimes also called *chameleon* commitment), is a commitment scheme with an associated pair of public and private keys (the latter also called the *trapdoor*). Knowledge of the trapdoor allows the sender to open the commitment in more than one way (this is often referred to as the *equivocality* property). On the other hand, without knowledge of the trapdoor, equivocality remains computationally infeasible. When the commitments computed by means of a trapdoor are distributed exactly as real commitments then the trapdoor commitment scheme is unconditionally hiding. Trapdoor commitments have been shown to exist under the assumption that one-way functions exist [38].

A commitment scheme is said to be *non-malleable* if – very informally – given a commitment c on some message m, knowledge of c does not help another party in constructing a new commitment c' of a message m' related to m. Such a property is referred to as non-malleability with respect to commitment and is studied for unconditionally binding commitment schemes. Another different property is non-malleability with respect to opening. In this case, c' is a commitment that once c is opened to m, can be opened to a value m' related to m. Non-malleable commitments with respect to commitment have been defined in [34], and the first constant-round construction in the plain model has been given in [2] for the stand-alone case (i.e., the protocol is interactive and has to be executed in isolation, concurrency is not allowed). Non-interactive non-malleable commitments with respect to opening have been originally constructed in [31,40,32]. Such constructions have been recently improved in [26] where once the commitment parameters have been established (by a trusted third party), it is possible to compute any polynomial number of non-malleable commitments.

An *extractable* commitment (also known as commitment scheme with *extractability*) is a commitment scheme where we allow the existence of a secret key whose knowledge permits us to *extract* the message stored in the commitment. At the same time, without knowledge of the secret key, the message remains (computationally) hidden.

Finally a *universally composable commitment* is a commitment scheme with the very useful property that – informally – even if one concurrently composes it with any other protocol, the security of the commitment scheme is preserved. Universal composability is a very strong notion, which, for the case of commitment schemes, seems to require concurrent non-malleability and extractability.

The first construction of a universally composable commitment scheme has been presented in [14] and it has been later improved in [16,28] and in [26].

1.1. Other notions of commitments

Simulation-sound trapdoor commitments. Garay et al. [42] introduced the notion of *simulation-sound trapdoor commitments* (SSTCs, for short) which was later relaxed by MacKenzie and Yang in [55]. In a nutshell an SSTC scheme is a trapdoor commitment scheme that associates a label called a tag to each commitment. An adversary of an SSTC scheme can not equivocate a commitment with a certain tag, even after seeing a polynomial number of equivocations for commitments with different tags.

In [42] SSTCs are used to construct efficient universally composable zero-knowledge arguments that remain secure even when facing adaptive adversaries (i.e., adversaries which are allowed to adaptively corrupt parties involved in the protocol). In [55] is presented a simpler – and slightly weaker – definition of SSTC and the authors prove that the resulting primitive is actually equivalent to secure signatures [49]. The notion of SSTC introduced in [55], being weaker with respect to the one in [42] allows one to construct more efficient number-theoretic implementations. Interestingly, this weaker notion still remains sufficient to construct many applications (such as simulation-sound zero-knowledge argument systems).

Multi-trapdoor commitments. The notion of a multi-trapdoor commitment scheme was introduced by Gennaro [44]. A multi-trapdoor commitment scheme is a family of trapdoor commitments such that each scheme in the family is a trapdoor commitment scheme. The main feature of multi-trapdoor commitments is that they admit a *master* trapdoor whose knowledge allows one to equivocate for any commitment scheme in the family. Furthermore every commitment in the family admits also its own *local* trapdoor. However knowledge of the local trapdoor for a given scheme *does not* allow one to equivocate on another scheme of the family (unless of course the master trapdoor is available). In [44], Gennaro presents two very efficient constructions (based on the strong RSA and on the strong Diffie–Hellman assumptions respectively) as well as several applications. The main application of multi-trapdoor commitments is the construction of left-concurrent non-malleable arguments of knowledge. Here the *left-concurrent non-malleable* setting is a model in which the adversary can play in a polynomial number of sessions the role of verifier while in one session he plays the role of prover. As already pointed out in [55], a left-concurrent non-malleable argument of knowledge is actually an unbounded simulation-sound zero-knowledge argument system. We stress that the multi-trapdoor commitments presented in [44] allows for simulation-sound zero-knowledge argument systems that are more efficient than the ones presented in [55]. The definition of multi-trapdoor commitments given in [44] requires unconditional hiding. This property is obviously enjoyed by both implementations of multi-trapdoor commitments given in [44]. However, in the applications given in [44], unconditional hiding is not used. Since even our applications will only use computational hiding, we will focus in both the definitions and the constructions on multi-trapdoor commitments that are computationally hiding.

We finally notice that in [44] multi-trapdoor commitments were constructed under number-theoretic assumptions only.

Mixed commitments. In [28] Damgård and Nielsen introduced the notion of mixed commitments. Informally, a mixed commitment scheme can be either a trapdoor commitment scheme or an extractable commitment scheme, the exact nature depends on the distribution according to which the public key is generated. If the public key is composed by some specific keys (referred to as E-keys) then the scheme is an unconditionally hiding trapdoor commitment scheme. If instead the public key is composed by some other specific keys (referred to as X-keys) then the scheme is an unconditionally binding and extractable commitment scheme. Of course no key can be both an E-key and an X-key. A crucial property of mixed commitment schemes is that no polynomially bounded adversary (not having access to secret keys or trapdoors corresponding to the public key) should be able to distinguish E-keys from X-keys.

In [28] some efficient implementations have been derived from Damgård–Jurik's [27] variant of the Paillier [60] cryptosystem and from the Okamoto–Uchiyama [59] cryptosystem. Moreover the authors showed how to use mixed commitments to build universally composable commitments with the constraint that the size of the public key depends on the number of players. More recently Damgård and Groth [26] improved this construction by proposing a universally composable commitment scheme based on the strong RSA assumption, that can work with a reference string whose size does not depend on the number of players involved in the protocol. This result is achieved by combining a non-malleable commitment scheme with mixed commitments.

1.2. Our contributions

In this paper we introduce the notion of a hybrid trapdoor commitment scheme. Informally a hybrid trapdoor commitment scheme is a general commitment primitive with commitment parameter generation algorithms `HGen` and `HTGen`. If the commitment parameters are obtained as the output of `HGen` then the resulting scheme is an unconditionally binding commitment scheme, while if the parameters are generated by `HTGen` the produced scheme is actually a trapdoor commitment scheme. Moreover, as for mixed commitments, no polynomially bounded adversary, taking as input only the (public) commitment parameters, should be able to tell the difference between keys generated from `HGen` and keys produced by `HTGen`.

Notice that the notion of hybrid trapdoor commitment may look very similar to that of mixed commitment. There is a crucial difference however. Depending on the way the parameters are generated a mixed commitment can be either an extractable commitment or a trapdoor commitment. In our case, on the other hand, we require only that the commitment is either unconditionally binding or a trapdoor commitment scheme.

As mentioned before, mixed commitments have been introduced to construct universally composable commitments and indeed Damgård and Nielsen proved that it is possible to construct a universally composable commitment from a mixed commitment where the number of E-keys over the total number of keys is negligible and that the number of X-keys over the total number of keys is almost 1. Interestingly, a recent result by Damgård and Groth [26] shows that universally composable commitments imply key exchange and, when implemented in the shared random string model, they imply oblivious transfer. Therefore it seems unlikely that universally composable commitments can be implemented from one-way functions only.

In this paper, on the other hand, we show that hybrid trapdoor commitments can be constructed from any one-way function.

To improve on efficiency we then turn our attention to specific number-theoretic constructions and we propose three different implementations. The first one is very efficient and relies on the Decisional Diffie–Hellman assumption. The remaining two are based on Paillier's [60] Decisional Composite Residuosity Assumption (DCRA for brevity) and build on previous constructions by Catalano et al. [18] and Bresson et al. [12]. Interestingly the DCRA-based solutions enjoy the extractability property and turn out to be mixed commitment schemes. In particular our first DCRA-based solution is very efficient and it is actually slightly more efficient than the two implementations proposed by Damgård and Nielsen [28]. Our construction of hybrid trapdoor commitments that are based on the Diffie–Hellman assumption is much more efficient than the known constructions of mixed commitments.

Stronger extensions. We study some stronger extensions of hybrid trapdoor commitments. In particular we show how to build hybrid simulation-sound trapdoor commitments and hybrid multi-trapdoor commitments from the sole assumption that one-way functions exist. Moreover, we present some practical implementations based on number-theoretic assumptions.

In this paper we show that multi-trapdoor commitment schemes are actually equivalent to digital signatures which are secure with respect to *generic* chosen message attack. Informally in a generic chosen message attack the adversary can obtain signatures only on a list of messages chosen *before* the public key of the signer is published. This is clearly a weaker notion with respect to the standard one where the adversary is allowed to choose the messages adaptively. Since SSTCs are actually equivalent to standard secure signatures, from a practical point of view, our result further clarifies why the known (practical) implementations of multi-trapdoor commitments are more efficient than the corresponding implementations of SSTC.

Techniques. The construction of a hybrid trapdoor commitment scheme consists in producing a scheme that is a trapdoor commitment scheme for some commitment parameters while it is an unconditionally binding commitment scheme for other commitment parameters. We stress that the two distributions of the commitment parameters are indistinguishable. Once we achieve this result, we combine the basic hybrid trapdoor commitment scheme with a tag-based simulation-sound trapdoor (resp., multi-trapdoor) commitment scheme. This is based on a parallel execution of the two commitment schemes, thus obtaining a hybrid tag-based simulation-sound trapdoor (resp., multi-trapdoor) commitment scheme. We finally stress that the results of [26] actually achieve hybrid trapdoor commitments based on one-way functions only, even though this is not explicitly formalized and claimed in their work.

Applications. The main contribution of this paper is to show how to use the different variants of hybrid trapdoor commitments to achieve the following results.

1. Using hybrid trapdoor commitments we show how to construct three-round concurrent zero-knowledge proof systems, in the shared random string model, for all $\mathcal{NP}$ languages. We give a construction based on the existence of any one-way function and an efficient construction that is based on the DDH assumption. These results improve the computational soundness achieved in a previous result by Damgård [29] in the sense that ours are actually zero-knowledge proofs rather than zero-knowledge arguments.
2. Using either hybrid SSTC or hybrid multi-trapdoor commitments we show how to construct an unbounded simulation-sound zero-knowledge proof system in the common reference string model. This improves the recent results of [55,44] where similar results were presented for unbounded simulation-sound zero-knowledge arguments (rather than proofs).

Comparison with other previous results. One of the techniques that we use in the constructions for hybrid trapdoor commitments is that of having two different algorithms for generating the reference string. The former outputs an "honest" reference string (i.e., a reference string that does not contain any trapdoor), while the latter outputs a fake reference string along with the corresponding trapdoor. The distributions of the two reference strings are indistinguishable. This technique was first introduced in [8] for achieving non-interactive zero-knowledge proofs. The same technique has been used in [30] for achieving non-malleable non-interactive zero-knowledge proofs under the assumption that trapdoor permutations exist. Both proof systems crucially use an $\mathcal{NP}$-reduction by following the FLS-paradigm [37]. In this paper we basically show that this same technique can be used to construct (1) three-round simulation-sound zero knowledge proofs under the sole assumption that one-way functions exist; (2) very efficient three-round simulation-sound zero knowledge proofs under standard number-theoretic assumptions.

The first non-interactive non-malleable commitment based on one-way functions was previously presented in [31] but there the adversary can only see one commitment computed on a given reference string. In [16] it is shown that the trapdoor commitment scheme based on Hamiltonian graphs of [38] allows the adversary to see any polynomial number of commitments before computing the mauled one. We stress that our contribution shows that hybrid trapdoor commitments are commitments secure even against computationally unbounded adversarial committers and that admit very efficient constructions based on number-theoretic assumptions and constructions based on the sole assumption that one-way functions exist.

Proofs versus arguments. A proof system has the following property: any adversarial prover (regardless of his computing power) has negligible probability of making an honest verifier accept a false statement. This strong notion of soundness differs from the corresponding notion of soundness of an argument system, where security for honest verifiers holds only against polynomial-time adversarial provers. The notions of argument and proof differ dramatically when zero knowledge is considered. For example, while it is known that any $\mathcal{NP}$ language has a perfect zero-knowledge argument [11], if an $\mathcal{NP}$-complete language has a perfect zero-knowledge proof then the polynomial hierarchy collapses to its second level [41,10]. With respect to constant-round zero knowledge, the current state of knowledge gives us a constant-round (computational) zero-knowledge proof for $\mathcal{NP}$ under the assumption that collections of claw-free functions or collision-resistant hash functions exist [46,51], while constant-round zero-knowledge arguments for $\mathcal{NP}$ are known to exist under the assumption that one-way functions exist [5].

As discussed above, in this paper we show that our new notion of commitment scheme can be used to obtain some strong variants of zero-knowledge proof systems.

2. Definitions

We now give some basic definitions that we will use in this paper. We use the notation $\{\beta_1, \ldots, \beta_k : \alpha\}$ to specify the probability distribution of α after the sequential executions of events $\beta_1, \ldots, \beta_k$. In general, we assume that an algorithm $\mathcal{A}$ has access to some random (auxiliary) input even though this is not explicitly specified. Moreover, if A is a probabilistic algorithm we denote by $A(x)$ the random variable describing the output of A on input x. We say that a function ν is *negligible* iff for all constants c there exists n_0 such that for all $n > n_0$ it holds that $0 \le \nu(n) < 1/n^c$.

A binary relation R is polynomially bounded if it is decidable in polynomial time and there exists a polynomial p such that for all pairs $(x, w) \in R$ it holds that $|w| \le p(|x|)$. We denote by $L_R = \{x | \exists y : (x, w) \in R\}$ the $\mathcal{NP}$-language associated with R.

For an $\mathcal{NP}$-language L we denote by R_L the *witness relation* associated with L defined as $x \in L \Leftrightarrow \exists w : (x, w) \in R_L$.

Definition 2.1. A function $f : \{0, 1\}^\star \to \{0, 1\}^\star$ is called *one way* if the following conditions hold:

1. there exists a deterministic polynomial-time algorithm $\mathcal{A}$ such that on input x, $\mathcal{A}$ outputs $f(x)$;
2. for every non-uniform probabilistic polynomial-time algorithm $\mathcal{A}'$ there exists a negligible function ν such that for all sufficiently large k, it holds that

$$Pr\left(x \leftarrow \{0, 1\}^k; \mathcal{A}'(f(x)) \in f^{-1}(f(x))\right) < \nu(k).$$

We now give definitions for several notions of commitment schemes. For readability we will use "for all m" to mean any possible message m of length polynomial in the security parameter. We start with the standard notion of commitment scheme with its two main variants (i.e., unconditionally binding and unconditionally hiding). Note that all definitions will use a commitment generator function that outputs the commitment parameters. Therefore, such commitments have a straightforward implementation in the common reference string model where a trusted third party generates a reference string that is later received as common input by all parties. In some cases the commitment parameters generated by the commitment generator function will be strings with uniform distribution; in such cases the corresponding commitments can be implemented in the shared random string model which is a set-up assumption weaker than the common reference string model.

Definition 2.2. $(\mathtt{Gen}, \mathtt{Com}, \mathtt{Ver})$ is a *commitment scheme* (CS, for short) if:

- **efficiency:** $\mathtt{Gen}$, $\mathtt{Com}$ and $\mathtt{Ver}$ are polynomial-time algorithms;
- **completeness:** for all m it holds that

$$\mathtt{Prob}\left(\mathtt{crs} \leftarrow \mathtt{Gen}(1^k); (\mathtt{com}, \mathtt{dec}) \leftarrow \mathtt{Com}(\mathtt{crs}, m) : \mathtt{Ver}(\mathtt{crs}, \mathtt{com}, \mathtt{dec}, m) = 1\right) = 1;$$

- **binding:** for any polynomial-time algorithm $\mathtt{sender}$ there is a negligible function ν such that for all sufficiently large k it holds that

$$\mathtt{Prob}\Big(\mathtt{crs} \leftarrow \mathtt{Gen}(1^k); (\mathtt{com}, m_0, m_1, \mathtt{dec}_0, \mathtt{dec}_1) \leftarrow \mathtt{sender}(\mathtt{crs}) :$$
$$m_0 \neq m_1 \text{ and } \mathtt{Ver}(\mathtt{crs}, \mathtt{com}, \mathtt{dec}_0, m_0) = \mathtt{Ver}(\mathtt{crs}, \mathtt{com}, \mathtt{dec}_1, m_1) = 1\Big) \le \nu(k);$$

- **hiding:** for any adversary $\mathtt{receiver}$ there is a negligible function ν such that for all m_0, m_1 where $|m_0| = |m_1|$ and all sufficiently large k it holds that

$$\mathtt{Prob}\left(\mathtt{crs} \leftarrow \mathtt{Gen}(1^k); b \leftarrow \{0, 1\}; (\mathtt{com}, \mathtt{dec}) \leftarrow \mathtt{Com}(\mathtt{crs}, m_b) : b \leftarrow \mathtt{receiver}(\mathtt{com})\right) < \frac{1}{2} + \nu(k).$$

If the binding property holds with respect to a computationally unbounded algorithm $\mathtt{sender}$, the commitment scheme is said to be *unconditionally binding*; if instead, the hiding property holds with respect to a computationally unbounded algorithm $\mathtt{receiver}$, the commitment scheme is said to be *unconditionally hiding*.

We now give the definition of a trapdoor commitment scheme. In particular we strengthen the computational indistinguishability of the computed commitments so that it holds even if the distinguisher takes as input the trapdoor information. This allows one to use the same commitment parameters for any polynomial number of commitments (and actually all our results hold in this stronger setting).

