Computer Science and Artificial Intelligence Laboratory
Technical Report

MIT-CSAIL-TR-2008-022 April 16, 2008

New-Age Cryptography

Rafael Pass and Vinod Vaikuntanathan

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

CSAIL

New-Age Cryptography

Rafael Pass
Cornell University[*]

Vinod Vaikuntanathan
MIT CSAIL[†]

Abstract

We introduce new and general complexity theoretic hardness assumptions. These assumptions abstract out concrete properties of a random oracle and are significantly stronger than traditional cryptographic hardness assumptions; however, assuming their validity we can resolve a number of long-standing open problems in cryptography.

Keywords. Cryptographic Assumptions, Non-malleable Commitment, Non-malleable Zero-knowledge

[*] E-mail: rafael@cs.cornell.edu

[†] E-mail: vinodv@theory.csail.mit.edu. Supported in part by NSF Grant CNS-0430450.

1 Introduction

The state-of-the-art in complexity theory forces cryptographers to base their schemes on unproven hardness assumptions. Such assumptions can be general (e.g., the existence of one-way functions) or specific (e.g., the hardness of RSA or the Discrete logarithm problem). Specific hardness assumptions are usually stronger than their general counterparts; however, as such assumptions consider primitives with more structure, they lend themselves to constructions of more efficient protocols, and sometimes even to the constructions of objects that are not known to exist when this extra structure is not present. Indeed, in recent years, several new and more exotic specific hardness assumptions have been introduced (e.g., [11, 4, 10]) leading to, among other things, signatures schemes with improved efficiency, but also the first provably secure construction of identity-based encryption.

In this paper, we introduce a new class of strong but *general* hardness assumptions, and show how these assumptions can be used to resolve certain long-standing open problems in cryptography. Our assumptions are all abstractions of concrete properties of a random oracle. As such, our results show that for the problems we consider, random oracles are not necessary; rather, provably secure constructions can be based on concrete hardness assumptions.

1.1 New-Age Assumptions

We consider *adaptive* strengthenings of standard general hardness assumptions, such as the existence of one-way functions and pseudorandom generators. More specifically, we introduce the notion of collections of adaptive 1-1 one-way functions and collections of adaptive pseudorandom generators. Intuitively,

- A *collection of adaptively 1-1 one-way functions* is a family of 1-1 functions $\mathcal{F}_n = \{f_{tag} : \{0,1\}^n \mapsto \{0,1\}^n\}$ such that for every tag, it is hard to invert $f_{tag}(r)$ for a random r, even for an adversary that is granted access to an "inversion oracle" for $f_{tag'}$ for every $tag \neq tag'$. In other words, the function f_{tag} is one-way, even with access to an oracle that invert all the other functions in the family.

- A collection of *adaptive pseudo-random generators* is a family of functions $\mathcal{G}_n = G_{tag} : \{0,1\}^n \mapsto \{0,1\}^m$ such that for every tag, G_{tag} is a pseudorandom even if given access to an oracle that decides whether given y is in the range of G.

Both the above assumptions are strong, but arguably not "unrealistically" strong. Indeed, both these assumptions are satisfied by a (sufficiently) length-extending random oracle.[1] As such, they provide concrete mathematical assumptions that can be used to instantiate random oracles in certain applications.

We also present some concrete candidate instantiations of these assumptions. For the case of adaptive 1-1 one-way functions, we provide construction based on the the "adaptive security" of Factoring, or the Discrete Log problem.

For the case of adaptive PRGs, we provide a candidate construction based on a generalization of the advanced encryption standard (AES).

Related Assumptions in the Literature. Assumptions of a related flavor have appeared in a number of works. The class of "one-more" assumptions introduced by Bellare, Namprempre, Pointcheval and Semanko [4] are similar in flavor. Informally, the setting of the one-more RSA-inversion problem is the following: The adversary is given values $z_1, z_2, \dots, z_k \in \mathbb{Z}_N^*$ (for a composite $N = pq$, a product of two

[1] Note that a random function over, say, $\{0,1\}^n \to \{0,1\}^{4n}$ is 1-1 except with exponentially small probability.

primes) and is given access to an oracle that computes RSA inverses. The adversary wins if the number of values that it computes an RSA inverse of, exceeds the number of calls it makes to the oracle. They prove the security of Chaum's blind-signature scheme under this assumption. This flavor of assumptions has been used in numerous other subsequent works [5, 6].

Prabhakaran and Sahai [28] use an assumption of the form that there are collision-resistant hash functions that are secure even if the adversary has access to a "collision-sampler". In a related work, Malkin, Moriarty and Yakovenko [21] assume that the discrete logarithm problem in $\mathbb{Z}_p^*$ (where p is a k-bit prime) is hard even for an adversary that has access to an oracle that computes discrete logarithms in $\mathbb{Z}_q^*$ for *any* k-bit prime $q \neq p$. Both these works use the assumption to achieve secure computation in a relaxation of the universal composability framework.

1.2 New-Age Results

Non-Interactive Concurrently Non-Malleable Commitment Schemes. Non-malleable commitment schemes were first defined and constructed in the seminal paper of Dolev, Dwork and Naor [16]. Informally, a commitment scheme is non-malleable if no adversary can, upon seeing a commitment to a value v, produce a commitment to a related value (say $v - 1$). Indeed, non-malleability is crucial to applications which rely on the *independence* of the committed values. A much stronger property – called concurrent non-malleability – requires that no adversary, after receiving commitments of $v_1, \ldots, v_m$, can produce commitments to related values $\tilde{v}_1, \ldots, \tilde{v}_m$.

The first non-malleable commitment scheme of [16] was interactive, and required $O(\log n)$ rounds of interaction, where n is a security parameter. Barak [1] and subsequently, Pass and Rosen [26] presented constant-round non-malleable commitment schemes. The only known construction of a concurrent non-malleable commitment scheme is due to Pass and Rosen [25], and requires 12 rounds of interaction between the committer and the receiver.

We note that of the above commitment schemes, [16] is the only one with a black-box proof of security, whereas the schemes of [1, 26, 25] rely on the novel non-black-box proof technique introduced by [1]. In particular, there is no known concurrently non-malleable commitment schemes with a black-box proof of security.

Our first result is a construction of a *non-interactive, concurrently non-malleable* string commitment scheme, from a family of adaptive one-way permutations. Additionally, our construction is the first concurrently non-malleable commitment scheme with a black-box proof of security.

Theorem 1 (Informal) *Assume the existence of collections of adaptive 1-1 permutations. Then, there exists a non-interactive concurrently non-malleable string commitment scheme with a black-box proof of security.*

If instead assuming the existence of adaptive PRGs, we show the existence of 2-round concurrent non-malleable commitment with a black-box proof of security.

Theorem 2 (Informal) *Assume the existence of collections of adaptive PRGS. Then, there exists a 2-round concurrently non-malleable string commitment scheme with a black-box proof of security.*

Round-optimal Black-box Non-malleable Zero-knowledge. Dolev, Dwork and Naor [16] defined non-malleable zero-knowledge (ZK) and presented an $O(\log n)$-round ZK proof system. Barak [1] and subsequently, Pass and Rosen [26] presented constant-round non-malleable zero-knowledge argument system. Of the above protocols, [16] is the only one with a black-box proof of security, whereas the schemes of [1, 26] rely on the non-black-box proof technique of [1].

We construct a 4-*round non-malleable zero-knowledge argument* system with a black-box proof of security (that is, a black-box simulator). Four rounds is known to be optimal for black-box zero-knowledge [18] (even if the protocol is not required to be non-malleable) and for non-malleable protocols (even if they are not required to be zero-knowledge) [20].

Theorem 3 (Informal) *Assume the existence of collections of adaptive 1-1 one-way function. Then, there exists a 4-round non-malleable zero-knowledge argument system with a black-box proof of security. Assume, instead, the existence of collections of adaptive one-way permutations. Then, there exists a 5-round non-malleable zero-knowledge argument system with a black-box proof of security.*

It is interesting to note that the (seemingly) related notion of concurrent zero-knowledge cannot be achieved in $o(\log n)$ rounds with a black-box proof of security. Thus, our result shows that (under our new-age assumptions), the notion of non-malleability and concurrency in the context of zero-knowledge are quantitatively different.