Definition 2.3. $(\mathtt{Gen}, \mathtt{Com}, \mathtt{TCom}, \mathtt{TDec}, \mathtt{Ver})$ is a *trapdoor* commitment scheme (TCS, for short) if $\mathtt{Gen}(1^k)$ outputs a pair $(\mathtt{crs}, \mathtt{aux})$, $\mathtt{Gen}_{\mathtt{crs}}$ is the related algorithm that restricts the output of $\mathtt{Gen}$ to the first element $\mathtt{crs}$, $(\mathtt{Gen}_{\mathtt{crs}}, \mathtt{Com}, \mathtt{Ver})$ is a commitment scheme and $\mathtt{TCom}$ and $\mathtt{TDec}$ are polynomial-time algorithms such that:
– trapdoorness: for all m the probability distributions:

$$\{(\mathtt{com}, \mathtt{dec}) \leftarrow \mathtt{Com}(\mathtt{crs}, m) : (\mathtt{crs}, \mathtt{com}, \mathtt{dec}, m)\} \quad \text{and}$$
$$\{(\mathtt{com}', \mathtt{aux}_{\mathtt{com}'}) \leftarrow \mathtt{TCom}(\mathtt{crs}, \mathtt{aux}); \mathtt{dec}' \leftarrow \mathtt{TDec}(\mathtt{aux}_{\mathtt{com}'}, m) : (\mathtt{crs}, \mathtt{com}', \mathtt{dec}', m)\}$$

are computationally indistinguishable to any adversary that knows $\mathtt{aux}$.

In the next theorem we show that the existence of one-way functions is sufficient for constructing trapdoor commitment schemes under the above definition.

Theorem 2.4. *Under the assumption that one-way functions exist, there exists a trapdoor commitment scheme.*

Proof. Let f be a one-way function. Consider the commitment scheme proposed by Naor (see Appendix B for a discussion about Naor's commitment scheme), an instance $(\mathtt{Gen}, \mathtt{Com}, \mathtt{Ver})$ of such a commitment scheme can be constructed by using f. We now construct the following trapdoor commitment scheme $(\mathtt{Gen}, \mathtt{Com}, \mathtt{TCom}, \mathtt{TDec}, \mathtt{Ver})$.

Algorithm $\mathtt{Gen}$ produces a random string $\mathtt{crs}$ that can be interpreted as $\mathtt{crs} = \mathtt{setup} \circ \mathtt{com}_1$ (where $\circ$ denotes concatenation). Here $\mathtt{setup}$ denotes the first message of Naor's commitment scheme and $\mathtt{com}_1$ is computed as a commitment of the string 1^k. The second output of $\mathtt{Gen}$ is the decommitment $\mathtt{aux}$ corresponding to the commitment $\mathtt{com}_1$.

Algorithm $\mathtt{Com}$ on input $\mathtt{crs}$ and a message m runs the simulator S of Blum's Σ-protocol (again see Appendix B for more details about Σ protocols and an implementation of Blum's Σ-protocol based on the existence of one-way functions only) for proving that $(\mathtt{setup}, \mathtt{com}_1)$ is a commitment of string 1^k. In particular S runs on input $(\mathtt{setup}, m)$ to obtain an accepting transcript $(\mathtt{setup}, a, m, z)$. The output of $\mathtt{Com}$ is the pair $(\mathtt{com} = a, \mathtt{dec} = z)$.

Algorithm $\mathtt{Ver}$ simply verifies that $(\mathtt{setup}, \mathtt{com}, m, z)$ is an accepting transcript for the statement "$(\mathtt{setup}, \mathtt{com}_1)$ is a commitment of the string 1^k".

Algorithm $\mathtt{TCom}$, on input $(\mathtt{crs}, \mathtt{aux})$ computes a valid first message a for Blum's Σ-protocol for proving that $(\mathtt{setup}, \mathtt{com}_1)$ is a commitment of the string 1^k. $\mathtt{TCom}$ outputs a and $\mathtt{aux}_a = \mathtt{crs} \circ \mathtt{aux} \circ \mathtt{aux}'$ where $\mathtt{aux}'$ is the auxiliary information generated by the prover algorithm of Blum's Σ-protocol when the first message a is computed.

Algorithm $\mathtt{TDec}$ on input $\mathtt{aux}_a$ and a string m, runs the prover algorithm of Blum's Σ-protocol on input the transcript $(\mathtt{setup}, a, m)$ to obtain the last message z such that $(\mathtt{setup}, a, m, z)$ is an accepting transcript. $\mathtt{TDec}$ can run the prover algorithm since he can pick from $\mathtt{aux}_a$ both the decommitment $\mathtt{aux}$ corresponding to $\mathtt{com}_1$ (i.e., the witness for the statement) and the auxiliary information $\mathtt{aux}'$ corresponding to a. The output of $\mathtt{TDec}$ is $\mathtt{dec} = z$.

$(\mathtt{Gen}, \mathtt{Com}, \mathtt{TCom}, \mathtt{TDec}, \mathtt{Ver})$ **is a trapdoor CS.** First notice that the scheme is (computationally) binding. Indeed if there exists an algorithm $\mathtt{sender}$ that manages to open a commitment in two different ways, by running the extractor algorithm of Blum's Σ-protocol one can obtain a cycle in the graph and then a witness for the statement "$\mathtt{com}$ *is a commitment of the string* 1^k". Thus, the existence of such a $\mathtt{sender}$ would lead to breaking the hiding property of Naor's commitment scheme.

Next, knowing a cycle in the graph (or, equivalently, knowing a witness for the given statement) one can efficiently compute a valid decommitment $\mathtt{dec} = z$ for any commitment a and any string m. Such a triple (a, m, z) obtained using algorithms $\mathtt{TCom}$ and $\mathtt{TDec}$ is computationally indistinguishable from a triple (a', m, z') computed by using $\mathtt{Com}$ even to a distinguisher that knows the trapdoor, i.e., the cycle in the graph. Indeed, by the honest-verifier zero-knowledge property of Blum's Σ-protocol the only difference is in the pair (a, a'). This is because, the first message of

Blum's Σ-protocol contains commitments to graphs. In fact, the message a computed by the honest prover algorithm corresponds to commitments of randomly permuted graphs, some of them are opened to show the original graphs, other are opened to show Hamiltonian cycles in the randomly permuted graphs, the remaining commitments are never opened. The message a' computed by the simulator corresponds in part to commitments of permuted graphs that are opened to show the original graphs, in part to commitments of randomly chosen Hamiltonian cycles that are opened to show the Hamiltonian cycles and in part by commitments that are never opened. Since in both a and a' the commitments are computed using Naor's commitment scheme, the only difference is in the commitments that are never opened. Therefore, any distinguisher (even on input the trapdoor) would allow one to break (with a polynomially related probability) the hiding property of the commitment scheme used in the first round of Blum's Σ-protocol (which in our case, is Naor's commitment scheme).

We now discuss the case in which the distinguisher is allowed to query any polynomial number l of commitments and decommitments and it is challenged to distinguish the case in which commitments are honestly computed and opened from the *opposite* case in which commitments are computed and opened by means of the trapdoor. Using a standard hybrid argument we show that if A is a (polynomially bounded) adversary succeeding in the above task, then one can build an adversary B out of it that violates the trapdoorness property of the underlying commitment scheme. Such a B receives on input the public parameters (together with the trapdoor) corresponding to some trapdoor commitment scheme. We remark that B is allowed access to a challenging oracle O that, first outputs crs and aux and then, when asked on input m outputs a challenge (com, dec) that is sampled using either Com or TCom, TDec according to some random and secret bit b (wlog we assume that $b = 0$ means that the challenge is produced using TCom, TDec).

B works as follows: first it obtains crs and aux from O, then it chooses an index j uniformly and at random in the range $\{1, \ldots, l\}$. Next, whenever A on input crs asks its i-th query, for some message m_i, B answers in the following way:

if $i < j$ it computes a commitment and a decommitment of m_i using Com, and hands the resulting values to A;
if $i > j$ it computes a commitment and a decommitment of m_i using TCom, TDec, and hands the resulting values to A;
if $i = j$ it queries O on input m_j and hands the obtained challenge (com, dec) to A.

At some point A may output a bit d. In particular we can assume that A outputs 1 if it thinks that all the commitments were created using Com and 0 if it thinks that all of the commitments are fake ones. B simply outputs the same d as its own guess.

Let us denote by G_k the scenario where the first k queries are answered using Com while the remaining $l-k$ queries are answered using TCom, TDec. We denote by P_k the probability that A wins when running in scenario G_k. Clearly, one has that the advantage μ_A of A in succeeding in its guess is $\mu_A = |P_0 - P_l|$. The probability μ_B that B correctly guesses b, on the other hand, is given by

$$\mu_B = Pr[B \to b' | b = b'] = \frac{1}{l} \left| \sum_{k=1}^{l} P_k - \sum_{k=1}^{l} P_{k-1} \right| = \frac{1}{l} \mu_A$$

thus concluding our proof. □

The definitions of commitment and trapdoor commitment schemes presented above can be extended by adding one more input to algorithms Com, Ver and TCom that is, a label referred to as "tag". In this case, algorithm Ver has an additional constraint, it outputs 1 only if the same tag has been used as input by algorithms Com or TCom. In particular, we will use such a tag-based definition of commitment when we consider the notion of a simulation-sound trapdoor commitment.

Now we define multi-trapdoor commitment schemes.

Definition 2.5. (Gen, Sel, Tkg, Com, TCom, TDec, Ver) is a *multi-trapdoor* commitment scheme (MTCS, for short) if:

- **efficiency:** Gen, Sel, Tkg, Com, TCom, TDec and Ver are polynomial-time algorithms;
- **completeness:** for all m it holds that

$$\texttt{Prob}((\texttt{PK}, \texttt{TK}) \leftarrow \texttt{Gen}(1^k); \texttt{pk} \leftarrow \texttt{Sel}(\texttt{PK}); (\texttt{com}, \texttt{dec}) \leftarrow \texttt{Com}(\texttt{pk}, m) : \texttt{Ver}(\texttt{pk}, \texttt{com}, \texttt{dec}, m) = 1) = 1;$$

- **multi-trapdoorness (implies hiding):** for all $(\mathtt{PK}, \mathtt{TK})$ generated with non-zero probability by $\mathtt{Gen}(1^k)$, for all $\mathtt{pk}$ where $\mathtt{pk} \leftarrow \mathtt{Sel}(\mathtt{PK})$, for all m the following probability distributions

$$\{(\mathtt{com}, \mathtt{dec}) \leftarrow \mathtt{Com}(\mathtt{pk}, m) : (\mathtt{pk}, \mathtt{com}, \mathtt{dec}, m)\}$$
$$\{\mathtt{tk} \leftarrow \mathtt{Tkg}(\mathtt{TK}, \mathtt{pk}); (\mathtt{com}', \mathtt{aux}) \leftarrow \mathtt{TCom}(\mathtt{tk}); \mathtt{dec}' \leftarrow \mathtt{TDec}(\mathtt{aux}, m) : (\mathtt{pk}, \mathtt{com}', \mathtt{dec}', m)\}$$

  are computationally indistinguishable to any adversary knowing $\mathtt{TK}$.
- **binding:** there is a negligible function ν such that for all $m_0 \neq m_1$ and any pair of polynomial-time algorithms $(\mathtt{sender}_0, \mathtt{sender}_1)$ it holds that

$$\begin{aligned}&\mathtt{Prob}((\widetilde{\mathtt{pk}} = \{\mathtt{pk}_1, \ldots, \mathtt{pk}_k\}, \mathtt{aux}) \leftarrow \mathtt{sender}_0(1^k); (\mathtt{PK}, \mathtt{TK}) \leftarrow \mathtt{Gen}(1^k);\\ &(\mathtt{pk}, \mathtt{com}, m_0, m_1, \mathtt{dec}_0, \mathtt{dec}_1) \leftarrow \mathtt{sender}_1^{\mathcal{O}_{PK}}(\mathtt{PK}, \mathtt{aux}) : \mathtt{pk} \notin \widetilde{\mathtt{pk}} \wedge\\ &\mathtt{Ver}(\mathtt{pk}, \mathtt{com}, \mathtt{dec}_0, m_0) = \mathtt{Ver}(\mathtt{pk}, \mathtt{com}, \mathtt{dec}_1, m_1) = 1) \leq \nu(k)\end{aligned}$$

  where $\mathcal{O}_{PK}$ operates as follows:
  - On input $(\mathtt{pk} \in \widetilde{\mathtt{pk}})$: compute $(\widetilde{\mathtt{com}}, \xi) \leftarrow \mathtt{TCom}(\mathtt{tk})$, where $\mathtt{tk} = \mathtt{Tkg}(\mathtt{TK}, \mathtt{pk})$, store $(\widetilde{\mathtt{com}}, \mathtt{pk}, \xi)$ and output $\widetilde{\mathtt{com}}$.
  - On input $(\widetilde{\mathtt{com}}, m)$: if for some $\mathtt{pk}$ and some ξ, a tuple $(\widetilde{\mathtt{com}}, \mathtt{pk}, \xi)$ is stored, compute $\widetilde{\mathtt{dec}} \leftarrow \mathtt{TDec}(\xi, m)$ and output $\widetilde{\mathtt{dec}}$.

  Note that an efficient algorithm $\mathcal{O}_{PK}$ can be implemented using the master trapdoor.

Remark 2.6. Notice that our definition of multi-trapdoor commitments is slightly different with respect to the one given by Gennaro in [43]. First of all, we allow the possibility of using a trapdoor even during the commitment phase (as done in the definition of SSTC given in [55]). One can easily verify that the construction given in [44] remains sound with respect to the new definition as well.

Our definition also assumes that the hiding property holds only with respect to polynomially bounded adversaries (while in the original definition the scheme is required to be unconditionally hiding). This (more general) definition allows us to construct multi-trapdoor commitments from any one-way function (see Theorem 3.3). On the other hand it seems unlikely that unconditionally hiding trapdoor commitment schemes can be constructed from one-way functions (and even from trapdoor permutations [39]). Moreover, we point out here, that for all our applications it is sufficient to consider multi-trapdoor commitment schemes which are hiding only in a computational sense.

Finally we assume wlog that a local public key also contains the master public key.

Now we define simulation-sound trapdoor commitment schemes which are based on the notion of tag-based commitment schemes briefly discussed above.

Definition 2.7. $(\mathtt{Gen}, \mathtt{Com}, \mathtt{TCom}, \mathtt{TDec}, \mathtt{Ver})$ is a tag-based *Simulation-Sound* trapdoor commitment scheme (SSTCS, for short) if:

- $(\mathtt{Gen}, \mathtt{Com}, \mathtt{TCom}, \mathtt{TDec}, \mathtt{Ver})$ is a tag-based trapdoor commitment scheme;
- **Simulation-Sound Binding:** for any polynomial-time algorithm $\mathtt{sender}$ there is a negligible function ν such that

$$\begin{aligned}&\mathtt{Prob}\Big((\mathtt{crs}, \mathtt{aux}) \leftarrow \mathtt{Gen}(1^k); (\mathtt{com}, \mathtt{tag}, m_0, m_1, \mathtt{dec}_0, \mathtt{dec}_1) \leftarrow \mathtt{sender}^{\mathcal{O}_{crs}}(\mathtt{crs}) : (m_0 \neq m_1) \wedge\\ &\mathtt{Ver}(\mathtt{crs}, \mathtt{com}, \mathtt{dec}_0, m_0, \mathtt{tag}) = \mathtt{Ver}(\mathtt{crs}, \mathtt{com}, \mathtt{dec}_1, m_1, \mathtt{tag}) = 1 \wedge \mathtt{tag} \notin Q\Big) \leq \nu(k),\end{aligned}$$

  where $\mathcal{O}_{crs}$ operates as follows:
  - On input $(\mathtt{tag})$: compute $(\widetilde{\mathtt{com}}, \xi) \leftarrow \mathtt{TCom}(\mathtt{aux}, \mathtt{tag})$, add $\mathtt{tag}$ to Q, store $(\widetilde{\mathtt{com}}, \mathtt{tag}, \xi)$, and output $\widetilde{\mathtt{com}}$.
  - On input $(\widetilde{\mathtt{com}}, m)$: if for some $\mathtt{tag}$ and some $\widetilde{\mathtt{com}}$, a tuple $(\widetilde{\mathtt{com}}, \mathtt{tag}, \xi)$ is stored, compute $\widetilde{\mathtt{dec}} \leftarrow \mathtt{TDec}(\xi, m)$ and output $\widetilde{\mathtt{dec}}$.

  Again, an efficient algorithm $\mathcal{O}_{crs}$ can be implemented by using the auxiliary information generated along with $\mathtt{crs}$.

A brief review of the main tools used in our constructions is given in Appendix B.

2.1. Hybrid trapdoor commitments: Definitions

Now we are ready to introduce the notion of a hybrid trapdoor commitment. As sketched in the introduction, such a notion considers the existence of two commitment generation functions whose public outputs are computationally indistinguishable. Still the properties of the two resulting commitment schemes are very different. We start with the basic notion of a hybrid trapdoor commitment scheme.

Definition 2.8. $(\mathtt{HGen}, \mathtt{HTGen}, \mathtt{HCom}, \mathtt{HTCom}, \mathtt{HTDec}, \mathtt{HVer})$ is a hybrid *trapdoor* commitment scheme (HTCS, for short) if:

- **binding:** $(\mathtt{HGen}, \mathtt{HCom}, \mathtt{HVer})$ is an unconditionally binding commitment scheme;
- **trapdoorness:** $(\mathtt{HTGen}, \mathtt{HCom}, \mathtt{HTCom}, \mathtt{HTDec}, \mathtt{HVer})$ is a *trapdoor* commitment scheme.
- **hybridness:** let $\mathtt{HTGen}'$ be an algorithm that restricts the output $(\mathtt{crs}, \mathtt{aux})$ of $\mathtt{HTGen}(1^k)$ to $\mathtt{crs}$, then the following probability distributions are computationally indistinguishable: $\{\mathtt{crs}_0 \leftarrow \mathtt{HGen}(1^k) : \mathtt{crs}_0\}$ and $\{\mathtt{crs}_1 \leftarrow \mathtt{HTGen}'(1^k) : \mathtt{crs}_1\}$.

The notion given above can be extended to be a tag-based commitment scheme (as for the case of standard trapdoor commitment schemes).

We now define the notions of hybrid multi-trapdoor and hybrid simulation-sound trapdoor commitment schemes. We stress that for the latter we focus on tag-based commitments (obtained by adding a label – the tag – as input to the algorithms that compute and verify commitments).

Intuitively, since multi-trapdoor and simulation-sound trapdoor commitment schemes define families of trapdoor commitment schemes, for the *hybrid* variant of such primitives, we require that each scheme in the family is a *hybrid* trapdoor commitment scheme.

Definition 2.9. $(\mathtt{HGen}, \mathtt{HTGen}, \mathtt{HSel}, \mathtt{HTkg}, \mathtt{HCom}, \mathtt{HTCom}, \mathtt{HTDec}, \mathtt{HVer})$ is a hybrid *multi-trapdoor* commitment scheme (HMTCS, for short) if:

- **multi-trapdoorness:** $(\mathtt{HTGen}, \mathtt{HSel}, \mathtt{HTkg}, \mathtt{HCom}, \mathtt{HTCom}, \mathtt{HTDec}, \mathtt{HVer})$ is a multi-trapdoor commitment scheme;
- **hybridness:** let $\mathtt{HGen}'(1^k)$ be an algorithm that runs $\mathtt{HGen}(1^k)$ obtaining $(\mathtt{crs}, \mathtt{aux})$, runs $\mathtt{HSel}(\mathtt{crs})$ obtaining $\mathtt{pk}$ and then outputs $\mathtt{pk}$; moreover let $\mathtt{HTGen}'$ be an algorithm that runs $\mathtt{HTGen}(1^k)$ obtaining $(\mathtt{crs}', \mathtt{aux}')$, runs $\mathtt{HSel}(\mathtt{crs}')$ obtaining $\mathtt{pk}$, runs $\mathtt{HTkg}(\mathtt{aux}', \mathtt{pk})$ obtaining $\mathtt{tk}$ and then outputs $(\mathtt{pk}, \mathtt{tk})$; then it holds that $(\mathtt{HGen}', \mathtt{HTGen}', \mathtt{HCom}, \mathtt{HTCom}, \mathtt{HTDec}, \mathtt{HVer})$ is a hybrid trapdoor commitment scheme.

Definition 2.10. $(\mathtt{HGen}, \mathtt{HTGen}, \mathtt{HCom}, \mathtt{HTCom}, \mathtt{HTDec}, \mathtt{HVer})$ is a hybrid *simulation-sound* trapdoor commitment scheme (HSSTCS, for short) if:

- **simulation soundness:** $(\mathtt{HTGen}, \mathtt{HCom}, \mathtt{HTCom}, \mathtt{HTDec}, \mathtt{HVer})$ is a simulation-sound trapdoor commitment scheme;
- **hybridness:** $(\mathtt{HGen}, \mathtt{HTGen}, \mathtt{HCom}, \mathtt{HTCom}, \mathtt{HTDec}, \mathtt{HVer})$ is a hybrid trapdoor commitment scheme.

3. Hybrid trapdoor commitments: Constructions

We now show that hybrid trapdoor commitment schemes exist with respect to all the variants defined in Section 2.1. In particular, for each definition, we show both a construction based on general primitives and a practical construction based on number-theoretic assumptions.

We start with a construction for a hybrid trapdoor commitment scheme.

Theorem 3.1. *Under the assumption that one-way functions exist, there exists a hybrid trapdoor commitment scheme.*

Proof. Let f be a one-way function. Consider the commitment scheme proposed by Naor (see Appendix B for a discussion about Naor's commitment scheme), an instance $(\mathtt{Gen}, \mathtt{Com}, \mathtt{Ver})$ of such a commitment scheme can be constructed by using f. We now construct the following hybrid trapdoor commitment scheme HTCS $=$ $(\mathtt{HGen}, \mathtt{HTGen}, \mathtt{HCom}, \mathtt{HTCom}, \mathtt{HTDec}, \mathtt{HVer})$.

Algorithm $\mathtt{HGen}$ produces a random string $\mathtt{crs}$ that can be interpreted as $\mathtt{crs} = \mathtt{setup} \circ \mathtt{com}_0$ (where $\circ$ denotes concatenation). Here $\mathtt{setup}$ denotes the first message of Naor's commitment scheme and $\mathtt{com}_0$ is a random string.

Notice that since Naor's commitments are pseudorandom $\mathtt{com}_0$ corresponds to a valid commitment to 1^k only with negligible probability (this follows from the properties of Naor's commitment scheme).

Algorithms HTGen, HCom, HTCom, HTDec, HVer correspond to algorithms Gen, Com, TCom, TDec, Ver defined in the proof of Theorem 2.4. We report here again their descriptions by adding some comments that are useful for this proof.

Algorithm HCom on input crs and a message m runs the simulator S of Blum's Σ-protocol (again see Appendix B for more details about Σ protocols and an implementation of Blum's Σ-protocol based on the existence of one-way functions only) for proving that $(\mathtt{setup}, \mathtt{com}_0)$ is a commitment of string 1^k (which is with overwhelming probability a false statement). In particular S runs on input $(\mathtt{setup}, m)$ to obtain an accepting transcript $(\mathtt{setup}, a, m, z)$. The output of HCom is the pair $(\mathtt{com} = a, \mathtt{dec} = z)$.

Algorithm HVer simply verifies that $(\mathtt{setup}, \mathtt{com}, m, z)$ is an accepting transcript for the statement "$(\mathtt{setup}, \mathtt{com}_0)$ is a commitment of the string 1^k".

Algorithm HTGen outputs a string $\mathtt{crs} = \mathtt{setup} \circ \mathtt{com}_1$ precisely as HGen, the only difference is that $\mathtt{com}_1$ is computed as a commitment of the string 1^k. The second output of HTGen is the decommitment aux corresponding to the commitment $\mathtt{com}_1$. Notice that, assuming that Naor's commitment scheme is computationally hiding, the distribution of the crs generated by HTGen, is computationally indistinguishable from the distribution of the crs generated by HGen (which is a random string).

Algorithm HTCom, on input (crs, aux) computes a valid first message a for Blum's Σ-protocol for proving that $(\mathtt{setup}, \mathtt{com}_1)$ is a commitment of the string 1^k (which in this case is a true statement). HTCom outputs a and $\mathtt{aux}_a = \mathtt{crs} \circ \mathtt{aux} \circ \mathtt{aux}'$ where $\mathtt{aux}'$ is the auxiliary information generated by the prover algorithm of Blum's Σ-protocol when the first message a is computed.

Algorithm HTDec on input $\mathtt{aux}_a$ and a string m, runs the prover algorithm of Blum's Σ-protocol on input the transcript $(\mathtt{setup}, a, m)$ to obtain the last message z such that $(\mathtt{setup}, a, m, z)$ is an accepting transcript. HTDec can run the prover algorithm since he can pick from $\mathtt{aux}_a$ both the decommitment aux corresponding to $\mathtt{com}_1$ (i.e., the witness for the statement) and the auxiliary information $\mathtt{aux}'$ corresponding to a. The output of HTDec is $\mathtt{dec} = z$.

Efficiency, completeness and hybridness. Efficiency and completeness can be easily verified by inspection. Hybridness follows from the fact that a distinguisher of the distributions of the commitment parameters generated by HGen and HTGen can be used for breaking the pseudorandomness of Naor's commitments.

(HGen, HCom, HVer) is an unconditionally binding CS. When the commitment parameters are generated by algorithm HGen, unconditional binding follows from the fact that, when the sender runs the protocol, the statement $\mathtt{com}_0$ *is a commitment of the string* 1^k – inferred by the commitment parameters – is false. Thus, once the pair $(\mathtt{setup}, \mathtt{com} = a)$ is established, there exists only one pair (m, z) such that $(\mathtt{setup}, a, m, z)$ is an accepting transcript for the false statement "$\mathtt{com}_0$ *is a commitment of the string* 1^k" (otherwise it would be possible to extract a witness for a false statement, which is clearly impossible).

Now we show that a polynomial-time algorithm receiver that breaks the hiding property can be used to construct an algorithm $\mathcal{B}$ that breaks the hiding property of Naor's commitments. Assume there exists a receiver receiver that can distinguish between the distributions $\{(\mathtt{com}_0, \mathtt{dec}_0) \leftarrow \mathtt{HCom}(\mathtt{crs}, m_0) : \mathtt{com}_0\}$ and $\{(\mathtt{com}_1, \mathtt{dec}_1) \leftarrow \mathtt{HCom}(\mathtt{crs}, m_1) : \mathtt{com}_1\}$ with probability $1/2 + \mu$ (for some non-negligible quantity μ). On input a challenge com (which is either a commitment of 1^k or a commitment of 0^k) and the set-up message setup for Naor's commitment scheme, $\mathcal{B}$ constructs the commitment (HGen, HCom, HVer) as follows.

HGen simply sets $\mathtt{crs} = \mathtt{setup} \circ \mathtt{com}$. The algorithms HCom and HVer remain unchanged. Now, $\mathcal{B}$ flips a bit b, chooses two random messages m_0, m_1, computes $\mathtt{com}_b = \mathtt{HCom}(\mathtt{crs}, m_b)$ and runs receiver on input $\mathtt{com}_b, m_0, m_1$ (with common reference string crs). Eventually receiver stops and outputs $\hat{b}$ as its guess for the hidden bit b.

Now, if $\hat{b} = b$, $\mathcal{B}$ claims that com is a commitment of 0^k, otherwise it claims that com is a commitment of 1^k. This is because if com is a commitment of 0^k then the construction above gives rise to a valid commitment (i.e., $\mathtt{com}_b$ is a valid commitment of m_b) and thus the advantage of receiver in predicting b is close to μ.

On the other hand, if com is a commitment of 1^k then receiver can not guess b with probability non-negligibly better than $1/2$. This is because, this time, the statement "com is a commitment of the string 1^k" is a true one. Thus, for a fixed value $\mathtt{com}_b$, and for any message m there always exists a value z such that $(\mathtt{setup}, \mathtt{com}_b, m, z)$ is an accepting transcript.[1] This means that the value $\mathtt{com}_b$ does not reveal any information about m_b. Thus, `receiver` can not guess b with probability non-negligibly better than $1/2$.

(`HTGen`, `HCom`, `HTCom`, `HTDec`, `HVer`) **is a trapdoor CS.** This follows directly by Theorem 3.1, by observing that (`HTGen`, `HCom`, `HTCom`, `HTDec`, `HVer`) are the same algorithms as used in that Theorem.

From the previous properties, we have that HTCS enjoys both hybridness and trapdoorness, therefore the claim holds.

Theorem 3.2. *Under the assumption that the Decisional Diffie–Hellman problem is hard, there exists an efficient hybrid trapdoor commitment scheme.*

Proof. We use the same proof ideas as used to prove Theorem 3.1. To gain in efficiency however, we consider more specific number-theoretic constructions. In what follows we only discuss the aspects of the proof which are different with respect to the previous one. First of all, while in Theorem 3.1 the security is proved by resorting to the properties of Naor's commitment scheme, in this case we use the DDH assumption.

Let G be a group of order q where the DDH problem is conjectured to be hard. Informally this means that given two random elements $g, h \neq 1 \in G$, it is computationally infeasible to distinguish between (g, h, g^r, h^r) and (g, h, g^{r_1}, h^{r_2}) for three random elements $r, r_1, r_2 \in Z_q$.

Algorithm `HGen` outputs a random string `crs`. From a random string it is possible to deterministically extract a quadruple such that the probability that its distribution is different from (g, h, g^{r_1}, h^{r_2}) is negligible. Instead, `HTGen` outputs a `crs` and the auxiliary information $\mathtt{aux} = r$, such that by running the same deterministic extraction procedure, the resulting quadruple has the same distribution as (g, h, g^r, h^r). Notice that by the hardness of the DDH problem, the output `crs` of `HTGen` is computationally indistinguishable from a random string.

Consider the Σ-protocol for proving the equality of two discrete logarithms (see Appendix B). Let us call ∇ this protocol. Denoting by (g, h, g_1, h_1) the quadruple in the `crs`, we use ∇ to prove that g_1 and h_1 have the same discrete logarithm with respect to bases g and h, respectively.

As for the case of the proof of Theorem 3.1, when `HGen` is used, the instance for the Σ-protocol is false. Consequently, once a commitment a is sent, it can correspond to at most one message m such that there exists an accepting transcript (a, m, z). When the commitment parameters (g, h, g^r, h^r) are established by `HTGen`, then knowledge of r (the trapdoor) allows one to send a commitment a that can later be opened as any possible message m. This is because, by running the prover algorithm of the Σ-protocol on input r (as witness) and m (as challenge), it is always possible to find a z such that (a, m, z) is an accepting transcript. In this a commitment computed and opened by means of the trapdoor is statistically indistinguishable from the one that is honestly computed and opened and therefore, is indistinguishable even if the receiver knows the trapdoor. This means that, standard hybrid arguments show that the security is preserved for any polynomial number of commitments. The efficiency of this commitment scheme directly follows from the efficiency of the considered Σ-protocol. □

Other efficient implementations based on Paillier's Decisional Composite Residuosity Assumption [60] enjoy also the extractability property and can be found in Appendix C.

A construction for hybrid multi-trapdoor (resp., simulation-sound trapdoor) commitment schemes may seem, at first, much harder to achieve. After all, multi-trapdoor commitments need more complex parameters (with respect to basic trapdoor ones) and, to have a hybrid version of them, we need to make sure that these parameters remain distributed in such a way that it should be hard to say which of the two commitment generation algorithms was used to produce them.

Informally, we solve this problem by composing a multi-trapdoor (resp., simulation-sound trapdoor) commitment scheme with a hybrid trapdoor commitment scheme as those described so far. The composition is made by considering the concatenation of *both* commitment parameters. Moreover all the operations made by the committing and decommitting algorithms are performed twice, once for each *subscheme*. Intuitively, using this technique, when a multi-trapdoor (resp., simulation-sound trapdoor) commitment scheme is composed with the hybrid trapdoor commitment scheme instantiated as a trapdoor commitment scheme, the resulting scheme is still a multi-trapdoor

[1] Note that, even though such a z is guaranteed to exist, $\mathcal{B}$ may not be able to efficiently compute it. This is not a problem for our proof as all that we need is to make sure that such a z actually exists.

(resp., simulation-sound trapdoor) commitment scheme. On the other hand, if the hybrid trapdoor commitment scheme is instantiated as an unconditionally binding commitment scheme, then the resulting scheme is unconditionally binding. By the indistinguishability of the commitment parameters of the two instantiations we obtain the desired result.

More formally, we prove the following theorems.

Theorem 3.3. *Under the assumption that multi-trapdoor commitment schemes exist there exists a hybrid multi-trapdoor commitment scheme.*

Proof. First of all, the existence of multi-trapdoor commitment schemes implies the existence of one-way functions and therefore by Theorem 3.1 implies the existence of hybrid trapdoor commitment schemes.

Let $C_0 = (\mathtt{Gen}, \mathtt{Sel}, \mathtt{Tkg}, \mathtt{Com}, \mathtt{TCom}, \mathtt{TDec}, \mathtt{Ver})$ and $C_1 = (\mathtt{HGen}, \mathtt{HTGen}, \mathtt{HCom}, \mathtt{HTCom}, \mathtt{HTDec}, \mathtt{HVer})$ be respectively a multi-trapdoor commitment scheme and a hybrid trapdoor commitment scheme.

We now show a construction of $C = (\mathtt{HGen}', \mathtt{HTGen}', \mathtt{HSel}', \mathtt{HTkg}', \mathtt{HCom}', \mathtt{HTCom}', \mathtt{HTDec}', \mathtt{HVer}')$ which is a hybrid multi-trapdoor commitment scheme.

$\mathtt{HGen}'(1^k)$ outputs $\mathtt{crs} = \mathtt{PK}_0 \circ \mathtt{crs}_1$ where $(\mathtt{PK}_0, \mathtt{TK}_0)$ is the output of $\mathtt{Gen}(1^k)$ and $\mathtt{crs}_1$ is the output of $\mathtt{HGen}(1^k)$.

$\mathtt{HTGen}'(1^k)$ outputs $(\mathtt{crs} = \mathtt{PK}_0 \circ \mathtt{crs}_1, \mathtt{aux} = (\mathtt{TK}_0, \mathtt{aux}_1))$ where $(\mathtt{PK}_0, \mathtt{TK}_0)$ is the output of $\mathtt{Gen}(1^k)$ and $(\mathtt{crs}_1, \mathtt{aux}_1)$ is the output of $\mathtt{HTGen}(1^k)$.

$\mathtt{HSel}'(\mathtt{crs} = \mathtt{PK}_0 \circ \mathtt{crs}_1)$ outputs $\mathtt{pk} = (\mathtt{pk}_0, \mathtt{crs}_1)$ where $\mathtt{pk}_0$ is the output of $\mathtt{Sel}(\mathtt{PK}_0)$.

$\mathtt{HTkg}'(\mathtt{aux} = (\mathtt{TK}_0, \mathtt{aux}_1), \mathtt{pk})$ outputs $\mathtt{tk} = (\mathtt{tk}_0, \mathtt{aux}_1)$ where $\mathtt{tk}_0$ is the output of $\mathtt{Tkg}(\mathtt{TK}_0, \mathtt{pk})$.

$\mathtt{HCom}'(\mathtt{pk} = \mathtt{pk}_0 \circ \mathtt{crs}_1, v)$ outputs $(\mathtt{com} = (\mathtt{com}_0, \mathtt{com}_1), \mathtt{dec} = (\mathtt{dec}_0, \mathtt{dec}_1))$ where $(\mathtt{com}_0, \mathtt{dec}_0)$ is the output of $\mathtt{Com}(\mathtt{pk}_0, v)$ and $(\mathtt{com}_1, \mathtt{dec}_1)$ is the output of $\mathtt{HCom}(\mathtt{crs}_1, v)$.

$\mathtt{HTCom}'(\mathtt{tk} = (\mathtt{tk}_0, \mathtt{aux}_1), v)$ outputs $(\mathtt{com} = (\mathtt{com}_0, \mathtt{com}_1), \mathtt{aux}_{\mathtt{com}} = (\mathtt{aux}_{\mathtt{com}_0}, \mathtt{aux}_{\mathtt{com}_1}))$ where $(\mathtt{com}_0, \mathtt{aux}_{\mathtt{com}_0})$ is the output of $\mathtt{TCom}(\mathtt{tk}_0)$ and $(\mathtt{com}_1, \mathtt{aux}_{\mathtt{com}_1})$ is the output of $\mathtt{HTCom}(\mathtt{aux}_1)$.

$\mathtt{HTDec}'(\mathtt{aux}_{\mathtt{com}} = (\mathtt{aux}_{\mathtt{com}_0}, \mathtt{aux}_{\mathtt{com}_1}), v')$ outputs $\mathtt{dec}' = (\mathtt{dec}'_0, \mathtt{dec}'_1)$ where $\mathtt{dec}'_0$ and $\mathtt{dec}'_1$ are respectively the output of $\mathtt{TDec}(\mathtt{aux}_{\mathtt{com}_0}, v')$ and the output of $\mathtt{HTDec}(\mathtt{aux}_{\mathtt{com}_1}, v')$.