Efficient Chosen-Ciphertext Secure Encryption. Chosen ciphertext (CCA) security was introduced in the works of [23, 29] and has since been recognized as a *sine-qua-non* for secure encryption. Dolev, Dwork and Naor [16] gave the first construction of CCA-secure encryption schemes based on general assumptions. Their construction, as well as the construction of Sahai [30], uses the machinery of non-interactive zero-knowledge proofs, which renders them less efficient than one would like. In contrast, the construction of Cramer and Shoup [14, 15] are efficent, but are based on specific number-theoretic assumptions.

Bellare and Rogaway [7] proposed an encryption scheme that is CCA-secure in the random oracle model (see below for more details about the random oracle model). We show complexity-theoretic assumptions that are sufficient to replace the random oracle in this construction. We mention that, previously, Canetti [12] showed how to replace random oracles in a related construction to get a semantically secure encryption scheme (but without CCA security). Our construction of CCA-secure encryption is in Appendix D.

Interactive Arguments for which Parallel-repetition does not reduce the soundness error. A basic question regarding interactive proof is whether parallel repetition of such protocols reduces the soundness error. Bellare, Impagliazzo and Naor [3] show that there are interactive *arguments* (i.e., computationally-sound) proofs in the Common Reference String (CRS) model, for which parallel-repetition does not reduce the soundness error. Their construction relies on non-malleable encryption, and makes use of the CRS to select the public-key for this encryption scheme. However, if instead relying on a non-interactive concurrent non-malleable commitment schemes in their construction, we can dispense of the CRS altogether. Thus, by Theorem 1, assuming the existence of collections of adaptive 1-1 one-way functions, we show that there exists an interactive argument for which parallel repetition does not reduce the soundness error. We also mention that the same technique can be applied also to the strengthened construction of [27].

Our Techniques. All our constructions are simple and efficient. In particular, for the case of non-malleable commitment schemes, we show that appropriate instantiations of the Blum-Micali [9] or Naor [22] commitment schemes in fact are non-malleable. The proof of these schemes are also "relatively straight-forward" and follow nicely from the adaptive property of the underlying primitives.

Next, we show that by appropriately using our non-malleable commitment protocols in the Feige-Shamir [17] zero-knowledge argument for $\mathcal{NP}$, we can also get a round-optimal black-box non-malleable $\mathcal{ZK}$ proof for $\mathcal{NP}$. Although the construction here is straight-forward, its proof of correctness is less so. In particular, to show that our protocol is non-malleable, we rely on a techniques that are quite different from traditional proofs of non-malleability: in particular, the power of the "adaptive" oracle will only be used inside hybrid experiments; the simulation, on the other hand, will proceed by traditional rewinding. Interestingly, to get a round-optimal solution, our proof inherently relies on the actual Feige-Shamir protocol and high-lights some novel features of this protocol.

Interpreting Our Results. We offer two interpretations of our results:

- The *optimistic* interpretation: Although our assumptions are strong, they nonetheless do not (a priori) seem infeasible. Thus, if we believe that e.g., AES behaves as an adaptively secure PRG, we show *practical* solutions to important open questions.

- The *conservative* interpretation: As mentioned, our constructions are black-box; namely, both the construction of the cryptographic objects and the associated security proof utilize the underlying primitive—adaptive one-way permutations or adaptive PRGs—as a black-box, and in particular, do not refer to a specific implementation of these primitives. Thus, a conservative way to view our results is that to show even black-box lower-bounds and impossibility results for non-interactive concurrent non-malleable commitments and non-malleable zero-knowledge proofs, one first needs to to refute our assumptions. Analogously, it means that breaking our CCA-secure encryptions scheme, or proving a general parallel-repetition theorem for interactive arguments, first requires refuting our assumptions.

1.3 New-Age Perspective

A cryptographer could choose to make "mild" assumptions such as $\mathcal{P} \neq \mathcal{NP}$, "relatively mild" ones such as the existence of one-way functions, secure encryption schemes or trapdoor permutations, or "preposterous" ones such as "this scheme is secure". Whereas preposterous assumptions clearly are undesirable, mild assumptions are—given the state-of-the-art in complexity theory—too weak for cryptographic constructions of non-trivial tasks. Relatively mild assumptions, on the other hand, are sufficient for showing the feasibility of essentially all known cryptographic primitives.

Yet, to obtain practical constructions, such assumptions are—given the current-state-of-art—not sufficient. In fact, it is a priori not even clear that although feasibility of a cryptographic task can be based on a relatively mild assumptions, that a "practical" construction of the primitive is possible (at all!). One approach to overcome this gap is the random oracle paradigm, introduced in the current form by Bellare and Rogaway [7]: the proposed paradigm is to prove the security of a cryptographic scheme in the random-oracle model—where all parties have access to a truly random function—and next instantiate the random oracle with a concrete function "with appropriate properties". Nevertheless, as pointed out in [13] (see also [19, 2]) there are (pathological) schemes that can be proven secure in the random oracle model, but are rendered insecure when the random oracle is replaced by any concrete function (or family of functions).

In this work we, instead, investigate a different avenue for overcoming this gap between theory and practice, by introducing strong, but general, hardness assumption. When doing so, we, of course, need to be careful to make sure that our assumptions (although potentially "funky") are not preposterous. One criterion in determining the acceptability of a cryptographic assumption A is to consider (1) what the assumption is used for (for instance, to construct a primitive P, say) and (2) how much more "complex" the primitive P is, compared to A. For example, a construction of a pseudorandom generator assuming a one-way function is non-trivial, whereas the reverse direction is not nearly as interesting. Unfortunately, the notion of “complexity” of an assumption is hard to define. We here offer a simple interpretation: view complexity as “succinctness”. General assumption are usually more succinct than specific assumptions, one-way functions are “easier” to define than, say, pseudorandom functions. Given this point of view, it seems that our assumptions are not significantly more complex than traditional hardness assumption; yet they allow us to construct considerably more complex objects (e.g., non-malleable zero-knowledge proofs).

On Falsifiability/Refutability of Our Assumptions Note that the notions of non-malleable commitment and non-malleable zero-knowledge both are defined using simulation-based definitions. As such, simply assuming that a practical scheme is, say, non-malleable zero-knowledge, seems like a very strong assumption, which is hard to falsify[2]—in fact, to falsify it one needs to show (using a mathematical proof) that no Turning machine is a good simulator. In contrast, to falsify our assumptions it is sufficient to exhibit an attacker (just as with the traditional cryptographic hardness assumptions).

To make such “qualitative” differences more precise, Naor [24] introduced a framework for classifying assumptions, based on how “practically” an assumption can refuted. Whereas non-malleability, a priori, seems impossible to falsify (as there a-priori is not a simple way to showing that no simulator exists). In contrast, traditional assumptions such as “factoring is hard” can be easily refuted simply by publishing challenges that a “falsifier” is required to solve. Our assumptions cannot be as easily refuted, as even if a falsifier exhibits an attack against a candidate adaptive OWF, it is unclear how to check that this attack works. However, the same can be said also for relatively mild (and commonly used) assumptions, such as “factoring is hard for subexponential-time”.[3]

Additionally, we would like to argue that our assumptions enjoy a similar “win/win” situation as traditional cryptographic hardness assumptions. The adaptive security of the factoring or discrete logarithm problems seem like natural computational number theoretic questions. A refutation of our assumptions (and its implication to factoring and discrete logarithm problem) would thus be interesting in its own right. Taken to its extreme, this approach suggest that we might even consider assumptions that most probably are *false*, such as e.g., assuming that AES is an (adaptive one-way) *permutation*, as long as we believe that it might be hard to prove that the assumption is false.

2 New Assumptions and Definitions

The following sections introduce our definitions of adaptively secure objects—one-way functions, pseudorandom generators and commitment schemes—and posit candidate constructions for adaptively secure one-way functions and pseudorandom generators. Standard cryptographic definitions (non-malleable commitments and zero-knowledge) are delegated to Appendix A.