$\mathtt{HVer}'(\mathtt{pk} = \mathtt{pk}_0 \circ \mathtt{crs}_1, \mathtt{com} = (\mathtt{com}_0, \mathtt{com}_1), \mathtt{dec} = (\mathtt{dec}_0, \mathtt{dec}_1), v)$ outputs a bit $b = (b_0 \wedge b_1)$ where $\mathtt{Ver}(\mathtt{pk}_0, \mathtt{com}_0, \mathtt{dec}_0, v) = b_0$ and $\mathtt{HVer}(\mathtt{crs}_1, \mathtt{com}_1, \mathtt{dec}_1, v) = b_1$.
Efficiency and completeness are straightforward.

Multi-trapdoorness. Now we prove that $(\mathtt{HTGen}', \mathtt{HSel}', \mathtt{HTkg}', \mathtt{HCom}', \mathtt{HTCom}', \mathtt{HTDec}', \mathtt{HVer}')$ is a multi-trapdoor commitment scheme. In particular we have to prove the following properties: multi-trapdoorness, hiding and binding.

To prove that the given scheme is a multi-trapdoor one, we assume that there exists an adversary $\mathcal{A}$ that breaks the multi-trapdoor property with some (non-negligible) advantage μ. Then we show how to build an efficient algorithm $\mathcal{B}$ which uses $\mathcal{A}$ to break either the multi-trapdoorness of C_0 or the trapdoorness of C_1. We consider the following games:

Game G_0: A commitment on a message m is generated by concatenating a commitment generated using the function $\mathtt{Com}$ (i.e., the commitment function of C_0) and a commitment generated using $\mathtt{HCom}$ (i.e., the commitment function of C_1).

Game G_1: A commitment on a message m is generated by concatenating a commitment generated using $\mathtt{Com}$ and a commitment generated using the function $\mathtt{HTCom}$ (i.e., the equivocation function of C_1).

Game G_2: A commitment on a message m is generated by concatenating a commitment generated using the function $\mathtt{TCom}$ (i.e., the equivocation function of C_0) and a commitment generated using $\mathtt{HTCom}$.

We define P_i $(i = 0, 1, 2)$ as the probability that $\mathcal{A}$ succeeds in game G_i. Since we are assuming that $\mathcal{A}$ has a (non-negligible) advantage μ in breaking the multi-trapdoorness property of C, we have that

$$\mu = |P_0 - P_2| \leq |P_0 - P_1| + |P_1 - P_2|$$

Thus either $|P_0 - P_1|$ or $|P_1 - P_2|$ has to be non-negligible.

In what follows we focus on the simpler case in which the adversary asks one commitment only. The more general case in which A may ask polynomially many commitments goes as discussed in the proof of Theorem 3.1.

If $|P_0 - P_1|$ is non-negligible, we proceed as follows. $\mathcal{B}$ receives on input, from a challenging oracle $\mathcal{O}_M$ the public parameters for a multi-trapdoor commitment scheme C_0 and uses $\mathcal{A}$ to break the multi-trapdoorness property of C_0 as follows. First it runs the HGen algorithm to generate a hybrid trapdoor commitment scheme C_1 (for which, of course it will know the trapdoor to equivocate). Next it chooses a random message m and asks $\mathcal{O}_M$ to commit to it. The oracle works as follows. On input m it flips a random bit b and, if $b = 0$ it commits to m using the function Com, otherwise it commits to m using the function TCom (in fact, in this latter case B would commit to nothing). In both cases the oracle outputs the commitment com, and the corresponding decommitment dec for m. Next $\mathcal{B}$ commits to m using HCom and outputs the composition of the two commitments (i.e., the one received from the oracle and the one it produced). Clearly if $\mathcal{A}$ on input m, the received commitment and the corresponding decommitment, has some non-negligible advantage in guessing if it is playing game G_0 or game G_1, this can be used here to guess b with non-negligible advantage (and then to break the multi-trapdoorness property of C_0).

If $|P_1 - P_2|$ is non-negligible, $\mathcal{B}$ receives on input, from a challenging oracle $\mathcal{O}_h$ the public parameters for a hybrid trapdoor commitment scheme C_1 and uses $\mathcal{A}$ to break the trapdoorness property of C_1 as follows. First it runs the Gen algorithm to generate a multi-trapdoor commitment scheme C_0. Next it chooses a random message m and asks $\mathcal{O}_h$ to either produce a commitment on m using HCom or using HTCom (according to a hidden bit b). Let com*, dec* be the commitment and decommitment returned by the oracle. Next $\mathcal{B}$ commits to m using TCom and outputs the composition of the two commitments (i.e., the one received from the oracle and the one it produced) and the two decommitments. Once again, if $\mathcal{A}$ on input m, the commitment and decommitment, has some non-negligible advantage in guessing if it is playing game G_1 or game G_2 this can be used to guess b with non-negligible advantage (and then to break the trapdoorness property of C_1).

The hiding property can be proved similarly. Assume that we have an adversary $\mathcal{A}$ that breaks the hiding property of C; we show here how to use it to break the hiding property of either C_0 or C_1. We consider the following games based on two messages m_1, m_0 (such that $m_0 \neq m_1$ and $|m_0| = |m_1|$).

Game G_0: A commitment on the message m_0 is generated by concatenating a commitment (to m_0) generated using the function Com (i.e., the commitment function of C_0) and a commitment generated using HCom (i.e., the commitment function of C_1).

Game G_1: A commitment is generated by concatenating a commitment to m_0 generated using the function Com and a commitment to m_1 generated using HTCom.

Game G_2: A commitment on the message m_1 is generated by concatenating a commitment generated using the function TCom and a commitment generated using HTCom.

The rest proceeds exactly as for the multi-trapdoorness property.

The binding property, on the other hand, can be reduced to the binding property of the multi-trapdoor commitment scheme only. More precisely we prove that if there is an attacker $\mathcal{A}$ for this property then it can be used to construct an efficient algorithm $\mathcal{B}$ that breaks the binding property of the underlying multi-trapdoor commitment scheme.

$\mathcal{B}$ goes as follows. It receives on input the public parameters for the multi-trapdoor commitment scheme C_0 and it generates a hybrid trapdoor commitment scheme C_1. Finally it sets $C = C_0 \circ C_1$ as the hybrid multi-trapdoor commitment scheme. Note that, by these positions, the public parameters of C_1 are totally independent with respect to those of C_0. This is crucial for our proof to go through correctly.

Now whenever $\mathcal{A}$ asks for an equivocation for some key $\mathtt{pk}_i$, $\mathcal{B}$ "transforms" $\mathtt{pk}_i$ into a valid public key $\mathtt{pk}'_i$ for C_0 and asks a similar question of equivocation oracle for the multi-trapdoor commitment scheme. Once it gets back an answer a, it creates a fake commitment a' for C_1 using its own trapdoor and sends $a \circ a'$ to $\mathcal{A}$. If $\mathcal{A}$ produces an equivocation on a previously unasked key, $\mathcal{B}$ can easily use it to break the binding property of C_0 in an obvious way.

Hybridness. The commitment parameters have two different distributions that only depend on the generation algorithm of the hybrid trapdoor commitment subscheme. Therefore, a distinguisher of the two distributions of the commitment parameters generated by our scheme can be easily reduced to a distinguisher of the distributions of the commitment parameters generated by the hybrid trapdoor commitment subscheme used in the construction.

Remark. Algorithms HGen and HTGen run on input the same security parameter 1^k and obviously each of them uses a different randomness. Notice that this is required by our construction of hybrid multi-trapdoor commitment schemes. Indeed, if the master trapdoor of the multi-trapdoor commitment subscheme and the trapdoor of the hybrid trapdoor commitment subscheme are related, the binding of the resulting hybrid multi-trapdoor commitment scheme could not hold anymore. This is due to the game played by the adversary of the binding property that by means of a polynomial number of accesses to a decommitting oracle, could compute from two different openings of the same commitment, the trapdoor of the hybrid trapdoor commitment subscheme. Obviously, if this trapdoor is related to the master trapdoor of the multi-trapdoor commitment subscheme or even to a local trapdoor not used during the queries to the oracle, the adversary violates the binding of the scheme. □

Note that, with the theorem above, we show how to construct hybrid multi-trapdoor commitments from the hypothesis that multi-trapdoor commitments exist. We also show that multi-trapdoor commitments exist iff secure signature against *generic* chosen message attack exist (see Appendix D for further discussions about these objects).

Notice that one-way functions are equivalent to secure signatures [67] in the sense of [49], which, in turn, imply secure signatures against *generic* chosen message attack exist. This means that Theorem 3.3 can be restated as follows.

Theorem 3.4. *Under the assumption that one-way functions exist, there exists a hybrid multi-trapdoor commitment scheme.*

As for the case of hybrid trapdoor commitments we briefly discuss some efficient implementations based on number theoretic assumptions. These constructions can be obtained straightforwardly from the two efficient multi-trapdoor commitment schemes presented in [44] (the reader is referred to [44] for more details). In particular, the first relies on the strong RSA assumption and on the decisional Diffie–Hellman assumption. The second implementation can be shown to be secure under the strong Diffie–Hellman assumption and, again, on the decisional Diffie–Hellman assumption. The reader is referred to Appendix A for a discussion about the strong RSA assumption and the strong Diffie–Hellman assumption.

Theorem 3.5. *Under the assumption that the strong RSA and DDH problems are hard (or under the assumption that the strong Diffie–Hellman [9] and DDH problems are hard), there exists an efficient hybrid multi-trapdoor commitment scheme.*

Proof. The construction of a hybrid multi-trapdoor commitment scheme given in the proof of Theorem 3.3 used a modular composition of a multi-trapdoor commitment scheme and a hybrid trapdoor commitment scheme. Therefore, by using the results of [44] (see the full version at [43]) and Theorem 3.2 the claim holds. □

Theorem 3.6. *Under the assumption that one-way functions exist, there exists a hybrid simulation-sound trapdoor commitment scheme.*

Proof. Let $C_0 = (\mathtt{Gen}, \mathtt{Com}, \mathtt{TCom}, \mathtt{TDec}, \mathtt{Ver})$ be the simulation-sound trapdoor commitment scheme presented in [55] that is based on the existence of a one-way function f. By Theorem 3.1 the existence of f implies the existence of a hybrid trapdoor commitment scheme $C_1 = (\mathtt{HGen}, \mathtt{HTGen}, \mathtt{HCom}, \mathtt{HTCom}, \mathtt{HTDec}, \mathtt{HVer})$. We use both C_0 and C_1 to construct $C = (\mathtt{HGen}', \mathtt{HTGen}', \mathtt{HCom}', \mathtt{HTCom}', \mathtt{HTDec}', \mathtt{HVer}')$ which is a hybrid simulation-sound trapdoor commitment scheme.

$\mathtt{HGen}'(1^k)$ outputs ($\mathtt{crs} = \mathtt{crs}_0 \circ \mathtt{crs}_1$) where $\mathtt{crs}_0$ is the output of $\mathtt{Gen}(1^k)$ and $\mathtt{crs}_1$ is the output of $\mathtt{HGen}(1^k)$.

$\mathtt{HTGen}'(1^k)$ outputs ($\mathtt{crs} = \mathtt{crs}_0 \circ \mathtt{crs}_1, \mathtt{aux} = (\mathtt{aux}_0, \mathtt{aux}_1)$ where $(\mathtt{crs}_0, \mathtt{aux}_0)$ is the output of $\mathtt{Gen}(1^k)$ and $(\mathtt{crs}_1, \mathtt{aux}_1)$ is the output of $\mathtt{HTGen}(1^k)$.

$\mathtt{HCom}'(\mathtt{crs}, v, \mathtt{tag})$ outputs $(\mathtt{com} = (\mathtt{com}_0, \mathtt{com}_1), \mathtt{dec} = (\mathtt{dec}_0, \mathtt{dec}_1))$ where $(\mathtt{com}_0, \mathtt{dec}_0)$ is the output of $\mathtt{Com}(\mathtt{crs}_0, v, \mathtt{tag})$ and $(\mathtt{com}_1, \mathtt{dec}_1)$ is the output of $\mathtt{HCom}(\mathtt{crs}_1, v)$.

$\mathtt{HTCom}'(\mathtt{crs} = \mathtt{crs}_0 \circ \mathtt{crs}_1, \mathtt{aux} = (\mathtt{aux}_0, \mathtt{aux}_1), v, \mathtt{tag})$ outputs $(\mathtt{com} = (\mathtt{com}_0, \mathtt{com}_1), \mathtt{aux}_{\mathtt{com}} = (\mathtt{aux}_{\mathtt{com}_0}, \mathtt{aux}_{\mathtt{com}_1}))$ where $(\mathtt{com}_0, \mathtt{aux}_{\mathtt{com}_0})$ is the output of $\mathtt{TCom}(\mathtt{aux}_0, v, \mathtt{tag})$ and $(\mathtt{com}_1, \mathtt{aux}_{\mathtt{com}_1})$ is the output of $\mathtt{HTCom}(\mathtt{aux}_1, v)$.

$\mathtt{HTDec}'(\mathtt{aux}_{\mathtt{com}} = (\mathtt{aux}_{\mathtt{com}_0}, \mathtt{aux}_{\mathtt{com}_1}), \mathtt{com} = (\mathtt{com}_0, \mathtt{com}_1), v, \mathtt{tag})$ outputs $\mathtt{dec} = (\mathtt{dec}_0, \mathtt{dec}_1)$ where $\mathtt{dec}_0$ is the output of $\mathtt{TDec}(\mathtt{aux}_{\mathtt{com}_0}, v, \mathtt{tag})$ and $\mathtt{dec}_1$ is the output of $\mathtt{HTDec}(\mathtt{aux}_{\mathtt{com}_1}, v)$.

$\mathtt{HVer}'(\mathtt{crs} = \mathtt{crs}_0 \circ \mathtt{crs}_1, \mathtt{com} = (\mathtt{com}_0, \mathtt{com}_1), \mathtt{dec} = (\mathtt{dec}_0, \mathtt{dec}_1), v, \mathtt{tag})$ outputs a bit $b = (b_0 \wedge b_1)$ where $\mathtt{Ver}(\mathtt{crs}_0, \mathtt{com}_0, \mathtt{dec}_0, v, \mathtt{tag}) = b_0$ and $\mathtt{HVer}(\mathtt{crs}_1, \mathtt{com}_1, \mathtt{dec}_1, v) = b_1$.

The proof of this theorem follows the one for Theorem 3.3.

Efficiency and completeness are straightforward.

Simulation soundness. Now we prove that $(\mathtt{HTGen}', \mathtt{HCom}', \mathtt{HTCom}', \mathtt{HTDec}', \mathtt{HVer}')$ is a simulation-sound trapdoor commitment scheme. In other words we have to prove that:

1. $(\mathtt{HTGen}', \mathtt{HCom}', \mathtt{HTCom}', \mathtt{HTDec}', \mathtt{HVer}')$ is a tag-based trapdoor commitment scheme;
2. the simulation-sound binding property holds.

To prove the first statement we start by showing that given $(\mathtt{crs}, \mathtt{aux}) \leftarrow \mathtt{HTGen}'(1^k)$ then the two distributions

$$\{(\mathtt{com}, \mathtt{dec}) \leftarrow \mathtt{HCom}'(\mathtt{crs}, v) : (\mathtt{crs}, \mathtt{com}, \mathtt{dec}, v)\} \quad \text{and}$$
$$\{(\mathtt{com}', \mathtt{aux}_{\mathtt{com}'}) \leftarrow \mathtt{HTCom}'(\mathtt{aux}); \mathtt{dec}' \leftarrow \mathtt{HTDec}'(\mathtt{aux}_{\mathtt{com}'}, v) : (\mathtt{crs}, \mathtt{com}', \mathtt{dec}', v)\}$$

are computationally indistinguishable.

The following proof is, *mutatis mutandis*, the same as the one given to prove the multi-trapdoorness property of the scheme presented in Theorem 3.3. Thus, here we skip some obvious details already discussed there.

Assume, for the sake of contradiction, that this does not hold, than we show that either C_0 or C_1 can not be a trapdoor commitment scheme. We consider the following games

Game G_0: A commitment (with respect to tag *tag*) on a message m is generated by concatenating a commitment generated using the function $\mathtt{Com}$ (i.e., the commitment function of C_0) and a commitment generated using $\mathtt{HCom}$ (i.e., the commitment function of C_1).

Game G_1: A commitment (with respect to tag *tag*) on a message m is generated by concatenating a commitment generated using the function $\mathtt{TCom}$ (i.e., the equivocation function of C_0) and a commitment generated using $\mathtt{HCom}$.

Game G_2: A commitment on a message m (with respect to tag *tag*) is generated by concatenating a commitment generated using the function $\mathtt{TCom}$ and a commitment generated using $\mathtt{HTCom}$ (i.e., the equivocation function of C_1).

We define P_i ($i = 0, 1, 2$) as the probability that $\mathcal{A}$ (i.e., the adversary that is assumed to break the trapdoorness property of C) succeeds in game G_i. Since we are assuming that $\mathcal{A}$ has a (non-negligible) advantage μ in breaking the simulation-soundness property of C, we have that

$$\mu = |P_0 - P_2| \leq |P_0 - P_1| + |P_1 - P_2|.$$

Thus either $|P_0 - P_1|$ or $|P_1 - P_2|$ has to be non-negligible.

In the former case we proceed as follows. We construct an algorithm $\mathcal{B}$ that receives on input, from a challenging oracle $\mathcal{O}_{SS}$ the public parameters for a simulation sound trapdoor commitment scheme C_0 and uses $\mathcal{A}$ to break the trapdoor property of C_0 as follows. First it runs the $\mathtt{HTGen}$ algorithm to generate a hybrid trapdoor commitment scheme C_1. Next it chooses a random message m and asks $\mathcal{O}_{SS}$ to commit to it. The oracle works as follows. On input m it flips a random bit b and, if $b = 0$ it commits to m using the function $\mathtt{Com}$, otherwise it commits using the function $\mathtt{TCom}$. In both cases, the oracle outputs the computed commitment along with the corresponding decommitment to m. Next $\mathcal{B}$ commits to m using $\mathtt{HCom}$ and outputs the composition of the two commitments (i.e., the one received from the oracle and the one it produced) and of the two decommitments. Clearly if $\mathcal{A}$ has some non-negligible advantage in guessing if it is playing game G_0 or game G_1 this can be used here to guess b with non-negligible advantage (and then to break the trapdoorness property of C_0).

A similar argument can be done for the case on which $|P_1 - P_2|$ is non-negligible.

The hiding property can be proved similarly. Assume that we have an adversary $\mathcal{A}$ that breaks the hiding property of C; we show how to use it to break the hiding property of either C_0 or C_1. The rest of the proof goes exactly as for the case of the hiding property for the multi-trapdoor commitment scheme discussed in the proof of Theorem 3.3.

The binding property can be proved exactly as we proved the binding property for the multi-trapdoor commitment scheme of Theorem 3.3.

It remains to prove the second statement, i.e., that the simulation sound binding property holds. We prove this by showing that if there exists an adversary $\mathcal{A}$ that breaks the simulation sound binding property of C, it can be used, by an algorithm $\mathcal{B}$, to break the same property of C_0. $\mathcal{B}$ goes as follows. On a preliminary phase $\mathcal{B}$ runs a key generation algorithm to generate a hybrid trapdoor commitment scheme C_1. Then it receives as input, from a challenging oracle $\mathcal{O}_{\mathtt{SSTC}}$, the public parameters for a simulation sound trapdoor commitment scheme C_0 and generates C as described in the very first part of the proof of this theorem.

Whenever $\mathcal{A}$ asks an equivocation query on C, $\mathcal{B}$ asks for a corresponding equivocation query on C_0 to the oracle $\mathcal{O}_{\mathrm{SSTC}}$ and uses the received answer to produce a valid equivocation for C. Thus if $\mathcal{A}$ breaks the simulation-sound binding property of C, this immediately leads $\mathcal{B}$ to break the simulation-sound binding property of C_1.

Hybridness. The commitment parameters have two different distributions that only depend on the generation algorithm of the hybrid trapdoor commitment subscheme. Therefore, a distinguisher of the two distributions of the commitment parameters generated by our scheme can be easily reduced to a distinguisher of the distributions of the commitment parameters generated by hybrid trapdoor commitment subscheme used in the construction. □

Theorem 3.7. *Under the assumption that the DSA signature scheme is secure and the DDH problem is hard (or under the assumption that the Cramer–Shoup signature scheme [25] is secure and the DDH problem is hard), there exists an efficient hybrid simulation-sound trapdoor commitment scheme.*

Proof. The construction of a hybrid simulation-sound trapdoor commitment scheme given in the proof of Theorem 3.6 used a modular composition of a simulation-sound trapdoor commitment scheme and a hybrid trapdoor commitment scheme. Therefore, by using the results of [55] and Theorem 3.2 the claim holds. □

4. Hybrid trapdoor commitments: Applications

In this section we describe some important applications of our primitive. In particular we show that hybrid trapdoor commitments can be used to construct interactive protocols that achieve strong notions of zero knowledge. More precisely we improve the computational soundness of the concurrent zero-knowledge arguments of [29], and of both the simulation-sound and left-concurrent non-malleable zero-knowledge arguments of [44,55] by showing how to achieve zero-knowledge *proofs* (rather than only *arguments*). Therefore the security of our constructions holds even against computationally unbounded adversarial provers. Moreover, our zero-knowledge proofs can be based on the same complexity-theoretic assumptions as used in [29,44,55]. The efficient constructions also require the hardness of the *DDH* problem.

4.1. Background

The classical notion of a zero-knowledge proof system has been introduced in [48]. Roughly speaking, in a zero-knowledge proof system a prover can prove to a verifier the validity of a statement without releasing any additional information. Since its introduction, the concept of a zero-knowledge proof system and the simulation paradigm have been widely used to prove the security of many protocols.

Concurrent zero knowledge. The notion of *concurrent zero knowledge* [35] formalizes security in a scenario in which several verifiers access concurrently a prover and maliciously coordinate their actions so as to extract information from the prover. In [15] it has been shown that in the black-box model $\tilde{\Omega}(\log n)$ rounds are necessary for concurrent zero knowledge for non-trivial languages. The first concurrent zero-knowledge proof system for $\mathcal{NP}$ has been given by [65] that showed that $O(n^{\epsilon})$ rounds are sufficient for any $\epsilon > 0$. Poly-logarithmic round-complexity was achieved in [52] and, finally, in [64] it is shown that $\tilde{O}(\log n)$ rounds are sufficient. The proof systems presented in [65,52,64] are black-box zero knowledge and the round-complexity of the proof system of [64] is almost optimal in view of the lower bound proved in [15]. Thus unlike the stand-alone case, black-box concurrent zero knowledge can not be achieved in a constant number of rounds.

Different models have been presented in which round-efficient black-box concurrent zero knowledge is possible. In [35,36,45] constant-round concurrent zero-knowledge proof systems have been presented by relaxing the asynchrony of the model or the zero-knowledge property. In [29,8], constant-round concurrent zero-knowledge proof systems have been presented assuming the existence of a common reference string or a shared random string (i.e., a trusted third party) while in [33,70] a constant-round concurrent zero-knowledge with concurrent soundness argument system is shown by assuming that there exists a public repository that contains the public keys of the verifiers. In [63] an almost constant-round concurrent zero-knowledge argument systems is presented by assuming the existence of only one stateful prover. Furthermore, Pass [61] gave a constant-round concurrent zero-knowledge argument with a super-polynomial-time simulator. In [1], Barak presented a *non-black-box* constant-round *bounded-concurrent* zero-knowledge argument system. The construction of [1] assumes that the maximum number of concurrent sessions is known in advance.

Simulation-sound zero knowledge. Simulation-sound zero knowledge has been introduced in [68] for the purpose of constructing cryptosystems secure against adaptive chosen-ciphertext attacks. This concept is related to the concept of non-malleability introduced in [34]. Indeed, both notions deal with an adversary (called the *man-in-the-middle*) that simultaneously participates in many executions of a proof systems and acts as a prover in some of them and as a verifier in the other ones. The adversary has complete control over the scheduling of the messages in the executions of the protocols. Informally, a zero-knowledge proof system is said to be concurrent *simulation sound* if the information that the man-in-the-middle adversary collects as a verifier from concurrent sessions played with a *simulated* prover does not help him to prove a *false* statement. Here the man-in-the-middle can choose to see *simulated* proofs of true and false statements.

Simulation-sound zero knowledge plays an important role for proving the security of protocols. Indeed, when the simulation paradigm is used to prove the security of a protocol, the simulator could, in some cases, need to simulate the proof of a false statement. Here simulation soundness is crucial since the adversary could gain knowledge from such a proof in order to prove a false statement in another protocol.

4.2. Definitions

Notation. We use the same notation as [55,54]. Let A and B be two interactive algorithms, we define $\langle A, B\rangle_{\mathtt{crs}}(x)$ as the local output of B after an interactive execution with A using crs as common reference string. The *transcript* of an algorithm is the pair composed by its input and output. Two transcripts *match* if the ordered input messages of the first are equivalent to the ordered output of the other, and vice versa. We use the notation $tr \bowtie tr'$ to indicate that tr matches tr'. Moreover we denote by $\overline{A}$ a wrapper algorithm that handles concurrent interactions of A. More precisely, $\overline{A}$ can receive two types of input:

1. $(\mathit{START}, \alpha, x, w)$. For this input the wrapper starts a new (interactive) session of A, with label α, common input x, (freshly generated) randomness r and private input w. We require that the new machine uses the common reference of $\overline{A}$.
2. $(\mathit{MSG}, \alpha, m)$. For this input the wrapper sends m to the interactive machine labeled with α (if such a machine exists) and outputs the output message of that machine.

We denote by (x, tr, b) the output of $\overline{A}$ where x is the common input, tr the transcript of the sessions (input and output messages) and b the output. We say that B and C are *coordinated* if they have a single control, but distinct inputs and outputs. Given four interactive algorithms A, B, C and D we denote by $(\langle A, B\rangle, \langle C, D\rangle)_{\mathtt{crs}}$ the output of D after an interactive execution with C and an interactive execution of A and B, all using crs as a common reference string.

Definition 4.1 (*Unbounded Zero Knowledge Proofs*). Let L be an $\mathcal{NP}$-language, $\mathtt{G}$, P, V, S_0, S_1 be probabilistic polynomial-time algorithms. We say that $\Pi = (\mathtt{G}, P, V, S = (S_0, S_1))$ is an unbounded concurrent zero-knowledge proof system in the common reference string model for L, if:

- **completeness:** for every $x \in L$ where $|x| = \mathtt{POLY}(k)$, all w such that $R_L(x, w) = 1$ there exists a negligible function ν such that

$$\mathtt{Prob}(\mathtt{crs} \leftarrow \mathtt{G}(1^k) : \langle P(w), V\rangle_{(\mathtt{crs})}(x) = 0) \leq \nu(k);$$

- **soundness:** for any unbounded algorithm $P^\star$ there exists a negligible function ν such that for any $x \notin L$,

$$\mathtt{Prob}(\mathtt{crs} \leftarrow \mathtt{HGen}(1^k); \langle P^\star, V\rangle_{(\mathtt{crs})}(x) = 1) \leq \nu(k);$$

- **unbounded concurrent zero knowledge:** for any polynomial-time adversarial verifier $V^\star$ there exists a negligible function ν such that

$$|\mathtt{Prob}(\mathtt{Expt}_{V^\star}(k) = 1) - \mathtt{Prob}(\mathtt{Expt}^S_{V^\star}(k) = 1)| \leq \nu(k)$$

where the experiments $\mathtt{Expt}_{V^\star}(k)$ and $\mathtt{Expt}^S_{V^\star}(k)$ are defined as follows:

$\mathtt{Expt}_{V^\star}(k)$:	$\mathtt{Expt}^S_{V^\star}(k)$:
$\mathtt{crs} \leftarrow \mathtt{G}(1^k)$	$(\mathtt{crs}, \mathtt{aux}) \leftarrow S_0(1^k)$
$\mathtt{return}\langle\overline{P}, V^\star\rangle_{\mathtt{crs}}$	$\mathtt{return}\langle\overline{S'(\mathtt{aux})}, V^\star\rangle_{\mathtt{crs}}$

where $S'(\mathtt{aux})$ on input a common reference string $\mathtt{crs}$, common input x and private input w, computes $b = R_L(x, w)$ and then if $b = 1$ he runs $S_1(\mathtt{aux}, x)$, otherwise he aborts (we assume that a prover on input a bad witness aborts as well).

Notice that, in the definition above, S_1 is required to simulate only valid proofs. This is realized by having S' access the witness for the given statement and executing S_1 only if the witness is valid. Thus, S_1 *does not* know the witness and has to simulate a valid proof from x only. This approach assumes that (as previously specified in [68,55]), for the case in which the statements are adaptively chosen by the adversary, the latter has to supply a witness to the prover (and thus to S'). This is because the prover here is restricted to probabilistic polynomial time, and then may not be able to generate witnesses by himself. As argued by Sahai [68], this seems to capture the correct notion of unbounded zero knowledge. In particular, this requirement does not allow A to test membership in L. This is because, intuitively, the adversary can produce valid witnesses and test their truthfulness only for statements of its own choice.

Definition 4.2 (*Unbounded Simulation-Sound Zero-Knowledge Proofs*). Let L be an $\mathcal{NP}$-language and $\mathtt{G}$, P, V, S_0, S_1 be probabilistic polynomial-time algorithms. We say that $\Pi = (\mathtt{G}, P, V, S = (S_0, S_1))$ is an unbounded simulation-sound zero-knowledge proof system in the common reference string model for L, if there exists a negligible function ν such that for all k it holds that:

- **unbounded concurrent zero knowledge:** Π is an unbounded concurrent zero-knowledge proof system for L;
- **unbounded simulation soundness:** for any polynomial-time adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$, where $\mathcal{A}_0$ and $\mathcal{A}_1$ are coordinated there exists a negligible function ν such that $\mathtt{Prob}(\mathtt{Expt}_{\mathcal{A}}(k) = 1) \le \nu(k)$ where the experiment $\mathtt{Expt}_{\mathcal{A}}(k)$ is defined as follows:

```
$\mathtt{Expt}_{\mathcal{A}}(k)$:
  $(\mathtt{crs}, \mathtt{aux}) \leftarrow S_0(1^k)$
  $(x, tr, b) \leftarrow (\langle \overline{S_1(\mathtt{aux})}, \mathcal{A}_0 \rangle, \langle \mathcal{A}_1, V \rangle)_{\mathtt{crs}}$
  Let $Q$ be the set of the transcripts of $\overline{S_1(\mathtt{aux})}$
  return 1 if and only if $b = 1$, $x \notin L$ and for all $tr' \in Q$ $tr \not\triangleleft tr'$.
```

Notice that since S'' does not check whether there is a witness for $x \in L$. S_1 potentially simulates both true and false proofs.

4.3. Concurrent zero-knowledge proof systems

In [29], three-round concurrent zero-knowledge arguments in the common reference string model are presented. More precisely Damgård [29] presents a general protocol based on the existence of one-way functions only and an efficient implementation based on number-theoretic assumptions. In this section we improve on this result by showing (constructively) the existence of three-round concurrent zero-knowledge proof (in contrast to argument) systems in the common reference string model. The first construction needs the sole assumption that one-way functions exist, while the second, more efficient, construction relies on the decisional Diffie–Hellman assumption. Interestingly both constructions hold in the shared random string model as well.

In our construction we consider unbounded black-box zero-knowledge proof systems with a non-rewinding simulator (which, consequently, is also concurrent zero knowledge).

Theorem 4.3. *If one-way functions exist, there exists a three-round concurrent zero-knowledge proof system in the common reference string model for any $\mathcal{NP}$ language.*

Proof. Consider the language for graph Hamiltonicity and the Σ-protocol by Blum [7] based on the existence of one-way functions (see Section B). We consider the following common reference string $\mathtt{crs} = \mathtt{crs}_0 \circ \mathtt{crs}_1$ where $\mathtt{crs}_0$ corresponds to the first message of Naor's commitment scheme while $\mathtt{crs}_1$ corresponds to the commitment parameters of a hybrid trapdoor commitment scheme HTCS= $(\mathtt{HGen}, \mathtt{HTGen}, \mathtt{HCom}, \mathtt{HTCom}, \mathtt{HTDec}, \mathtt{HVer})$ based on the existence of one-way functions only.

The prover P runs the prover algorithm of Blum's Σ-protocol to obtain the first message a. Then P uses HTCS (and in particular $\mathtt{HCom}$) to commit to a, i.e., it computes $(\mathtt{com}, \mathtt{dec})$ – that are, respectively, the commitment and decommitment of a – and it sends $\mathtt{com}$ to the verifier V.

V replies with a challenge c (as in Blum's Σ-protocol).

P runs again the prover algorithm of Blum's Σ-protocol to compute the last message z and sends $(a, \mathtt{dec}, z)$ to the verifier. Finally, if $\mathtt{HVer}(\mathtt{crs}_1, \mathtt{com}, \mathtt{dec}, a) = 1$ and $(\mathtt{crs}_0, a, c, z)$ is an accepting transcript (for Blum's Σ-protocol), the verifier accepts the proof.

Completeness is straightforward.

Soundness follows from the fact that when the prover and the verifier interact, the reference string specifies an unconditionally binding commitment scheme, therefore the proof of soundness proceeds exactly as the one for Blum's Σ-protocol that holds against any unbounded adversarial prover.

To prove that the protocol is concurrent zero knowledge we describe a simulator $S = (S_0, S_1)$ that works as follows. S_0 computes $\mathtt{crs}' = \mathtt{crs}'_0 \circ \mathtt{crs}'_1$ where $\mathtt{crs}'_0$ is the output of algorithm $\mathtt{receiver}$ of Naor's commitment scheme and $\mathtt{crs}'_1$ is computed by running algorithm $\mathtt{HTGen}$. Recall that $\mathtt{HTGen}$ returns a trapdoor $\mathtt{aux}$ as auxiliary output. This trapdoor will be used later by S_1. Note that, in this case, the string $\mathtt{crs}'_1$ specifies a trapdoor commitment scheme.

Once the common reference string is generated, S_1 runs the first round of the prover algorithm of Blum's Σ-protocol[2] and computes a. Next S_1 runs $\mathtt{HTCom}$ (with commitment parameters $\mathtt{crs}'_1$ and trapdoor $\mathtt{aux}$) and obtains back the pair $(\mathtt{com}, \mathtt{dec})$ corresponding to the committing and decommitting keys of a commitment of junk bits. S_1 proceeds by sending $\mathtt{com}$ to the verifier $V^\star$ and waits for $V^\star$ to send a challenge c'.

Once c' is received, S_1 runs the simulator corresponding to the special honest-verifier zero-knowledge property of Blum's Σ-protocol, on input $\mathtt{crs}'_0$ and c'. S_1 obtains a pair a', z' such that $(\mathtt{crs}'_0, a', c', z')$ is an accepting transcript for the verifier of Blum's protocol. S_1 runs algorithm $\mathtt{HTDec}$ by using the trapdoor $\mathtt{aux}$ in order to open $\mathtt{com}$ as a' and outputs decommitting key $\mathtt{dec}'$. Finally S_1 sends to $V^\star$ the last message $(a', \mathtt{dec}', z')$.

We can now focus on the difference between the concurrent proofs given by the simulator with respect to the ones given by the honest prover. We consider the following games.

Game G_0: this game corresponds to an execution of the honest prover.
Game G_1: the only difference from game G_0 is that here the reference string is generated using the simulator.
Game G_2: the only difference from game G_1 is that now the prover uses the cheating committing and decommitting algorithms $\mathtt{HTCom}$, $\mathtt{HTDec}$ instead of the honest one $\mathtt{HCom}$.
Game G_3: this game corresponds to an execution of the simulator and the the only difference from game G_2 is that now the prover uses the simulator of the special honest-verifier zero-knowledge property.

Standard hybrid arguments can be used to show that the two distributions are computationally indistinguishable. First of all, game G_0 and game G_1 are not distinguishable since otherwise the hybridness property of the hybrid trapdoor commitment scheme is contradicted. Game G_1 and game G_2 are not distinguishable since otherwise the trapdoorness of the trapdoor commitment scheme that is part of the hybrid trapdoor commitment scheme can be easily contradicted (notice that this holds even in the presence of any polynomial number of commitments, moreover we proved the indistinguishability of our commitments even with respect to adaptive adversarial receivers). Finally, game G_2 and game G_3 are not distinguishable since the special honest verifier zero-knowledge property of the Blum Σ-protocol works for any possible challenge (thus even for adaptively chosen challenges) and is known to be preserved even in the case of concurrent executions. □

Notice that the reference string used in the proof of Theorem 4.3 is uniformly distributed. Thus we have the following corollary.