2.1 Adaptive One-Way Functions

In this paper, we define a *family* of adaptively secure injective one-way functions, where each function in the family is specified by an index tag. The adaptive security requirement says the following: consider an adversary that picks an index tag^* and is given $y^* = f_{\mathsf{tag}^*}(x^*)$ for a random x^* in the domain of f_{tag^*},

[2] Recall that falsifiability is Popper’s classical criterion for distinguishing scientific and “pseudo-scientific” statements.

[3] Note that the assumption that factoring is hard for subexponential-time can be falsified by considering a publishing a very “short” challenge (or length $\mathrm{polylog} n$). However, in the same vein, our assumption can be falsified by considering challenges of length $\log n$; then it is easy to check if someone can exhibit an efficient attack on the adaptive security of an assumed one-way function, since the inverting oracle can also be efficiently implemented.

and the adversary is supposed to compute x^*. The adversary, in addition, has access to a "magic oracle" that on input (tag, y) where $\mathsf{tag} \neq \mathsf{tag}^*$, and get back $f_{\mathsf{tag}}^{-1}(y)$. In other words, the magic oracle helps invert all functions f_{tag} different from the "target function" f_{tag^*}. The security requirement is that the adversary have at most a negligible chance of computing x^*, even with this added ability. Note that the magic oracle is just a fictitious entity, which possibly does not have an efficient implementation (as opposed to the decryption oracle in the definition of CCA-security for encryption schemes which can be implemented efficiently given the secret-key). Inability to invert y^* even with access to such an oracle is indeed a strong security requirement on the function f! More formally,

Definition 1 (Family of Adaptive One-to-one One-way Functions) *A family of* injective *one-way functions $\mathcal{F} = \{f_{\mathsf{tag}} : D_{\mathsf{tag}} \mapsto \{0,1\}^*\}_{\mathsf{tag}\in\{0,1\}^n}$ is called adaptively secure if,*

- (EASY TO SAMPLE AND COMPUTE.) *There is an efficient randomized* domain-sampler D*, which on input* $\mathsf{tag} \in \bar{I}$*, outputs a random element in* D_{tag}*. There is a deterministic polynomial algorithm* M *such that for all* $\mathsf{tag} \leftarrow I$ *and for all* $x \in D_{\mathsf{tag}}$*,* $M(\mathsf{tag}, x) = f_{\mathsf{tag}}(x)$*.*
- (ADAPTIVE ONE-WAYNESS.) *Let* $\mathcal{O}(\mathsf{tag}, \cdot, \cdot)$ *denote an oracle that, on input* tag' *and* y *outputs* $f_{\mathsf{tag}'}^{-1}(y)$ *if* $\mathsf{tag}' \neq \mathsf{tag}$ *and* $\perp$ *otherwise.*[4]

  The family $\mathcal{F}$ is adaptively secure if, for any probabilistic polynomial-time adversary A, there exists a negligible function μ such that for all sufficiently large k, and for all tags $\mathsf{tag} \in \bar{I}_k$*,*

$$\Pr[x \leftarrow D(\mathsf{tag}, 1^k) : A^{\mathcal{O}(\mathsf{tag},\cdot,\cdot)}(\mathsf{tag}, f_{\mathsf{tag}}(x)) = x] \leq \mu(n)$$

  where the probability is over the random choice tag *output by I, the random choice of x and the coin-tosses of A.*

A potentially incomparable assumption is that of an adaptively secure injective one-way function (as opposed to a family of functions). However, it is easy to see that an adaptively secure one-way function with subexponential security and a dense domain implies a family of adaptively secure one-way functions, as defined above. In fact, our construction of a family of adaptively secure one-way functions based on factoring goes through this construction.

Hardness Amplification. A strong adaptively secure one-way function is one where no adversary can invert the function with probability better than some negligible function in k (even with access to the inversion oracle). A weak one, on the other hand, only requires that the adversary not be able to invert the function with a probability better than $1 - 1/\mathrm{poly}(k)$ (even with access to the inversion oracle).

We remark that we can construct a strong adaptively secure one-way function from a weak adaptively secure one-way function. The construction is the same as Yao's hardness amplification lemma. We defer the details to the full version.

2.1.1 Candidates

We present candidates for adaptively secure one-way functions, based on assumptions related to discrete-log and factoring.

[4] If tag' is not a valid tag, namely, if tag' is not in the range of $I(1^k)$, then again $\mathcal{O}$ outputs $\perp$.

Factoring. First, we show how to build an adaptively secure one-way function (not a family of functions) from the factoring assumption. Then, we show how to turn it into a family of functions, assuming, in addition, that factoring is subexponentially-hard.

The domain of the function f is $\{(p, q) \mid p, q \in \mathcal{P}_n, p < q\}$, where $\mathcal{P}_n$ is the set of all n-bit primes. Given this notation, $f(p, q)$ is defined to be pq. Assuming that it is hard to factor a number N that is a product of primes, even with access to an oracle that factors all other products of two primes, this function is adaptively secure.

We now show how to turn this into a family of adaptively secure one-way functions. The index is simply an $n' = n^{1/\epsilon}$-bit string (for some $\epsilon > 0$) $i = (i_1, i_2)$. The domain is the set of all strings (j_1, j_2) such that $p = i_1 \circ j_1$ and $q = i_2 \circ j_2$ are both n-bit primes. The function then outputs pq. Since we reveal the first $n' = n^{1/\epsilon}$ bits of the factors of $N = pq$, we need to assume that factoring is subexponentially hard (even with access to an oracle that factors other products of two primes). The function is clearly injective since factoring forms an injective function.

In the full version, we additionally provide candidates for adaptive one-way functions based on the RSA and Rabin functions.

Discrete Logarithms. The family of adaptive OWFs $\mathcal{F}_{DL}$ is defined as follows: The index set $\bar{I}_n = \{0, 1\}^n$. The domain of the function is a tuple (p, g, x) such that p is a $2n$-bit prime p whose first n bits equal the index i, g is a generator for $\mathbb{Z}_p^*$ and x is a $2n-1$-bit number. The domain is easy to sample–the sampler picks a "long-enough" random string r and a $2n-1$-bit number x. The function f_i uses r to sample a $2n$-bit prime p whose first n bits equal i (this can be done by repeated sampling, and runs in polynomial time assuming a uniformness conjecture on the density of primes in large intervals) and a generator $g \in \mathbb{Z}_p^*$. The output of the function on input (p, g, x) is $(p, g, g^x \bmod p)$. f_i is injective since the output determines p and g; given p and g, $g^x \bmod p$ next determines x uniquely since $x < 2^{2n-1}$ and p, being a $2n$-bit prime, is larger than 2^{2n-1}.

We also mention that the adaptive security of this family can be based on the subexponential adaptive security of the one-way function (as opposed to family) obtained by simply sampling random p, g, x (or even random p being a safe prime) and outputting p, g, g^x. Note that this assumption is different from the assumption of [21] in that we require security to hold only w.r.t to a random prime p whereas [21] requires it to holds also w.r.t to adversarially chosen p; in contrast we require security w.r.t sub-exponential adversaries.

2.2 Adaptive Pseudorandom Generator

A family of adaptively secure pseudorandom generators $\mathcal{G} = \{G_{\mathsf{tag}}\}_{\mathsf{tag} \in \{0,1\}^*}$ is defined in a similar way to an adaptive one-way function. We require that the output of the generator G, on a random input x and an adversarially chosen tag be indistinguishable from uniform, even for an adversary that can query a magic oracle with a value (tag', y) (where $\mathsf{tag}' \neq \mathsf{tag}$) and get back 0 or 1 depending on whether y is in the range of $G_{\mathsf{tag}'}$ or not.