Corollary 4.4. *If one-way functions exist, there exists a three-round concurrent zero-knowledge proof system in the shared random string model for any $\mathcal{NP}$ language.*

Theorem 4.5. *Given an $\mathcal{NP}$ language L that admits an efficient Σ-protocol, then under the DDH assumption there exists a three-round concurrent zero-knowledge proof system in the common reference string model for L.*

Proof. The proof follows the one of Theorem 4.3, however in this case we can use the efficient DDH-based hybrid trapdoor commitment scheme described in Theorem 3.2 and an unconditionally binding commitment scheme based on the DDH assumption (e.g., the one proposed in [56]). □

[2] In this round the prover does not use the witness.

Notice that the reference string used in the proof of Theorem 4.5 is uniformly distributed. Thus we have the following corollary.

Corollary 4.6. *Given an $\mathcal{NP}$ language L that admits an efficient Σ-protocol, then under the DDH assumption there exists a three-round concurrent zero-knowledge proof system in the shared random string model for L.*

4.4. Simulation-sound zero knowledge

The notion of *simulation soundness* has been used for the design of many secure cryptographic primitives (see for instance [68,62]). Informally, a proof system is simulation sound if an adversary that plays the role of verifier when the proofs are simulated for both true and false instances, is not able to play as a prover another session of the protocol in which he convinces an honest verifier of a false statement.

In [55], MacKenzie and Yang proposed three-round unbounded simulation-sound zero-knowledge argument systems in the common reference string model, in particular their arguments use simulation-sound trapdoor commitment schemes, therefore they obtain efficient argument systems based on the security of DSA [58] or the Cramer–Shoup [25] signature schemes and argument systems based on the existence of one-way functions.

As already discussed in Section 1.1, the multi-trapdoor commitments presented in [44] allow for more efficient constructions of unbounded simulation-sound zero-knowledge argument systems.

In this section we extend their results by showing the existence (constructively) of three-round unbounded simulation-sound zero-knowledge proof (in contrast to argument) systems in the common reference string model. We can achieve this result either by using hybrid simulation-sound trapdoor commitments instead of non-hybrid simulation-sound trapdoor commitments in the construction of [55] or by using hybrid multi-trapdoor commitments instead of non-hybrid multi-trapdoor commitments in the construction of [44].

For each of these two results we give a first construction that needs the sole assumption that one-way functions exist. Then we give a more efficient second construction that requires (on top of the assumptions described for the efficient constructions of [55] and [44]) the decisional Diffie–Hellman assumption.

Theorem 4.7. *If one-way functions exist, there exists a three-round unbounded simulation-sound zero-knowledge proof system in the common reference string model for any $\mathcal{NP}$ language.*

Proof. Given any one-way function, we know by Theorem 3.6 that there exists (constructively) a hybrid simulation-sound trapdoor commitment scheme HSSTCS. The basic idea of this proof is then to plug HSSTCS (rather than an SSTC) in the construction of [55] (see Fig. 2 of [54]). Now we show that this allows us to obtain a three-round unbounded simulation-sound zero-knowledge *proof* system in the common reference string model (rather than an argument as in [55]).

Completeness. This property follows directly by inspection.

(Classical) soundness. The first part of the common reference string is generated by $\mathtt{HGen}$, thus when the prover P runs the protocol to compute a commitment of a (where, again, a is the first message of the underlying Σ-protocol) he actually uses an unconditionally binding commitment scheme. This means that even an unbounded prover can not later open the commitment with a message $a' \neq a$. Indeed suppose that $P^\star$ succeeds in proving a false statement with some non-negligible probability p. Let (a, c, z) be the transcript of the corresponding Σ-protocol and let $\mathtt{com}_a$ be the commitment that has been opened as a commitment of a. Now since we are dealing with a Σ-protocol we can run an extractor that, having black-box access to $P^\star$, obtains another accepting transcript (a', c', z') which is still valid with respect to the commitment $\mathtt{com}_a$. Notice that such an extractor succeeds in polynomial time because p is non-negligible and the challenges are chosen uniformly and at random. Thus the statement being false, it has to be the case that $a' \neq a$ otherwise, by the special-soundness property of the Σ-protocol, we would obtain a witness for a false statement (and this is a contradiction). However, the fact that $a' \neq a$ and both correspond to the same commitment $\mathtt{com}_a$ violates the unconditional binding of the commitment scheme instantiated by $\mathtt{HGen}$.

Unbounded concurrent zero knowledge. The simulator uses $\mathtt{HTGen}$ to generate the reference string and uses the corresponding trapdoor to generate a straight-line simulation. The output of the simulator (that runs only on input true statements) and the output of a real interaction between a prover and an adversarial verifier $V^{\star}$ have the following differences:

1. the reference string is generated by HTGen instead of HGen;
2. the simulator uses HTCom and HTDec instead of HCom;
3. the simulator uses the special honest-verifier zero-knowledge property of the Σ-protocol instead of the prover algorithm of the Σ-protocol.

Suppose that an adversarial verifier $\mathcal{A} = V^{\star}$ outputs 1 with probability p_R after a real interaction but outputs 1 with probability p_S after a simulated interaction. We show that there exists a negligible function ν_0 such that $|p_R - p_S| \leq \nu_0(k)$. As discussed above, in the real game, algorithm HGen generates the commitment parameters and $\mathcal{A}$ interacts with P, while in the simulated game algorithm HTGen generates the commitment parameters and $\mathcal{A}$ interacts with a simulator. Consider the following hybrid games.

Game 0: the commitment parameters are generated by HTGen, $\mathcal{A}_0$ interacts with an algorithm H_0 that runs the honest sender protocol of the commitment scheme and the honest prover algorithm of the Σ-protocol.
Game 1: the commitment parameters are generated by HTGen, $\mathcal{A}_0$ interacts with an algorithm H_1 that uses HTCom and HTDec to compute and open commitments and runs the honest prover algorithm of the Σ-protocol.

Let p_0 be the probability that $\mathcal{A}$ outputs 1 at the end of the first hybrid game and p_1 be the probability that $\mathcal{A}$ outputs 1 at the end of the second hybrid game.

Since the only difference between the real game and the first hybrid game consists of the algorithm that generates the commitment parameters we have that there must be a negligible function ν_1 such that $|p_0 - p_R| < \nu_1(k)$, otherwise $\mathcal{A}$ can be used to break the hybridness property of the hybrid simulation-sound trapdoor commitment scheme.

Since the only difference between the two hybrid games is the use of HCom in the first hybrid game and of TCom and TDec in the second one, we have that there must be a negligible function ν_2 such that $|p_0 - p_1| \leq \nu_2(k)$, otherwise $\mathcal{A}$ can be used to break the trapdoorness of the hybrid trapdoor commitment scheme.

Finally, it must be the case that $|p_1 - p_S| \leq \nu_3(k)$ for some non-negligible function ν_3, otherwise $\mathcal{A}$ can be used to distinguish the second hybrid game from the simulated game. The only difference between these two games is the use of the honest prover algorithm of the Σ-protocol in the second hybrid game and of the simulator of the honest-verifier zero-knowledge property of the Σ-protocol in the simulated one. $\mathcal{A}$ is therefore a distinguisher between a transcript of the honest prover of the Σ-protocol and the output of the honest-verifier zero-knowledge protocol. This contradicts the special honest-verifier zero-knowledge property of the Σ-protocol.

From the previous discussion we have that there exists a negligible function ν_0 such that $|p_R - p_S| \leq \nu_0(k)$.

Unbounded simulation soundness. To prove unbounded simulation-soundness we proceed as follows. Suppose that an adversary $\mathcal{A}$ succeeds in proving a false statement after having seen simulated proofs (we stress that in this case the reference string is generated by HTGen as discussed in the proof of the unbounded concurrent zero-knowledge property). We show that the ability of $\mathcal{A}$ to produce a valid proof on a false statement can be used to violate the simulation soundness of a hybrid simulation-sound trapdoor commitment scheme (HGen, HTGen, HCom, HTCom, HDec, HVer), i.e., the simulation-sound binding property of (HTGen,HCom, HTCom,HDec, HVer).

More precisely, we show an algorithm $\mathcal{A}'$ that plays the role of sender in the game described above for the simulation-sound binding property of Definition 2.7 and, by black-box accessing to $\mathcal{A}$, it either breaks the simulation sound binding property of the commitment scheme or it breaks the unforgeability of the underlying one-time signature scheme (see [54]).

The proof closely follows that of Theorem 4.1 in [54].

$\mathcal{A}'$ receives as challenge the commitment parameters of a simulation-sound trapdoor commitment scheme generated by HTGen. The polynomial-time adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ interacts with $\mathcal{A}'$ for obtaining accepting proofs of both true and false statements. More specifically, in the simulated game, $\mathcal{A}'$ completes his proofs by asking to an oracle to equivocate some commitments with respect to his public keys for the one-time secure signature scheme (that correspond to different tags for the commitment scheme). If $\mathcal{A}$ succeeds in proving a false statement by using a public key already used by $\mathcal{A}'$, then $\mathcal{A}$ can be used to break the security of the one-time secure signature scheme. This is because there actually exist at least two different transcripts (one of $\mathcal{A}'$ and one of $\mathcal{A}$) with correct signatures with respect to the same public key. If instead $\mathcal{A}$ uses a new public key then the proof proceeds very similarly to the one of classical soundness. Indeed, by using the extractor of the Σ-protocol, with non-negligible probability $\mathcal{A}$ will complete again the protocol and from the two transcripts it is possible to extract two valid and different decommitments with respect to the same commitment (and of course with respect to the same tag). This case therefore violates the simulation-sound binding property of the commitment scheme since for this tag (that corresponds to the new public key) the equivocation oracle has not been used.

Note that the transcript (i.e., the distribution of the reference string and the messages exchanged) of the simulation performed by $\mathcal{A}'$ that we have described above, is perfectly indistinguishable with respect to the one described in $\texttt{Expt}_{\mathcal{A}}(k)$ of Definition 4.2. Therefore, we have shown that $\texttt{Prob}(\texttt{Expt}_{\mathcal{A}}(k) = 1) \leq \nu(k)$ and the claim holds.

Complexity-theoretic assumptions. To conclude this proof we remark here that when using the variation of the Σ-protocol of Blum discussed in Appendix B, we have that the Σ-protocol exists under the sole assumption that one-way functions exist. Moreover, it is well known that one-time secure signature scheme can be constructed from any one-way function [67]. □

Theorem 4.8. *Given an $\mathcal{NP}$-language L that admits an efficient Σ-protocol, then under the assumption that DSA is a secure signature scheme and that the DDH assumption holds (or that the Cramer–Shoup signature scheme is secure and that the DDH assumption holds) there exists an efficient three-round unbounded simulation-sound zero-knowledge proof system in the common reference string model for L.*

Proof. The proof follows the one of Theorem 4.7, however in this case we use the efficient hybrid simulation-sound trapdoor commitment scheme that we have shown to exist in Theorem 3.7 under the same number-theoretic assumptions as considered in the statement of this theorem. □

Hybrid multi-trapdoor commitments for simulation-sound zero-knowledge proofs. The approach used to achieve simulation-sound zero-knowledge proofs from hybrid simulation-sound trapdoor commitments by using the construction of [55] is quite general and can be used for simulation-sound zero-knowledge proofs from hybrid multi-trapdoor commitments by using the construction of [44]. In the proof of Theorem 4.7 we have shown that by replacing a (non-hybrid) simulation-sound trapdoor commitment scheme with a hybrid one, then an unbounded simulation-sound zero-knowledge argument can be transformed into an unbounded simulation-sound zero-knowledge proof. This same approach can be used for the case of (non-hybrid) multi-trapdoor commitment schemes and their application to unbounded simulation-sound zero-knowledge arguments [44] (we stress that in [44] such a notion is referred to as left-concurrent non-malleable zero-knowledge argument). Since the improvement is based on the same techniques as used for Theorem 4.7, we only discuss this case informally.

Consider the unbounded simulation-sound zero-knowledge argument of [44]. We require that the unbounded adversarial prover runs the protocol by using some unconditionally binding commitment scheme such as the one defined by the commitment parameters generated by algorithm $\texttt{HGen}$ of a hybrid multi-trapdoor commitment scheme HMTCS. As a consequence, the protocol resorts to the properties of the Σ-protocol and by using the variation of Blum's Σ-protocol presented in Appendix B we have that in the common reference string it is a proof (in contrast to argument) and it can be based on the existence of one-way functions only. The simulator S for the zero-knowledge property uses $\texttt{HTGen}$ to generate the commitment parameters and uses the auxiliary information to equivocate the commitments, precisely as discussed in the proof of Theorem 4.7 (i.e., the considered commitment scheme is a multi-trapdoor commitment scheme instead of an unconditionally binding one).

Suppose that the commitment parameters are established by S, and a polynomial-time adversary $\mathcal{A}$ that mounts a man-in-the-middle attack that violates the simulation-sound property. It is easy to verify that, in such a case, the proof strictly follows the one of [44] and it can be shown that the existence of $\mathcal{A}$ can be used to reach a contradiction (actually in this scheme we have as additional assumption the existence of a family of collision-resistant hash functions) or to show that a polynomial-time algorithm $\mathcal{A}'$ that has oracle access to $\mathcal{A}$ can give the same argument system as given by $\mathcal{A}$ without mounting the man-in-the-middle attack.

The following theorem follows from the above informal discussion and under the number-theoretic assumptions discussed in Appendix A.

Theorem 4.9. *Given an $\mathcal{NP}$-language L that admits an efficient Σ-protocol, if collision-resistant hash functions exist then under the sRSA and the DDH assumption or the sDH and the DDH assumptions there exists an* *efficient three-round unbounded simulation-sound zero-knowledge proof system in the common reference string model for L.*

5. Conclusions

In this paper we have motivated and formalized the notion of hybrid trapdoor commitment. We have shown different constructions based on the sole assumption that one-way functions exist and very efficient constructions under number-theoretic assumptions.

For the main result of this paper, we have crucially used these commitments in order to achieve strong notions of zero knowledge proofs (in contrast to arguments) that can be implemented either under weak complexity-theoretic assumptions or efficiently under number-theoretic assumptions.

More recently, in [21,19] hybrid trapdoor commitments have been shown to imply mercurial commitments [22] and thus are a building block for the construction of zero-knowledge sets [53].

Acknowledgments

The work described in this paper has been supported in part by the European Commission through the IST Program under Contract IST-2002-507932 ECRYPT. The work of the second author is also supported in part through the FP6 program under contract FP6-1596 AEOLUS. The authors would like to thank the anonymous reviewer of Theoretical Computer Science for many useful comments and suggestions.

The first author's work was entirely done while a CNRS researcher at the Département d'Informatique de l'Ecole Normale Supérieure, Paris, France. The second author's work was partially done while he was a post-doctoral fellow at the Département d'Informatique of the Ecole Normale Supérieure in Paris, France.

Appendix A. Some number theoretic assumptions

A.1. The strong RSA assumption

Let $N = pq$ be the product of two primes. We denote by $\phi(N)$ the Euler function of N; it is a well known fact from number theory that $\phi(N) = (p-1)(q-1)$. Let e be an integer relatively prime to $\phi(N)$, the RSA assumption [66] states that, given a random element $y \in Z_N^*$, it is computationally infeasible to compute the unique $x \in Z_N^*$ such that $x^e \equiv y \bmod N$. The strong RSA assumption, introduced in [3], states that given a random element y as above, it remains computationally infeasible to find a couple (x, e), with $e \neq 1$ such that $x^e \equiv y \bmod N$. Thus the difference, with respect to the standard RSA assumption, is that here the adversary is allowed to choose the exponent e for which she will be able to compute the required root. More formally, denoting with $RSA(k)$ the set of composite moduli, N such that N is the product of two $k/2$-bit primes, we state that

Assumption A.1. We say that the strong RSA assumption holds if, for all probabilistic polynomial time adversaries $\mathcal{A}$ and for sufficiently large k one has

$$Pr[N \leftarrow RSA(k);\ y \leftarrow Z_N^*;\ (x, e) \leftarrow \mathcal{A}(N, y) :\ x^e \equiv y \bmod N]$$

is negligible in k.

A.2. The strong Diffie–Hellman assumption

Let G be a cyclic group of order q and let g be a generator of G. The strong DH assumption, introduced by [9], states that no polynomially bounded adversary, on input $G, g, g^y, g^{y^2}, \ldots, g^{y^\ell}$, for some random secret $y \in Z_q$, should be able to produce a couple (x, e) such that $x^{y+e} = g$ in G. More formally

Assumption A.2. Let G be a cyclic group of prime order q, where $|q| = k$. We say that the strong $\ell - DH$ assumption holds in G if for all probabilistic polynomial time adversaries $\mathcal{A}$ and for sufficiently large k one has that

$$Pr[y \leftarrow Z_q;\ (x \in G,\ e \in Z_q) \leftarrow \mathcal{A}(G, g, g^y, g^{y^2}, \ldots, g^{y^\ell}) :\ x^{y+e} = g]$$

is negligible in k.

Appendix B. Tools

In this section we discuss some tools that we use in our constructions.

Naor's commitment scheme. Given a pseudorandom generator G, the bit commitment scheme proposed by Naor [57] works as follows. The receiver sends a $3k$-bit string r to the sender. The sender randomly selects a k-bit string s and commits to 0 by sending $G(s)$, instead he commits to 1 by sending $G(s) \oplus r$. Finally the sender opens the committed bit by sending s. Naor's scheme has the following properties:

- **Assumptions:** it is based on the existence of pseudorandom generators; it is known how to construct them on the assumptions that one-way functions exist [47].
- **Round complexity:** there exists a set-up stage in which the receiver sends one message, then the sender has to play only one round for each commitment.
- **Security:** it is an unconditionally binding commitment scheme.
- **Message distribution:** both the set-up message of the receiver and the commitments computed by the sender are indistinguishable from random strings for polynomial-time algorithms.

In the reference string model (in particular in the shared random string model) Naor's commitment scheme can be implemented as a non-interactive commitment scheme by replacing the set-up message sent by the receiver with a piece of the reference string. Obviously, the bit commitment scheme can be extended to a string commitment scheme by simply iterating the described scheme for each bit of the string.

Σ-protocols. A Σ-protocol is a three-round interactive protocol between two probabilistic algorithms, an honest prover P and an honest verifier V. The algorithms receive as common input a statement "$x \in L$". P has as auxiliary input a witness w such that $(x, w) \in R_L$ where L is an $\mathcal{NP}$-language and x belongs to L. At the end of the protocol V decides whether the transcript is accepting with respect to the statement or not. Since a cheating prover $P^\star$ can interact with a verifier on input a false statement "$x \in L$", we denote by $\widetilde{L}$ the language of the instances (both true and false) that can be used as common input with a verifier. Obviously it holds that $L \subseteq \widetilde{L}$.

The Σ-protocols we consider in this paper have the following properties:

- **Public coin:** V sends random bits only.
- **Special soundness:** let (a, c, z) and (a, c', z') be two accepting transcripts for a statement "$x \in L$". If $c \neq c'$ then $x \in L$ (i.e., the statement is true) and there exists an efficient algorithm E, referred to as extractor, that on input (x, a, c, z, c', z') outputs w such that $(x, w) \in R_L$. Therefore we consider proofs (in contrast to arguments).
- **Special honest-verifier zero knowledge:** there is an efficient algorithm S, referred to as simulator, that on input a true statement "$x \in L$" outputs for any c a pair (a, z) such that the triple (a, c, z) is indistinguishable from the transcript of a conversation between P and V. Moreover, the triple (a, c, z) generated so far is an accepting transcript independently of the truthfulness of the statement[3] (i.e., it holds for any $x \in \widetilde{L}$).

As a consequence of the properties described above, if $x \in \widetilde{L} \setminus L$ then for any first message a there is only one pair (c, z) such that (a, c, z) is an accepting transcript, therefore when the prover and the verifier run the protocol as input a false statement, a can be used as a commitment and z as a decommitment of a commitment of string c.

There are in the literature many Σ-protocols with these properties, most notably, the protocol of Blum [7] that is a Σ-protocol for the $\mathcal{NP}$-complete language Hamiltonicity, the protocol of Schnorr [69] for proving knowledge of a discrete logarithm and its variants (e.g., equality of discrete logarithms and other compositions by means of Boolean formulae [24]).

In general we assume that the prover computes message z by using some auxiliary information generated during the computation of message a.

Blum's Σ-protocol for Hamiltonicity. In [7] Blum presented a Σ-protocol for the $\mathcal{NP}$-complete language Hamiltonicity. A variation of Blum's Σ-protocol is instantiated on any one-way function by means of Naor's commitment scheme in the reference string model. This is achieved by first requiring that the reference string contains

[3] This property is also required by the Σ protocols of [55].

a string setup that corresponds to the message played by the receiver of Naor's commitment scheme. Secondly, for each proof the parties play the three rounds that we denote (a, c, z) of the original Blum's Σ-protocol with the following variation: the commitments encoded in message a are computed by using the second round of Naor's commitment scheme.

Auxiliary language. Given a commitment com and a message m, it is possible to reduce the decisional problem "is com a commitment of a commitment of string m ?" to the decisional problem "is G a Hamiltonian graph ?" by means of a general NP-reduction [23]. Therefore, we will assume that it is possible to use the previously discussed implementation of Blum's Σ-protocol to prove that a given commitment corresponds to a given string under the sole assumption that one-way functions exist.

Σ-protocol for equality of two discrete logarithms. An efficient Σ-protocol for proving the equality of two discrete logarithms can be achieved by means of an AND-composition of the Σ-protocol of Schnorr [69] for proving knowledge of a discrete logarithm, in particular the prover uses the same random tape in the two cases (and the same witness). More precisely, on input two generators $g_1, g_2 \in G$, and two values $h_1, h_2 \in G$ one proves knowledge of $x = \log_{g_1} h_1 = \log_{g_2} h_2$, by running two instances of the Schnorr protocol in parallel, using a common random choice, a common challenge and a common response. For details, see [13].

Appendix C. Additional efficient implementations of hybrid trapdoor commitments

In this section we described two simple and efficient constructions of hybrid trapdoor commitments that achieve also the extractability property. The constructions are based on the hardness of factoring and inverting RSA$[N, N]$ (i.e., RSA where the public exponent is set to N). In the following we focus on the basic notion of trapdoor commitment, stronger properties (such as simulation soundness) can be achieved straightforwardly by applying the techniques discussed in Section 3.

First, however we discuss an intractability assumption that is going to be used in our constructions.

Let N be the product of two $k/2$-bit safe primes (i.e., primes of the form $p = 2p' + 1$ where p' is prime). Let G be the group of elements of order $(\lambda(N)/2)N$ in $Z^*_{N^2}$ and G' the group of elements of order $\lambda(N)/2$ in $Z^*_{N^2}$. Now assume an element h is chosen uniformly and at random in G or in G' according to some random bit b. We consider the following (decisional) problem. Given on input h and N (but not the factorization of the latter) guess b with probability significantly better than $1/2$. In what follows we will refer to this problem as to the *Decisional Subgroup Composite Residuosity* problem (DSCR for brevity) and, for sufficiently large k, we assume it to be intractable.

The reader may have noticed that the conjecture sketched above looks very similar to the Decisional Composite Residuosity (DCR) assumption suggested by Paillier [60]. Informally the DCR assumption states that, for sufficiently large k, it is infeasible to distinguish N residues modulo N^2 (i.e., elements of the form $z = x^N \bmod N^2$) from random elements in $Z^*_{N^2}$. Indeed, for the case of moduli obtained as the product of safe primes, the following lemma shows that the two assumptions are actually equivalent.

Lemma C.1. *Let $N = pq$ be the product of two safe primes, then DCR is intractable if and only if DSCR is intractable.*

Proof. First observe that any adversary solving the DCR problem can be trivially used to break the DSCR problem. Consequently we focus here on proving that the inverse direction holds as well.

Let $\mathcal{A}$ be an adversary for the DSCR problem, our goal is to construct an adversary $\mathcal{A}'$ which receives on input a random challenge for the DCR problem and "uses" $\mathcal{A}$ to solve it. In particular $\mathcal{A}'$ receives on input h, N and has to determine if it is an N-th residue or not. As a first step $\mathcal{A}'$ computes $c = h^2 \bmod N^2$ and if $c = 1$ it outputs yes. If $c \neq 1$, $\mathcal{A}'$ gives it to $\mathcal{A}$ and outputs whatever A outputs.

Let us denote by QR_{N^2} the group of quadratic residues modulo N^2 and with QNR_{N^2} the group of quadratic residues which are also N-th residues in $Z^*_{N^2}$. Now observe that c is a random element either in QR_{N^2} or in QNR_{N^2}. In the first case the probability that $c \in G$ is overwhelming (roughly $1 - 1/\sqrt{N}$). Similarly, if $c \in QNR_{N^2}$, the probability that $c \in G'$ is overwhelming (roughly $1 - 1/\sqrt{\lambda(N)}$). Thus, skipping some simple details, it holds that

$$Prob[\mathcal{A}' \text{ succeeds}] \approx Prob[\mathcal{A} \text{ succeeds}]. \quad \square$$

C.1. First construction

In this section we present a factoring-based construction which uses as a basic building block a trapdoor commitment scheme proposed by Bresson et al. [12]. In the following we describe in detail the algorithms HGen, HTGen, HCom, HTCom, HTDec, HVer.

Key generation. The algorithm HTGen goes as follows. It takes as input the quantity 1^k, where k is the usual security parameter, and randomly chooses two random $k/2$-bit *safe* primes p, q and sets $N = pq$. Next, denoting with G the cyclic group of quadratic residues modulo N^2, it chooses uniformly and at random an element h of maximal order in G. Finally it outputs $\mathtt{crs} = \langle h, N\rangle$ and $\mathtt{aux} = \langle p, q\rangle$.

The algorithm HGen is very similar to HTGen: the only difference is that here the algorithm chooses uniformly and at random an element h' in the subgroup G' – of G – of elements having order $\lambda(N)/2$. The output of $\mathtt{HGen}(1^k)$ is then $\mathtt{crs}' = \langle h', N\rangle$.

Notice that, assuming the DCR assumption holds, the two distributions according to which crs and crs′ are generated are computationally indistinguishable.

Committing to a message. The algorithms HCom and HTCom are actually identical. To commit to a message $m \in Z_N$ the sender chooses $r \in_R Z_{N^2}$ and sets

$$(\mathtt{com} = h^r(1 + mN) \bmod N^2, \mathtt{dec} = r) = \mathtt{HCom}(\mathtt{crs}', m).$$

Algorithm HTCom has on input crs and aux instead of crs′ and m and performs the same computation of HCom by using a randomly chosen m'.

As already proved by Bresson et al. [12], under the assumption that factoring large safe-prime moduli is hard, HTGen, HTCom, HTDec, HVer is a trapdoor commitment scheme.

It remains to show that HGen, HCom, HVer is an unconditionally binding commitment scheme. First we prove that, for any (m', r'), (m, r) such that

$$h^{r'}(1 + m'N) \bmod N^2 = h^r(1 + mN) \bmod N^2$$

it holds that $m = m' \bmod N^2$. To see this observe that the equation above can be rewritten as

$$h^{r'-r}(1 + (m' - m)N) = 1 \bmod N^2. \tag{1}$$

Now notice that h has order $\lambda(N)/2$ while $1 + (m' - m)N$ has order N. Thus the only possibility for equation 1 to be true is to have that $r = r' \bmod \lambda(N)/2$ and $m = m' \bmod N$ as required.

The hiding property, on the other hand, is implied by the DCR assumption.

C.2. An alternative construction based on RSA[N, N]

Now we present a solution based on the hardness of inverting the RSA[N, N] function. This construction builds on (trapdoor) commitment scheme originally proposed by Catalano et al. [18]. Again, we describe in detail the algorithms HGen, HTGen, HCom, HTCom, HTDec, HVer.

Key generation. The algorithm HTGen is as follows. It takes as input the quantity 1^k, where k is the security parameter, and randomly chooses two random $k/2$-bit *safe* primes p, q and sets $N = pq$. Next it chooses a random element $h \in Z^*_{N^2}$ and outputs $\mathtt{crs} = \langle h, N\rangle$ and $\mathtt{aux} = \langle p, q\rangle$.

The algorithm HGen looks very similar to HTGen: as before the only difference is in the choice of h. In this case the algorithm chooses, uniformly and at random, an N-th residue h' in $Z^*_{N^2}$. The output of $\mathtt{HGen}(1^k)$ is then $\mathtt{crs}' = \langle h', N\rangle$.

Once again, under the DCR assumption, the two distributions according to which crs and crs′ are generated are computationally indistinguishable.

Committing to a message. Similarly to the construction presented in the previous section, the algorithms HCom and HTCom are actually the same. To commit to a message $m \in Z_N$ the sender chooses $r \in_R Z^*_N$, $s \in Z_N$ and sets

$$(\mathtt{com} = (1 + mN)r^N h^s \bmod N^2, \mathtt{dec} = (r, s)) = \mathtt{HCom}(\mathtt{crs}, m).$$

Algorithm HTCom has on input crs and aux instead of $\mathtt{crs}'$ and m and performs the same computation of HCom by using a randomly chosen m'.

In [18] Catalano et al. proved that, under the assumption that inverting RSA with public exponent set to N is hard, HTGen, HTCom, HTDec, HVer is a trapdoor commitment scheme.

It remains to prove that HGen, HCom, HVer is an unconditionally binding commitment scheme. However this follows immediately from the fact that, when h is chosen as a random N-th residue in $Z^*_{N^2}$, our commitment scheme actually becomes Paillier's encryption scheme with public base $g = (1 + N)$.

Appendix D. Multi-trapdoor commitments and signature schemes

In this section we clarify the relationship between multi-trapdoor commitment schemes and signature schemes. In particular we show that multi-trapdoor commitment schemes are actually equivalent to signature schemes secure against generic chosen message attack (for brevity we will refer to this class of signature schemes as *weak* signatures). In a nutshell in a generic chosen message attack the adversary is allowed to obtain (from the signer S) valid signatures for a chosen list of messages $m_1 \dots m_\ell$ before trying to produce a forgery. More precisely, in this scenario, we allow the adversary to choose the messages but only in a generic way, meaning with this that the list of messages is chosen before S's public key is revealed (and thus the attack is generic in the sense that it does not depend on the specific public key).

We prove the equivalence between the two primitives by showing how to construct a multi-trapdoor commitment scheme from any weak signature scheme and vice versa.

D.1. Multi-trapdoor commitments based on weak signatures

Here we show how to construct a multi-trapdoor commitment scheme from any weak signature scheme. Our basic construction is not completely new in the sense that we adapt to our case a reduction method previously proposed by Canetti et al. [17] and by MacKenzie and Yang [55]. Thus before going into the details of the reduction we briefly discuss the specific tools we need.

1. We use the Naor [57] commitment scheme based on pseudorandom generators. Pseudorandom generators can be constructed from any one-way function [47]. One-way functions are equivalent to secure signatures (and therefore are also equivalent to weak signatures).
2. A parameter LPK_i is included. Intuitively LPK_i is the i-th (public) key for the multi-trapdoor commitment scheme we are about to construct.
3. As underlying $\mathcal{NP}$ language we consider the verification relation

   $$\{((\mathtt{VK}, LPK_i), \sigma_i) \mid \mathtt{Verify}((\mathtt{VK}, LPK_i), \sigma_i) = 1\}$$

   where VK is the verification key for the given weak signature scheme and Verify is the signature verification algorithm. Intuitively a local trapdoor for the scheme we have in mind is just a signature σ_i (with respect to the public key VK) of the local public key LPK_i.

The multi-trapdoor commitment scheme goes as follows. Gen (on input 1^k) generates a pair of matching signing and verification keys MSK and MVK for a weak signature scheme. Next it generates a set LS which is the space from which local public keys are sampled (in particular we assume that the statement "$x \in LS$" is efficiently verifiable). Gen outputs the public key $\mathtt{MPK} = (\mathtt{MVK}, LS)$.

The algorithm Sel on input MPK outputs some element in LS that we denote LPK_i.

Tkg on input LPK_i and MSK checks that $LPK_i \in LS$ and, if yes, produces a signature σ_i on LPK_i. Tkg outputs σ_i as local trapdoor.

To commit to a bit b, $\mathtt{Com}((\mathtt{MPK}, LPK_i), b)$ uses the $\mathcal{NP}$ relation

$$\{(\mathtt{MPK}, LPK_i) \mid \exists \sigma_i \ : \ \mathtt{Verify}((\mathtt{MPK}, LPK_i), \sigma_i) = 1\}$$

and produces an $\mathcal{NP}$-reduction from this relation to the Hamiltonicity relation for a graph G, having q nodes, so that finding a Hamiltonian cycle in G is equivalent to computing σ_i. Then it proceeds as follows.

- To commit to 0, pick a random permutation π of the graph G and commit to the entries of the adjacency matrix one by one, using the underlying commitment scheme based on pseudorandom generators. To decommit, just release π and decommit to every entry of the adjacency matrix.
- To commit to 1, choose a randomly labelled q-cycle in the graph and commit to all the entries in the adjacency matrix which correspond to edges of the q-cycle. For the remaining entries just provide random values (in the appropriate range). To decommit open only the entries corresponding to the randomly chosen q-cycle.

The algorithm $\mathtt{TCom}$ on input $((\mathtt{MPK}, \mathit{LPK}_i), \sigma_i)$ computes the graph G associated with $(\mathtt{MPK}, \mathit{LPK}_i)$ and using σ_i finds a Hamiltonian cycle $HG(G)$ in G. It then selects a random permutation π of the nodes of G and, using $\mathtt{Com}$, commits to all the entries of the adjacency matrix of $\pi(G)$ one by one. Finally it sets $\omega = (G, HG(G))$.

$\mathtt{TDec}$ on input ω a commitment $\mathtt{com}$ and a bit b goes as follows. If $b = 0$ then it decommits using the decommitment of zero. If $b = 1$ it decommits via the decommitment procedure for 1, using $HG(G)$ as underlying Hamiltonian cycle.

It remains to show that if we have an adversary $\mathcal{A}$ that can break the binding property of the scheme presented above then we can use this adversary to implement an adversary $\mathcal{B}$ (whose running time is polynomially related to that of $\mathcal{A}$) that breaks the weak signature scheme.

First of all $\mathcal{B}$ receives as input a message space $\mathcal{M}$. It sets $LS = \mathcal{M}$ and forwards this information to $\mathcal{A}$. $\mathcal{A}$ proceeds by choosing ℓ public keys (for some parameter ℓ) $pk_1 \dots pk_\ell$. Next $\mathcal{B}$ sets $m_1 = pk_1, \dots, m_\ell = pk_\ell$ as its own list of candidate messages to be signed. Note that once $\mathcal{B}$ receives the signatures $(\sigma_1, \dots, \sigma_\ell)$ for the selected messages, it can easily equivocate on every pk_i (for $i = 1, \dots, \ell$). This is because for each public key in the list it knows the corresponding (local) private key. Now assume $\mathcal{A}$ manages to output a triple $(\mathtt{com}, \mathtt{dec}, \mathtt{dec}')$ such that $\mathtt{Ver}((\mathtt{MPK}, pk), \mathtt{com}, \mathtt{dec}, m) = \mathtt{Ver}((\mathtt{MPK}, pk), \mathtt{com}, \mathtt{dec}', m') = 1$ where $m \neq m'$ and $pk \neq pk_i$ for $i = 1, \dots, \ell$. In such a case, however, $\mathcal{B}$ can use this double opening to extract a Hamiltonian cycle in G and thus a signature on the (previously unasked) message $m = pk$, which means that $\mathcal{B}$ can break the weak signature scheme.

D.2. Weak signatures from multi-trapdoor commitments

Here we show the converse direction, namely how to construct a weak signature scheme out of a multi-trapdoor commitment scheme. The basic idea of this construction is to set, as message space for the signature scheme, the public key space of the multi-trapdoor commitment scheme. Then to sign a message M one shows the ability to open a commitment with public key M to both zero and one. In detail the construction goes as follows. Let $\mathtt{MTC}$ be the given multi-trapdoor commitment scheme. We define $\mathtt{Sig}_{\mathtt{MTC}} = (\mathtt{SigGen}_{\mathtt{MTC}}, \mathtt{Sign}_{\mathtt{MTC}}, \mathtt{Verify}_{\mathtt{MTC}})$ as follows.