Definition 2 (Adaptive PRG) *A family of functions $\mathcal{G} = \{G_{\mathsf{tag}} : \{0, 1\}^n \mapsto \{0, 1\}^{s(n)}\}_{\mathsf{tag} \in \{0,1\}^n}$ is an adaptively secure pseudorandom generator (PRG) if $|G_{\mathsf{tag}}(x)| = s(|x|)$ for some function s such that $s(n) \geq n$ for all n and,*

- *(*EFFICIENT COMPUTABILITY.*) There is a deterministic polynomial-time algorithm M_G such that $M_G(x, \mathsf{tag}) = G_{\mathsf{tag}}(x)$.*

- *(*Adaptive Pseudorandomness.*) Let* $\mathcal{O}(\mathsf{tag}, \cdot, \cdot)$ *denote an oracle that, on input* (tag', y) *such that* $\mathsf{tag}' \neq \mathsf{tag}$*, outputs* 1 *if* y *is in the range of* $G_{\mathsf{tag}'}$ *and* 0 *otherwise.*

  The PRG G *is adaptively secure if, for any probabilistic polynomial-time adversary* A*, there exists a negligible function* μ *such that for all sufficiently large* n *and for all tags* $\mathsf{tag} \in \{0,1\}^n$*,*

  $$\left|\Pr[y \leftarrow G_{\mathsf{tag}}(U_n) : A^{\mathcal{O}(\mathsf{tag},\cdot,\cdot)}(y) = 1] - \Pr[y \leftarrow U_m : A^{\mathcal{O}(\mathsf{tag},\cdot,\cdot)}(y) = 1]\right| \leq \mu(n)$$

  where the probability is over the random choice of y *and the coin-tosses of* A*.*

2.2.1 Candidates

For the case of adaptive PRGs, we provide a candidate construction based on the advanced encryption standard (AES). AES is a permutation on 128 bits; that is, for a 128-bit seed s, AES_s is a permutation defined on $\{0,1\}^{128}$. However, due to the algebraic nature of the construction of AES, it can easily be generalized to longer input length. Let AES_n denote this generalized version of AES to n-bit inputs. Our candidate adaptive pseudorandom generator AESG_{tag} is simply $\mathsf{AESG}_{tag}(s) = \mathsf{AES}_s(tag \circ 0) \circ \mathsf{AES}_s(tag \circ 1)$.

2.3 Adaptively Secure Commitment Schemes

In this subsection, we define adaptively secure commitment schemes. Let $\{\mathrm{COM}_{\mathsf{tag}} = \langle S_{\mathsf{tag}}, R_{\mathsf{tag}}\rangle\}_{\mathsf{tag}\in\{0,1\}^*}$ denote a family of commitment protocols, indexed by a string tag. We require that the commitment scheme be secure, even against an adversary that can query a magic oracle on the transcript of a commitment interaction and get back a message that was committed to in the transcript. More precisely, the adversary picks an index tag and two equal-length strings x_0 and x_1 and gets a value $y_b = \mathrm{COM}_{\mathsf{tag}}(x_b; r)$, where b is a random bit and r is random. The adversary can, in addition, query a magic oracle on (y', tag') where $\mathsf{tag}' \neq \mathsf{tag}$ and get back the some x' such that $y' \in \mathrm{COM}_{\mathsf{tag}'}(x'; r')$ (if y' is a legal commitment) and $\perp$ otherwise. [5] The security requirement is that the adversary cannot distinguish whether y_b was a commitment to x_0 or x_1, even with this extra power.

Definition 3 (Adaptively-Secure Commitment) *A family of functions* $\{\mathrm{COM}_{\mathsf{tag}}\}_{\mathsf{tag}\in\{0,1\}^*}$ *is called an adaptively secure commitment scheme if* S_{tag} *and* R_{tag} *are polynomial-time and*

- Statistical Binding: *For any* tag*, over the coin-tosses of the receiver* R*, the probability that a transcript* $\langle S^*, R_{\mathsf{tag}}\rangle$ *has two valid openings is negligible.*

- Adaptive Security: *Let* $\mathcal{O}(\mathsf{tag}, \cdot, \cdot)$ *denote the oracle that, on input* $\mathsf{tag}' \neq \mathsf{tag}$ *and* c*, returns an* $x \in \{0,1\}^\ell$ *such that for some random strings* r_S *and* r_R*,* c *is the transcript of the interaction between* S *with input* x *and random coins* r_S *and* R *with random coins* r_R*.*

  For any probabilistic polynomial-time oracle TM A*, there exists a negligible function* $\mu(\cdot)$ *such that for all sufficiently large* n*, for all* $\mathsf{tag} \in \{0,1\}^*$ *and for all* $x, y \in \{0,1\}^\ell$*,*

  $$\left|\Pr[c \leftarrow \langle S_{\mathsf{tag}}(x), R_{\mathsf{tag}}\rangle; A^{\mathcal{O}(\mathsf{tag},\cdot,\cdot)}(c, \mathsf{tag}) = 1] - \Pr[c \leftarrow \langle S_{\mathsf{tag}}(y), R_{\mathsf{tag}}\rangle; A^{\mathcal{O}(\mathsf{tag},\cdot)}(c, \mathsf{tag}) = 1]\right| \leq \mu(n)$$

[5] In case the transcript corresponds to the commitment of multiple messages, the oracle returns a canonical one of them. In fact, one of our commitment schemes is perfectly binding and thus, does not encounter this problem.

3 Non-Malleable Commitment Schemes

In this section, we construct non-malleable string-commitment schemes. We first construct adaptively-secure bit-commitment schemes based on an adaptively secure injective OWF and an adaptively secure PRG – the first of these constructions is non-interactive and the second is a 2-round commitment scheme. We then show a simple "concatenation lemma", that constructs an adaptively secure string commitment scheme from an adaptively-secure bit-commitment scheme. Finally, we show that an adaptively secure commitment scheme is also non-malleable. For full proofs, see Appendix B.

Lemma 4 *Assume that there exists a family of adaptively secure injective one-way functions. Then, there exists an adaptively secure bit-commitment scheme. Furthermore, the commitment scheme is non-interactive.*

Further, assuming the existence of a family of adaptively secure pseudorandom generators, there exists a 2*-round adaptively secure bit-commitment scheme.*

The first of these constructions follows by replacing the injective one-way function in the Blum-Micali [9] commitment scheme, with an adaptively secure one, and the second follows from the Naor commitment scheme [22] in an analogous way.

Lemma 5 (Concatenation Lemma) *If there is an adaptively secure family of bit-commitment schemes, then there is an adaptively secure family of string-commitment schemes.*

The concatenation lemma follows by simply committing to each bit of the message independently using a single-bit commitment scheme $\textsc{Com}_{\mathsf{tag}}$.

Lemma 6 *Let* $\{\textsc{Com}_{\mathsf{tag}}\}_{\mathsf{tag}\in\{0,1\}^n}$ *be a tag-based adaptively secure commitment scheme Then, there is a non-malleable commitment scheme* $\textsc{Com}'$.

This is a standard proof using signature schemes to compile an adaptively secure commitment scheme to a plain non-malleable commitment scheme. Putting together, these lemmas prove theorems 1 and 2.

4 Four-Round Non-Malleable Zero-Knowledge

In this section, we present a 4-round non-malleable zero-knowledge argument system. The argument system is exactly the Feige-Shamir protocol [17], compiled with an adaptively secure commitment scheme. In our analysis we rely on the following properties of the Feige-Shamir protocol:

- The first property is that the first prover message is (perfectly) independent of the witness used by the prover (and even the statement!). This property has previously been used to simplify analysis, but here we inherently rely on this property to *enable* our analysis.
- The second property is that given a random accepting transcript, and the *openings* of the commitments in the first message, it is possible to "extract a witness". In other words, any transcript implicitly defines a witness; additionally, given a random transcript, this witness will be valid with a high probability (if the transcript is accepting).

In what follows, we present a sketch of the protocol and the proof and refer the reader to Appendix C for the full text of the protocol description and the proof.

4.1 An Adaptively Secure Witness Indistinguishable Proof of Knowledge

The main component in the NMZK protocol is a three-round witness-indistinguishable proof of knowledge (WIPOK) Π. The protocol is simply a parallelization of the 3-round zero-knowledge proof $\tilde{\Pi}$ for the $\mathcal{NP}$-complete language of Hamiltonicity [8, 17], with the only change that the commitment scheme used in the proof is adaptively secure.

In fact, we construct a family of protocols Π_{tag}, indexed by tag. The protocol Π_{tag} – which is a parallel repetition of $\tilde{\Pi}_{\mathsf{tag}}$ – has an adaptive WI property which, roughly stated, means that the transcripts of the protocol when the prover uses two different witnesses w_1 and w_2 are computationally indistinguishable, even if the distinguisher has access to a magic oracle that inverts all commitments $\textsc{Com}_{\mathsf{tag}'}$, where $\mathsf{tag}' \neq \mathsf{tag}$.

Lemma 7 *Let τ_w denote a random transcript of the protocol Π_{tag} between the prover P and the verifier V on common input x and where the prover has auxiliary input w and the verifier has auxiliary input z. Then, for every x, z and w, w' such that $R_L(w) = R_L(w') = 1$, and for every PPT machine D with oracle access to $\mathcal{O}(\mathsf{tag}, \cdot, \cdot)$, the following quantity is negligible in k:*

$$\left| \Pr[D^{\mathcal{O}(\mathsf{tag},\cdot,\cdot)}(x, z, w, w', \tau_w) = 1] - \Pr[D^{\mathcal{O}(\mathsf{tag},\cdot,\cdot)}(x, z, w, w', \tau_{w'}) = 1] \right|$$

The messages in the three rounds of the protocol $\tilde{\Pi}_{\mathsf{tag}}$ will be denoted A, C and Z respectively. It turns out that with probability $\frac{1}{2}$ over the choice of randomness in the protocol, a transcript of $\tilde{\Pi}_{\mathsf{tag}}$ uniquely defines a witness (even though not it is not computable in polynomial-time). We define this to be the *witness implicit in the transcript* in an instance of Π_{tag}. Furthermore, we show that the implicit witness in Π_{tag} is computable given access to $\mathcal{O}(\mathsf{tag}', \cdot, \cdot)$ for any $\mathsf{tag}' \neq \mathsf{tag}$. For more precise definitions, see Appendix C.

Lemma 8 *Given oracle access to the commitment-inversion oracle $\mathcal{O}(\mathsf{tag}', \cdot, \cdot)$ for an $\mathsf{tag}' \neq \mathsf{tag}$, the witness implicit in any accepting transcript of $\tilde{\Pi}_{\mathsf{tag}}$ can be computed in polynomial time.*

4.2 The Non-Malleable Zero-Knowledge Argument System

The non-malleable ZK protocol consists of two instances of the protocol Π_{tag} running in conjunction, one of them initiated by the verifier and the other initiated by the prover. We will denote the copy of Π_{tag} initiated by the verifier as Π^V_{tag} and the one initiated by the prover as Π^P_{tag}. We will use the notation from the previous subsection to describe the messages in these protocols – the messages in the protocol Π^V_{tag} (resp. Π^P_{tag}) appear with a superscript of V (resp. P).

Theorem 9 *Assume that* $\textsc{Com}$ *is a non-interactive adaptively secure commitment scheme. Then, the protocol in Figure 1 is a 4-round non-malleable zero-knowledge argument system.*

Proof: Completeness, soundness and zero-knowledge properties of the protocol follow directly from the corresponding properties of the Feige-Shamir protocol. In Lemma 10, we show that the protocol non-malleable.

In other words, for every man-in-the-middle adversary A that interacts with the prover P_{tag} on a statement x and convinces the verifier $V_{\mathsf{tag}'}$ (for a $\mathsf{tag}' \neq \mathsf{tag}$) in a right-interaction on a statement x' (possibly the same as x), we construct a stand-alone prover that convinces the verifier on x' with the same probability as A, but *without access to the left-interaction*. The construction of the stand-alone prover in the proof of non-malleability (see Lemma 10) relies on the adaptive security of the commitment scheme $\textsc{Com}_{\mathsf{tag}}$. It is important to note that the stand-alone prover itself runs in classical polynomial-time, and in particular does not use any oracles. Access to the commitment-inversion oracle is used only to show that the stand-alone prover works as expected (and in particular, that it convinces the verifier with the same probability as does the MIM adversary). □

Lemma 10 *The protocol* $\mathrm{NM}_{\mathsf{tag}}$ *in Figure 1 is non-malleable.*

Proof: For every man-in-the-middle adversary A, we construct a stand-alone prover S that convinces the verifier with essentially the same probability that A does. Very roughly, the construction of the stand-alone prover S proceeds in two steps.

Non-Malleable Zero-Knowledge Argument $\mathrm{NM}_{\mathsf{tag}}$

COMMON INPUT: An instance $x \in \{0,1\}^n$, presumably in the language L.

PROVER INPUT: A witness w such that $(x, w) \in R_L$.

ROUND 1: **(Verifier)** Pick w_1 and w_2 at random and compute $x_i = f(w_i)$ for $i \in \{1, 2\}$.

Let the $\mathcal{NP}$-relation $R_V = \{((x_1, x_2), w') \mid \text{ either } f(w') = x_1 \text{ or } f(w') = x_2\}$.

Initiate the WI protocol Π^V_{tag} with the statement $(x_1, x_2) \in L_V$. In particular,

$\mathbf{V} \to \mathbf{P}$: Send (x_1, x_2) to P. Send $A^V_1, A^V_2, \ldots, A^V_n$ to P.

ROUND 2: **(Prover)** Let the $\mathcal{NP}$-relation R_P be

$$\{((x, x_1, x_2), w) \mid \text{either } (x, w) \in R_L \text{ or } f(w) = x_1 \text{ or } f(w) = x_2\}$$

Initiate a WI protocol Π^P_{tag} with common input (x, x_1, x_2). Also, send the second-round messages of the protocol Π^V_{tag}. In particular,

(2a) $\mathbf{P} \to \mathbf{V}$: Send $A^P_1, A^P_2, \ldots, A^P_n$ to V.

(2b) $\mathbf{P} \to \mathbf{V}$: Send $C^V_1, C^V_2, \ldots, C^V_n$ to V.

ROUND 3: **(Verifier)** Send round-2 challenges of the protocol Π^P_{tag} and round-3 responses of Π^V_{tag}.

(3a) $\mathbf{V} \to \mathbf{P}$: Send $C^P_1, \ldots, C^P_n$ to P.

(3b) $\mathbf{V} \to \mathbf{P}$: Send $Z^V_1, \ldots, Z^V_n$ to P.

ROUND 4: **(Prover)** P verifies that the transcript $\{(A^V_i, C^V_i, Z^V_i)\}_{i \in [n]}$ is accepting for the subprotocol Π^V_{tag}. If not, abort and send nothing to V. Else,

$\mathbf{P} \to \mathbf{V}$: Send $Z^P_1, \ldots, Z^P_n$ to V.

V accepts iff the transcript $\{(A^P_i, C^P_i, Z^P_i)\}_{i \in [n]}$ is accepting for the subprotocol Π^P_{tag}.

Figure 1: NON-MALLEABLE ZERO-KNOWLEDGE PROTOCOL $\mathrm{NM}_{\mathsf{tag}}$ FOR A LANGUAGE L

1. Run the adversary A with "honestly generated" verifier-messages on the right interaction, and extract the witness for the WIPOK Π^V_{tag} that the adversary initiates on the left interaction.
2. Use the witness thus obtained to simulate the left-interaction of the adversary A and rewind the WI proof of knowledge $\Pi^P_{\mathsf{tag}'}$ it initiates on the right interaction to extract the witness for the statement x'.

Carrying out this agenda involves a number of difficulties. We first describe how to accomplish Step 1. This is done by invoking the simulator for the Feige-Shamir protocol, and is described below. Informally, S extracts the witness w' that the MIM A uses in the subprotocol Π^V_{tag} in the left-interaction. Then, S acts as the honest prover using the witness w' in the protocol Π^P_{tag}.