- On input 1^k, $\mathtt{SigGen}_{\mathtt{MTC}}$ runs $\mathtt{Gen}(1^k)$ and obtains back as output $(\mathtt{MPK}, \mathtt{MSK})$ together with a public key space LS. Then $\mathtt{SigGen}_{\mathtt{MTC}}$ sets $VK = \mathtt{MPK}$ and $SK = \mathtt{MSK}$ and $\mathcal{M} = LS$ (where by $\mathcal{M}$ we mean the message space).
- On input the signing key SK and a message m, $\mathtt{Sign}_{\mathtt{MTC}}$ first checks if $m \in \mathcal{M}$ and, if yes, it runs

  $$\mathtt{tk} \leftarrow \mathtt{Tkg}(SK, m), (C, \omega) \leftarrow \mathtt{TCom}(\mathtt{tk})$$

  and

  $$y_0 \leftarrow \mathtt{TDec}(\omega, 0) \quad y_1 \leftarrow \mathtt{TDec}(\omega, 1)$$

  The signature is then $\sigma = (C, y_0, y_1)$.
- To verify a signature for a message $m \in \mathcal{M}$, one just checks that the following equalities $\mathtt{Ver}(m, C, y_0, 0) = 1$ and $\mathtt{Ver}(m, C, y_1, 1) = 1$ are satisfied.

Theorem D.1. *Let* $\mathtt{MTC}$ *be a multi-trapdoor commitment scheme then* $\mathtt{Sig}_{\mathtt{MTC}}$ *is a signature scheme that is existentially unforgeable against generic chosen message attack.*

Proof. To prove the statement we assume there exists an adversary $\mathcal{A}$ for $\mathtt{Sig}_{\mathtt{MTC}}$ and then we show how to construct an attacker $\mathcal{B}$ which "uses" $\mathcal{A}$ to break the binding property of the underlying multi-trapdoor commitment scheme $\mathtt{MTC}$. When $\mathcal{B}$ receives the public key space LS for $\mathtt{MTC}$, it sets $\mathcal{M} = LS$ and forwards this information to $\mathcal{A}$. At some point $\mathcal{A}$ sends to $\mathcal{B}$ a list of messages $m_1, m_2, \dots, m_\ell$ for which it would like to have a signature. $\mathcal{B}$ then, sets $pk_1 = m_1, \dots, pk_\ell = m_\ell$ as the list of public keys to submit to the equivocation oracle $\mathcal{E}$. Then when $\mathcal{B}$ receives the public key $\mathtt{MPK}$ for the multi-trapdoor commitment scheme it sets $PK = \mathtt{MPK}$ as the public key for the signature scheme and sends this information to $\mathcal{A}$.

Next $\mathcal{B}$ simulates the signing oracle by simply asking $\mathcal{E}$ to generate commitments and decommitments for the pk_i's selected previously. Now if we assume that $\mathcal{A}$ can break the existential unforgeability of the signature scheme, this means that, with some non-negligible probability $\mathcal{A}$ will be able to generate a signature σ for a message $m \in \mathcal{M}$ not in the list it previously submitted. In particular σ will be a triplet of the form (C, y_0, y_1) such that $\texttt{Ver}(\texttt{MPK}, C, m, y_0, 0) = \texttt{Ver}(\texttt{MPK}, C, m, y_1, 1)$ and $pk = m$ was not previously used in any query to $\mathcal{E}$. This, clearly, leads to a contradiction. □

References

[1] B. Barak, How to go beyond the black-box simulation barrier, in: 42nd Symposium on Foundations of Computer Science, FOCS'01, IEEE Computer Society Press, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 2001, pp. 106–115.

[2] B. Barak, Constant-round coin-tossing with a man in the middle or realizing the shared random string model, in: 43th IEEE Symposium on Foundations of Computer Science, FOCS'02, 2002, pp. 345–355.

[3] N. Barić, B. Pfitzmann, Collision-free accumulators and fail-stop signature schemes without trees, in: Advances in Cryptology — Eurocrypt'97, in: Lecture Notes in Computer Science, vol. 1233, Springer Verlag, 1997, pp. 480–494.

[4] B. Barak, S.J. Ong, S. Vadhan, Derandomization in criptography, in: Crypto'03, 2003.

[5] M. Bellare, M. Jakobsson, M. Yung, Round-optimal zero-knowledge arguments based on any one-way function, in: W. Fumy (Ed.), Advances in Cryptology — Eurocrypt'97, in: Lecture Notes in Computer Science, vol. 1223, Springer-Verlag, 1997, pp. 280–305.

[6] M. Blum, Coin flipping by phone, in: 24th IEEE Computer Conference, CompCon, 1982, pp. 133–137.

[7] M. Blum, How to prove a theorem so no one else can claim it, in: Proceedings of the International Congress of Mathematicians, 1986, pp. 1444–1451.

[8] M. Blum, A. De Santis, S. Micali, G. Persiano, Non-interactive zero-knowledge, SIAM J. Comput. 20 (6) (1991) 1084–1118.

[9] D. Boneh, X. Boyen, Short signatures without random oracles, in: Advances in Cryptology — Eurocrypt'04, in: Lecture Notes in Computer Science, vol. 3027, Springer-Verlag, 2004, pp. 56–73.

[10] R. Boppana, J. Hastad, S. Zachos, Does co-NP have short interactive proofs? Inf. Process. Lett. 25 (2) (1987) 127–132.

[11] J. Brassard, D. Chaum, C. Crepéau, Minimum disclosure proofs of knowledge, J. Comput. Syst. Sci. 37 (2) (1988) 156–189.

[12] E. Bresson, D. Catalano, D. Pointcheval, A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications, in: Advances in Cryptology — Asiacrypt'03, in: Lecture Notes in Computer Science, vol. 2894, Springer-Verlag, 2003, pp. 37–54.

[13] J. Camenisch, Group Signature Schemes and Payment Systems Based on the Discrete Logarithm Problem, in: ETH-Series in Information Security an Cryptography, vol. 2, 1998.

[14] R. Canetti, M. Fischlin, Universally composable commitments, in: Advances in Cryptology — Crypto'01, in: Lecture Notes in Computer Science, vol. 2139, Springer-Verlag, 2001, pp. 19–40.

[15] R. Canetti, J. Kilian, E. Petrank, A. Rosen, Black-box concurrent zero-knowledge requires $\omega(\log n)$ rounds, in: 33st ACM Symposium on Theory of Computing, STOC'01, ACM, 2001, pp. 570–579.

[16] R. Canetti, Y. Lindell, R. Ostrovsky, A. Sahai, Universally composable two-party and multi-party secure computation, in: 34th ACM Symposium on Theory of Computing, STOC'02, ACM, 2002, pp. 494–503.

[17] R. Canetti, Y. Lindell, R. Ostrovsky, A. Sahai, Universally composable two-party and multi-party secure computation, in: 34th ACM Symposium on Theory of Computing, STOC'02, ACM, 2002, pp. 494–503.

[18] D. Catalano, R. Gennaro, N. Howgrave-Graham, P.Q. Nguyen, Paillier cryptosystem revisited, in: 8th ACM Conference on Computer and Communications Security, CCS'01, ACM, 2001, pp. 206–214.

[19] D. Catalano, Y. Dodis, I. Visconti, Mercurial commitments: Minimal assumptions and efficient constructions, in: Proc. of the 3rd Theory of Cryptography Conference, TCC'06, in: Lecture Notes in Computer Science, Springer-Verlag, 2006.

[20] D. Catalano, I. Visconti, Hybrid trapdoor commitments and their applications, in: 32nd International Colloquium on Automata, Languages, and Programming, ICALP 05, in: Lecture Notes in Computer Science, vol. 3580, Springer-Verlag, 2005, pp. 298–310.

[21] D. Catalano, I. Visconti, Non-interactive mercurial commitments from one-way functions, ECRYPT-Provilab Technical Report, October 12, 2005.

[22] M. Chase, A. Healy, A. Lysysanskaya, T. Malkin, L. Reyzin, Mercurial commitments with applications to zero-knowledge sets, in: Proc. of EUROCRYPT, 2005, pp. 422–439.

[23] S.A. Cook, The complexity of theorem proving procedures, in: 3rd Symposium on Foundations of Computer Science 1971, FOCS'71, 1971, pp. 151–158.

[24] R. Cramer, I. Damgård, B. Schoenmakers, Proofs of partial knowledge and simplified design of witness hiding protocols, in: Y. Desmedt (Ed.), Advances in Cryptology — Crypto'94, in: Lecture Notes in Computer Science, vol. 839, Springer-Verlag, 1994, pp. 174–187.

[25] R. Cramer, V. Shoup, Signature schemes based on the strong RSA assumption, in: 6th ACM Conference on Computer and Communications Security, CCS'99, ACM, 1999.

[26] I. Damgård, J. Groth, Non interactive and reusable non-malleable commitments, in: 35th ACM Symposium on Theory of Computing, ACM, 2003, pp. 426–437.

[27] I. Damgård, M. Jurik, A generalization, a simplification and some applications of paillier's probabilistic public-key system, in: Public key Cryptography, in: Lecture Notes in Computer Science, vol. 1992, Springer Verlag, 2001, pp. 119–136.

[28] I. Damgård, J.B. Nielsen, Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor, in: Advances in Cryptology — Crypto'02, in: Lecture Notes in Computer Science, vol. 2442, Springer-Verlag, 2002, pp. 581–596.

[29] I. Damgård, Efficient concurrent zero-knowledge in the auxiliary string model, in: Advances in Cryptology — Eurocrypt'00, in: Lecture Notes in Computer Science, vol. 1807, Springer-Verlag, 2000, pp. 418–430.

[30] A. De Santis, G. Di Crescenzo, R. Ostrovsky, G. Persiano, A. Sahai, Robust non-interactive zero knowledge, in: Advances in Cryptology — Crypto'01, in: Lecture Notes in Computer Science, vol. 2139, Springer-Verlag, 2001, pp. 566–598.

[31] G. Di Crescenzo, Y. Ishai, R. Ostrovsky, Non-interactive and non-malleable commitment, in: 30th ACM Symposium on Theory of Computing, STOC'98, ACM, 1998, pp. 141–150.

[32] G. Di Crescenzo, J. Katz, R. Ostrovsky, A. Smith, Efficient and non-interactive non-malleable commitment, in: Advances in Cryptology — Eurocrypt'01, in: Lecture Notes in Computer Science, vol. 2045, Springer-Verlag, 2001, pp. 40–59.

[33] G. Di Crescenzo, G. Persiano, I. Visconti, Constant-round resettable zero knowledge with concurrent soundness in the bare public-key model, in: Advances in Cryptology — Crypto'04, in: Lecture Notes in Computer Science, vol. 3152, Springer-Verlag, 2004, pp. 237–253.

[34] D. Dolev, C. Dwork, M. Naor, Non-malleable cryptography, SIAM J. Comput. 30 (2) (2000) 391–437.

[35] C. Dwork, M. Naor, A. Sahai, Concurrent zero-knowledge, in: 30th ACM Symposium on Theory of Computing, STOC'98, ACM, 1998, pp. 409–418.

[36] C. Dwork, A. Sahai, Concurrent zero-knowledge: Reducing the need for timing constraints, in: H. Krawczyk (Ed.), Advances in Cryptology — Crypto'98, in: Lecture Notes in Computer Science, vol. 1462, Springer-Verlag, 1998, pp. 442–457.

[37] U. Feige, D. Lapidot, A. Shamir, Multiple non-interactive zero knowledge proofs under general assumptions, SIAM J. Comput. 29 (1) (1999) 1–28.

[38] U. Feige, A. Shamir, Zero-knowledge proofs of knowledge in two rounds, in: Advances in Cryptology — Crypto'89, in: Lecture Notes in Computer Science, vol. 435, Springer-Verlag, 1990, pp. 526–544.

[39] M. Fischlin, On the impossibility of constructing non-interactive statistically-secret protocols from any trapdoor one-way function, in: CT-RSA'02, in: Lecture Notes in Computer Science, vol. 2271, Springer-Verlag, 2002, pp. 79–95.

[40] M. Fischlin, R. Fischlin, Efficient and non-malleable commitment schemes, in: Advances in Cryptology — Crypto'00, in: Lecture Notes in Computer Science, vol. 1880, Springer-Verlag, 2000, pp. 413–431.

[41] L. Fortnow, The complexity of perfect zero-knowledge, in: 19th ACM Symposium on Theory of Computing, STOC'87, 1987, pp. 204–209.

[42] J. Garay, P. MacKenzie, K. Yang, Strengthening zero-knowledge protocols using signatures, in: Advances in Cryptology — Eurocrypt'03, in: Lecture Notes in Computer Science, vol. 2045, Springer-Verlag, 2003, pp. 177–194.

[43] R. Gennaro, Multi-trapdoor commitments and their applications to proofs of knowledge secure under concurrent man-in-the-middle attacks (full version), In e-print archive, available at: http://eprint.iacr.org/2003/214/, 2003.

[44] R. Gennaro, Multi-trapdoor commitments and their applications to proofs of knowledge secure under concurrent man-in-the-middle attacks, in: Advances in Cryptology — Crypto'04, in: Lecture Notes in Computer Science, vol. 3152, Springer-Verlag, 2004, pp. 220–236.

[45] O. Goldreich, Concurrent zero-knowledge with timing, revisited, in: 34th ACM Symposium on Theory of Computing, STOC'02, ACM, 2002, pp. 332–340.

[46] O. Goldreich, A. Kahan, How to construct constant-round zero-knowledge proof systems for NP, J. Cryptol. 9 (3) (1996) 167–190.

[47] O. Goldreich, L. Levin, A hard-core predicate for all one-way functions, in: 21st ACM Symposium on Theory of Computing, STOC'89, 1989, pp. 25–32.

[48] S. Goldwasser, S. Micali, C. Rackoff, The knowledge complexity of interactive proof-systems, SIAM J. Comput. 18 (6) (1989) 186–208.

[49] S. Goldwasser, S. Micali, R. Rivest, A digital signature scheme secure against adaptive chosen message attacks, SIAM J. Comput. 17 (2) (1988) 281–308.

[50] I. Haitner, O. Horvitz, J. Katz, C.Y. Koo, R. Morselli, R. Shaltiel, Reducing complexity assumptions for statistically-hiding commitment, in: Advances in Cryptology — Eurocrypt'05, in: Lecture Notes in Computer Science, vol. 3494, Springer-Verlag, 2005, pp. 58–77.

[51] S. Halevi, S. Micali, Practical and provably-secure commitment schemes from collision-free hashing, in: Advances in Cryptology — Crypto'96, in: Lecture Notes in Computer Science, vol. 1109, Springer-Verlag, 1996, pp. 201–215.

[52] J. Kilian, E. Petrank, Concurrent and resettable zero-knowledge in poly-logarithmic rounds, in: 33rd ACM Symposium on Theory of Computing, STOC'01, ACM, 2001, pp. 560–569.

[53] S. Micali, M. Rabin, J. Kilian, Zero-knowledge sets, in: Proc. 44th IEEE Symposium on Foundations of Computer Science, FOCS, 2003.

[54] P. MacKenzie, K. Yang, On simulation-sound trapdoor commitments. In e-print archive, available at: http://eprint.iacr.org/2003/252/, 2003.

[55] P. MacKenzie, K. Yang, On simulation-sound trapdoor commitments, in: Advances in Cryptology — Eurocrypt'04, in: Lecture Notes in Computer Science, vol. 3027, Springer-Verlag, 2004, pp. 382–400.

[56] D. Micciancio, E. Petrank, Simulatable commitments and efficient concurrent zero-knowledge, in: Advances in Cryptology — Eurocrypt'03, in: Lecture Notes in Computer Science, vol. 2045, Springer-Verlag, 2003, pp. 140–159.

[57] M. Naor, Bit commitment using pseudorandomness, J. Cryptol. 4 (2) (1991) 151–158.

[58] NIST. Digital Signature Standard (DSS). FIPS PUB 186, December 1998.

[59] T. Okamoto, S. Uchiyama, A new public-key cryptosystem as secure as factoring, in: Advances in Cryptology — Eurocrypt'98, in: Lecture Notes in Computer Science, vol. 1233, Springer-Verlag, 1998, pp. 308–318.

[60] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, in: J. Stern (Ed.), EUROCRYPT'99, in: LNCS, vol. 1592, Springer-Verlag, 1999, pp. 223–238.

[61] R. Pass, Simulation in quasi-polynomial time and its applications to protocol composition, in: Advances in Cryptology — Eurocrypt'03, in: Lecture Notes in Computer Science, vol. 2045, Springer-Verlag, 2003, pp. 160–176.

[62] R. Pass, A. Rosen, Bounded-concurrent secure two-party computation in a constant number of rounds, in: 44th IEEE Symposium on Foundations of Computer Science, FOCS'03, 2003.

[63] G. Persiano, I. Visconti, Single-prover concurrent zero knowledge in almost constant rounds, in: 32nd International Colloquium on Automata, Languages, and Programming, ICALP 05, in: Lecture Notes in Computer Science, vol. 3580, Springer-Verlag, 2005, pp. 228–240.

[64] M. Prabhakaran, A. Rosen, A. Sahai, Concurrent zero-knowledge with logarithmic round complexity, in: 43th IEEE Symposium on Foundations of Computer Science, FOCS'02, 2002, pp. 366–375.

[65] R. Richardson, J. Kilian, On the concurrent composition of zero-knowledge proofs, in: Advances in Cryptology — Eurocrypt'99, in: Lecture Notes in Computer Science, vol. 1592, Springer-Verlag, 1999, pp. 415–431.

[66] R. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public key cryptosystems, Commun. ACM (1978) 120–126.

[67] J. Rompel, One-way functions are necessary and sufficient for digital signatures, in: 22nd ACM Symposium on Theory of Computing, STOC'90, 1990, pp. 12–19.

[68] A. Sahai, Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security, in: 40th Symposium on Foundations of Computer Science, FOCS'99, IEEE Computer Society Press, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1999, pp. 543–553.

[69] C.P. Schnorr, Efficient signature generation for smart cards, J. Cryptol. 4 (3) (1991) 239–252.

[70] I. Visconti, Efficient zero knowledge on the internet, in: 33rd International Colloquium on Automata, Languages, and Programming, ICALP'06, in: Lecture Notes in Computer Science, Springer-Verlag, 2006.