We now describe how to carry out Step 2 of the agenda, and show that at the end of Step 2, S extracts a witness for the statement $\tilde{x}$ that the MIM adversary A uses in the right-interaction with essentially the same probability that A convinces the verifier on the right-interaction. S starts by running the protocol in the left-interaction using the witness w' it extracted using the strategy in Step 1. Consider the moment when A outputs the first message on the left (that is, the first message in the subprotocol Π^V_{tag}). Consider two cases.
First Case: At this time, A has not yet received the round-3 messages in the right interaction (that is, the challenges in the subprotocol $\Pi^P_{\mathsf{tag}'}$) (See Figure 2(i)). In this case, the Round-1 message that A sends on the

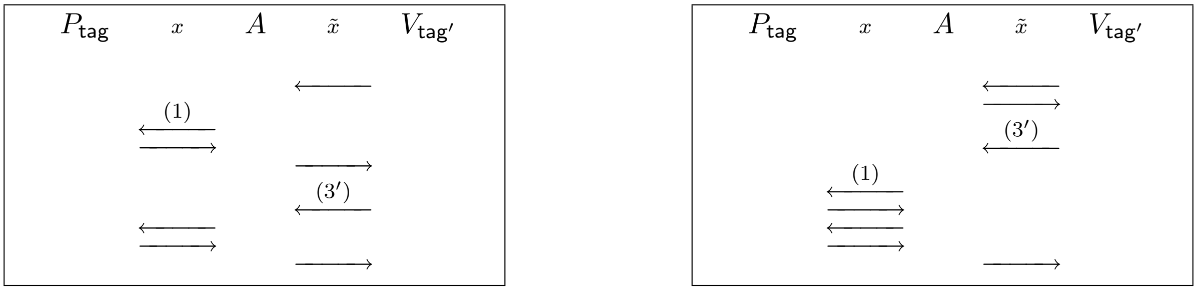


Figure 2: Two scheduling strategies (i) on the left and (ii) on the right

left interaction is independent of the Round-3 message in the right interaction. Now, S proceeds as follows: S runs the left-interaction as a normal prover P_{tag} would with the fake-witness w', and rewinds the protocol $\Pi^{P}_{\mathsf{tag}'}$ on the right-interaction to extracta witness for the statement $\tilde{x}$. Since the rewinding process does not change the messages in the right-interaction before round 3, S can use w' to produce the left-interaction just as an honest prover with witness w' would.

Second Case: A has already received the challenges in the subprotocol $\Pi^{P}_{\mathsf{tag}'}$ in the right interaction (See Figure 4(ii)). In this case, trying to rewind in the WIPOK $\Pi^{P}_{\mathsf{tag}'}$ on the right is problematic, since A could change the first message on the left, every time it is fed with a different challenge in round-3 on the right-interaction. In this case, S proceeds as follows: Every time the extractor for the WIPOK $\Pi^{P}_{\mathsf{tag}'}$ in the right-interaction rewinds, S repeats the entire procedure in Step 1 of the agenda to extract a witness w' corresponding to the (potentially new) Round-1 message in the left interaction. S then simulates the left-interaction with the witness thus extracted. The extraction procedure on the right-interaction is unaffected by the rewinding on the left.

Correctness. First, we show that the view generated by S following Step 1 of the agenda is indistinguishable from the view of A in a real interaction, even to a distinguisher that has access to the oracle $\mathcal{O}(\mathsf{tag}, \cdot, \cdot)$ that inverts $\textsc{Com}_{\mathsf{tag}'}$ for any $\mathsf{tag}' \neq \mathsf{tag}$ (Claim 1) Then, we use this to show that the *implicit witness* in the transcript of the subprotocol $\Pi^{P}_{\mathsf{tag}'}$ in the right-interaction is indistinguishable between the simulated and the real execution (Claim 2). This means that (1) the extraction on the right succeeds with essentially the same probability that A manages to convince the verififer $V_{\mathsf{tag}'}$ in the right-interaction, and moreover, (2) the witness that S extracts from the right interaction of A is computationally indistinguishable from the witness that A uses in the real interaction. Together, these claims imply the correctness of the stand-alone prover S. Note that the simulator is entirely classical, with no oracle access; the adaptive security of the commitment scheme is used only in Claim 2.

Running Time. Let X_1 be the random variable representing the number of times S has to rewind the protocol Π^{V}_{tag} in the left-interaction to extract a fake witness. Similarly, let X_2 be the random variable representing the number of times the extraction procedure on the subprotocol $\Pi^{P}_{\mathsf{tag}'}$ on the right interaction has to rewind to extract a witness.

The problematic case is the second one (Figure 2(ii)) where the total expected running time is $X = X_1 X_2$, since every time the extraction procedure on the right rewinds, S has to extract a new fake witness in the left-interaction. Thus, $E[X_1 X_2] = \sum_{a \in \mathbb{Z}^+} a \Pr[X_1 = a] E[X_2 | X_1 = a]$. However, noting that the number of times the extractor needs to rewind on the right is *independent of* the number of times the simulator rewinds on the left-interaction, we get that this is simply $E[X_1]E[X_2]$, which is polynomial. □

References

[1] Boaz Barak. Constant-round coin-tossing with a man in the middle or realizing the shared random string model. In *FOCS*, pages 345–355, 2002.

[2] Mihir Bellare, Alexandra Boldyreva, and Adriana Palacio. An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. In *EUROCRYPT*, pages 171–188, 2004.

[3] Mihir Bellare, Russell Impagliazzo, and Moni Naor. Does parallel repetition lower the error in computationally sound protocols? In *FOCS*, pages 374–383, 1997.

[4] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The power of rsa inversion oracles and the security of chaum's rsa-based blind signature scheme. In *Financial Cryptography*, pages 319–338, 2001.

[5] Mihir Bellare and Gregory Neven. Transitive signatures based on factoring and rsa. In *ASIACRYPT*, pages 397–414, 2002.

[6] Mihir Bellare and Adriana Palacio. Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In *CRYPTO*, pages 162–177, 2002.

[7] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *First ACM Conference on Computer and Communications Security*, pages 62–73, Fairfax, 1993. ACM.

[8] Manuel Blum. How to prove a theorem so no one can claim it. In *Proc. of The International Congress of Mathematicians*, pages 1444–1451, 1986.

[9] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo random bits. In *FOCS*, pages 112–117, 1982.

[10] Dan Boneh. The decision diffie-hellman problem. In *ANTS*, pages 48–63, 1998.

[11] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO*, pages 213–229, 2001.

[12] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information". In Burt Kaliski, editor, *Proceedings CRYPTO '97*, pages 455–469. Springer-Verlag, 1997. Lectures Notes in Computer Science No. 1294.

[13] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.

[14] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO*, pages 13–25, 1998.

[15] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT*, pages 45–64, 2002.

[16] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000.

[17] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *STOC*, pages 416–426, 1990.

[18] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM J. Comput.*, 25(1):169–192, 1996.

[19] Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the fiat-shamir paradigm. In *FOCS*, pages 102–, 2003.

[20] Jonathan Katz and Hoeteck Wee. Black-box lower bounds for non-malleable protocols. 2007.

[21] Tal Malkin, Ryan Moriarty, and Nikolai Yakovenko. Generalized environmental security from number theoretic assumptions. In *TCC*, pages 343–359, 2006.

[22] Naor. Bit commitment using pseudorandomness. *J. of Cryptology*, 4, 1991.

[23] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC '90: Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 427–437, New York, NY, USA, 1990. ACM Press.

[24] Moni Naor. On cryptographic assumptions and challenges. In *CRYPTO*, pages 96–109, 2003.

[25] Rafael Pass and Alon Rosen. Concurrent non-malleable commitments. In *FOCS*, pages 563–572, 2005.

[26] Rafael Pass and Alon Rosen. New and improved constructions of non-malleable cryptographic protocols. In *STOC*, pages 533–542, 2005.

[27] Krzysztof Pietrzak and Douglas Wikström. Parallel repetition of computationally sound protocols revisited. In *TCC*, pages 86–102, 2007.

[28] Manoj Prabhakaran and Amit Sahai. New notions of security: achieving universal composability without trusted setup. In *STOC*, pages 242–251, 2004.

[29] Charles Rackoff and Daniel R. Simon. Cryptographic defense against traffic analysis. In *STOC '93: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 672–681, New York, NY, USA, 1993. ACM Press.

[30] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS*, pages 543–553, 1999.

A Preliminaries and Standard Definitions

Concurrent Non-Malleable Commitment. Our definition of concurrent non-malleable commitment is almost identical to that of [26]; the main difference is that we use a definition of non-malleability w.r.t tags. Let $\textsc{Com}_{\mathsf{tag}} = \langle C_{\mathsf{tag}}, R_{\mathsf{tag}} \rangle$ be a family of commitment schemes. Consider man-in-the-middle adversaries that are participating in left and right interactions in which $m = \mathrm{poly}(n)$ commitments take place. We compare between a *man-in-the-middle* and a *simulated* execution. In the man-in-the-middle execution, the adversary A is simultaneously participating in m left and right commitments. A receives commitments to values $v_1, \ldots, v_m$ using identities $\textsc{tag}_1, \ldots, \textsc{tag}_m$ of its choice and attempts to commit to a sequence of related values $\tilde{v}_1, \ldots, \tilde{v}_m$ using identities $\tilde{\text{TAG}}_1, \ldots, \tilde{\text{TAG}}_m$. If any of the right commitments generated by the adversary are invalid, or undefined, its value is set to $\perp$. For any i such that $\tilde{\text{TAG}}_i = \text{TAG}_j$ for some j, set $\tilde{v}_i = \perp$. That is, any commitment where the adversary uses the same identity as one of the honest committers is considered invalid.

Let $\mathsf{mim}^A_{\langle C,R\rangle}(v_1, \ldots, v_m, z)$ denote a random variable that describes the values $\tilde{v}_1, \ldots, \tilde{v}_m$ and the view of A in the above experiment. In the simulated execution, the values $v_1, \ldots, v_m$ are chosen prior to the interaction but the simulator S gets nothing. We let $\mathsf{sta}^S_{\langle C,R\rangle}(1^n, z)$ denote a random variable that describes the values committed to in the output of S (which consists of a sequence of values $\tilde{v}_1, \ldots, \tilde{v}_m$) together with the view of S; as before, whenever the view contains a right interaction where the identity is the same as any of the left interactions, $\tilde{v}_i$ is set to $\perp$.

Definition 4 (Concurrent Non-Malleable Commitment [25]) *A commitment scheme* COM *is said to be* concurrent non-malleable with respect to commitment *if for every polynomial $p(\cdot)$, and every PPT A that participates in at most $m = p(n)$ commitments, there exists a PPT simulator S such that the ensembles $\left\{\mathsf{mim}^A_{\langle C,R\rangle}(v_1, \ldots, v_m, z)\right\}_{v_1,\ldots,v_m,z}$ and $\left\{\mathsf{sta}^S_{\langle C,R\rangle}(1^n, z)\right\}_{1^n,z}$ are computationally indistinguishable.*

Non-Malleable Zero-Knowledge. We consider a family of interactive proofs, where each member of the family is labeled with a tag string $\text{TAG} \in \{0,1\}^m$, and $m = m(n)$ is a parameter that potentially depends on the length of the common input (security parameter) $n \in N$. We consider a MIM adversary A that is simultaneously participating in a left and a right interaction. In the left interaction, A is verifying the validity of a statement x by interacting with a prover P_{TAG} while using a protocol that is labeled with a string TAG. In the right interaction A proves the validity of a statement $\tilde{x}$ to the honest verifier $V_{\tilde{\text{TAG}}}$ while using a protocol that is labeled with a string $\tilde{\text{TAG}}$. Let $\mathsf{mim}^A_V(\text{TAG}, \tilde{\text{TAG}}, x, \tilde{x}, w, z)$ be a random variable describing the the output of V in the man-in-the-middle experiment.

In the stand-alone execution only one interaction takes place. The stand-alone adversary S directly interacts with the honest verifier V. As in the man-in-the-middle execution, V receives as input a tag $\tilde{\text{TAG}}$ and an instance $\tilde{x}$. S receives instances $x, \text{TAG}, \tilde{x}, \tilde{\text{TAG}}$ and auxiliary input z. Let $\mathsf{sta}^S_V(\text{TAG}, \tilde{\text{TAG}}, x, \tilde{x}, z)$ be a random variable describing the the output of V in the above experiment when the random tapes of S and V are uniformly and independently chosen.

The formal definition of non-malleability is as follows.

Definition 5 (Tag-based non-malleable interactive proofs) *A family of interactive proofs $\langle P_{\text{TAG}}, V_{\text{TAG}}\rangle$ for a language L is said to be* non-malleable with respect to tags of length m *if for every probabilistic polynomial time man-in-the-middle adversary A, there exists a probabilistic expected polynomial time stand-alone prover S and a negligible function $\nu : N \to N$, such that for every $(x, w) \in L \times R_L(x)$, every $\tilde{x} \in \{0,1\}^{|x|}$, every $\text{TAG}, \tilde{\text{TAG}} \in \{0,1\}^m$ so that $\text{TAG} \neq \tilde{\text{TAG}}$, and every $z \in \{0,1\}^*$:*

$$\Pr\Big[\mathsf{mim}^A_V(\text{TAG}, \tilde{\text{TAG}}, x, \tilde{x}, w, z) = 1\Big] \; < \; \Pr\Big[\mathsf{sta}^S_V(\text{TAG}, \tilde{\text{TAG}}, x, \tilde{x}, z) = 1\Big] \; + \; \nu(|x|)$$

Non-malleable Zero-Knowledge. Non-malleable $\mathcal{ZK}$ proofs are non-malleable interactive proofs that additionally satisfy the $\mathcal{ZK}$ property.

B Non-Malleable Commitment Schemes

First, we present two constructions of adaptively secure bit-commitment schemes – the first construction assumes adaptively secure injective one-way functions (subsection B.1) and the second assumes adaptively secure PRGs (subsection B.2). The first construction is *non-interactive* and the second is a 2-round commitment scheme. Next, we show a simple "concatenation lemma", that constructs an adaptively secure string-commitment scheme from an adaptively-secure bit-commitment scheme(subsection B.3). Finally, we show that an adaptively secure commitment scheme is also non-malleable (subsection B.4).

B.1 Construction from Adaptively Secure Injective One-Way Functions

In this subsection, we present a construction of an adaptively secure bit-commitment scheme, given a family of adaptively secure injective one-way functions. The construction is the same as Blum commitment, with a slight twist: instead of xor-ing the hardcore bit with the input bit, we make sure (by the choice of randomness to the Goldreich-Levin predicate) that the hardcore bit is the same as the bit to be committed. This is to prevent an obvious malleability attack.

Lemma 11 *Assume that there exists a family of adaptively secure injective one-way functions. Then, there exists an adaptively secure bit-commitment scheme. Furthermore, the commitment scheme is non-interactive.*

Proof: Let $\mathcal{F}_n = \{f_{\mathsf{tag}} : \{0,1\}^n \mapsto \{0,1\}^{m(n)} \mid \mathsf{tag} \in \{0,1\}_n\}$ be a family of injective one-way functions. The commitment scheme $\textsc{Com}_{\mathsf{tag}}$ is constructed as follows. To commit a bit b, $\textsc{Com}$ picks a random $x \in \{0,1\}^n$ and a random $r \in \{0,1\}^n$ subject to the condition that $\langle r, x\rangle = b$. Compute $y = f_{\mathsf{tag}}(x)$. The commitment is (y, r).

Since each f_{tag} is injective, the commitment is perfectly binding.

Assume, for contradiction, that there is an adversary A that breaks the adaptive security of $\textsc{Com}$. Then, we construct a PPT adversary B that breaks the adaptive security of $\mathcal{F}$. B will use an intermediate adversary B', which on input (tag, y, r) computes the Goldreich-Levin hardcore bit $\langle f_{\mathsf{tag}}^{-1}(y), r\rangle$ with non-negligible probability, with access to the oracle $\mathcal{O}(\mathsf{tag}, \cdot, \cdot)$ that on input (tag', y') returns $f_{\mathsf{tag}'}^{-1}(y')$ if $\mathsf{tag}' \neq \mathsf{tag}$ and $\perp$ otherwise.

B' works as follows: On input (tag, y, s), B' runs A with input (tag, y, s). Note that (tag, y, s) is a commitment to a random bit. When A asks a query (tag', y', s') (where $\mathsf{tag}' \neq \mathsf{tag}$) B' uses the inversion oracle for the one-way function to compute $x' = f_{\mathsf{tag}'}^{-1}(y')$ and returns the inner product $\langle x', s'\rangle$ to A. Finally, B' outputs whatever A outputs.

It is easy to see that B' perfectly simulates the view of the adversary A where the challenge to A is a random commitment to the bit $b = \langle f_{\mathsf{tag}}^{-1}(y), s\rangle$. B' predicts the Goldreich-Levin hardcore bit with the same probability that A predicts the committed bit.

The construction of B from B' is exactly the same as the Goldreich-Levin construction of a one-way function inverter from a hardcore-bit predictor: one only needs to observe that the Goldreich-Levin construction is black-box and uses only queries to the hardcore-bit predictor on the same index tag, to invert the function f_{tag}. □

B.2 Construction from Adaptively Secure Pseudorandom Generators

We will construct a two-round statistically binding, adaptively secure bit-commitment scheme from adaptively secure pseudorandom generators (PRG). The construction is exactly the same as Naor commitment, except that we use a family of adaptively secure PRGs.

Lemma 12 *Assume that there exists a family of adaptively secure pseudorandom generators. Then, there exists a* 2*-round adaptively secure bit-commitment scheme.*

Proof: Let $\mathcal{G}_n = \{G_{\mathsf{tag}} : \{0,1\}^n \mapsto \{0,1\}^{m(n)} \mid \mathsf{tag} \in \{0,1\}_n\}$ be a family of adaptively secure pseudorandom generators, where $m(n) \geq 3n$. The commitment scheme $\text{COM}_{\mathsf{tag}}$ is constructed as follows. To commit to a bit b, the sender S and receiver R run the following protocol.

1. R picks a random string $r \in \{0,1\}^{m(n)}$ and sends it to S.
2. S picks a random string $s \in \{0,1\}^n$ and sends to R the value $y = G_{\mathsf{tag}}(s)$ if $b = 0$ and $y = r \oplus G_{\mathsf{tag}}(s)$ if $b = 1$.

The statistical binding of this commitment scheme follows by the same argument as that for the Naor commitment.

Assume, for contradiction, that there is an adversary A that breaks the adaptive security of COM. Then, we construct a PPT adversary B that breaks the adaptive security of the PRG $\mathcal{G}$.

On input (tag, y), B runs A until it returns the first-round message r for a commitment. B picks a random bit b: If $b = 0$, it returns (tag, y) to A. If $b = 1$, it returns $(\mathsf{tag}, y \oplus r)$ to A. When A asks B for a decommitment on a transcript (tag', r', y') (where $\mathsf{tag} \neq \mathsf{tag}'$), B does the following: Use the adaptive PRG-oracle to determine which of y' or $y' \oplus r'$ is in the range of $G_{\mathsf{tag}'}$. If y' is in the range, return 0, else if $y' \oplus r'$ is in the range, return 1. If both are in the range, return a random bit and if neither is in the range, return $\perp$. Finally, A returns a bit b'. B outputs 0 if $b = b'$ and 1 otherwise.

It is easy to see that B simulates the answers to the decommitment queries perfectly[6]. If y is pseudorandom, then the commitment returned by B is distributed exactly like a commitment to the bit b. On the other hand, if y is random, then the commitment returned by B is independent of the bit b, which means that A cannot guess the bit b with probability better than $\frac{1}{2}$. From this fact, it follows by a standard argument that B distinguishes between the case where y is random and when it is pseudorandom, if A predicts correctly. □

B.3 Concatenation Lemma

In this section, we show a "concatenation lemma", which gives a way to construct an adaptively secure string-commitment scheme from an adaptively-secure bit-commitment scheme.

Lemma 13 *If there is an adaptively secure family of bit-commitment schemes, then there is an adaptively secure family of string-commitment schemes.*

Proof: Given an adaptively secure bit commitment scheme $\{\text{COM}'_{\mathsf{tag}}\}_{\mathsf{tag} \in \{0,1\}^n}$, we construct an adaptively secure string-commitment scheme $\{\text{COM}_{\mathsf{tag}}\}_{\mathsf{tag} \in \{0,1\}^n}$ as follows: To commit to a string $m = m_1 \ldots m_\ell$, $\text{COM}_{\mathsf{tag}}$ runs many instances of $\text{COM}'_{\mathsf{tag}}$, in parallel. More precisely, COM does the following: Run $\text{COM}_{\mathsf{tag}}(m_1), \ldots, \text{COM}_{\mathsf{tag}}(m_\ell)$ and output the concatenation of the ℓ commitments.

Assume, for contradiction, that there is a PPT adversary A that breaks the adaptive security of COM. Then, we construct a PPT adversary B that breaks the adaptive security of COM'. The proof will essentially follow by a hybrid argument. B runs A until it produces two messages $\bar{m}_0 = m_0^1 \ldots m_0^\ell$ and $\bar{m}_1 = m_1^1 \ldots m_1^\ell$. B picks a random $i \in [\ell]$, internally simulates the commitment interactions of COM' using the tag tag and on input bit m_j^0, for $j < i$ and for input bit m_j^1 for $j > i$. For $j = i$, B runs a commitment instance of COM' on tag tag with input bit m_i interacting with the outside and forwarding the messages to and from the adversary A.

[6] In the case that the commitment transcript is ambiguous, namely it is a valid commitment to both 0 and 1, B returns a random bit, which is the expected behavior (See the definition of the commitment-inversion oracle)

To answer the commitment-inversion queries of A, note that the legal such queries contain a tag $\mathsf{tag}' \neq \mathsf{tag}$ and thus, B can obtain their decommitments from the commitment inversion oracle for $\textsc{Com}'$. The proof now follows by a simple hybrid argument, and is omitted here. □

B.4 Non-Interactive Concurrently Non-Malleable Commitment

In this section, we show that adaptively secure commitment schemes are also non-malleable. Furthermore, the first of our commitment schemes (based on adaptive injective one-way functions) actually yields a non-interactive concurrently non-malleable commitment scheme. This proves Theorems 1 and 2.

Lemma 14 *Let* $\{\textsc{Com}_{\mathsf{tag}}\}_{\mathsf{tag}\in\{0,1\}^n}$ *be a tag-based adaptively secure commitment scheme Then, there is a non-malleable commitment scheme* $\textsc{Com}'$.

Proof:(Sketch.) The commitment scheme $\textsc{Com}'$ works as follows: to commit to a message m, it first picks a pair $(\textsc{vk}, \textsc{sk})$ for a one-time signature scheme. Then, it runs $\textsc{Com}_{\textsc{VK}}(m)$ and signs the resulting commitment using the signature key $\textsc{sk}$.

We will first show that $\textsc{Com}'$ is a *one-many* non-malleable commitment scheme. That is, the man-in-the-middle adversary only gets one commitment from the left interaction (and he can produce many commitments on the right to “related messages”). Then, we will use a proposition of Pass and Rosen [26] that shows that any commitment scheme that is one-many non-malleable is also fully (many-many) non-malleable.

To show the first part: for every A that receives one commitment on the left and produces many commitments, we construct a simulator S that, without receiving the commitment on the left, commits to values that are indistinguishable from what A committed to. S computes a key-pair $(\textsc{vk}, \textsc{sk})$ for the one-time signature scheme and feeds A a commitment $c = \textsc{Com}_{\textsc{VK}}(0^\ell)$ and outputs whatever A outputs.

Suppose there is a distinguisher D that distinguishes between the values that A and S committed to. Then, we produce an algorithm D' that breaks the adaptive security of the commitment scheme. D' gets a commitment on the index $\textsc{vk}$, either to a random value x or to 0^n from the outside. It runs A with this commitment, and obtains the sequence of commitments $(c_1, c_2, \ldots, c_n)$ that A outputs. Each of these commitments c_i uses $\mathsf{tag}_i \neq \textsc{vk}$ since otherwise, it is possible to break the security of the signature scheme. On the other hand, whenever a commitment uses an $\mathsf{tag}_i \neq \textsc{vk}$, D' can use the commitment-inversion oracle on tag_i to compute the message underlying c_i, and use D to distinguish between the messages. □

Proposition 15 ([26]) *Any commitment scheme that is one-many non-malleable is also concurrently non-malleable.*

C Four-Round Non-Malleable Zero-Knowledge

In this section, we present a 4-round non-malleable zero-knowledge argument system. The argument system is exactly the Feige-Shamir protocol [17], compiled with an adaptively secure commitment scheme. In our analysis we rely on the following properties of the Feige-Shamir protocol:

- The first property is that the first prover message is (perfectly) independent of the witness used by the prover (and even the statement!). This property has previously been used to simplify analysis, but here we inherently rely on this property to *enable* our analysis.

- The second property is that given a random accepting transcript, and the *openings* of the commitments in the first message, it is possible to “extract a witness”. In other words, any transcript implicitly defines a witness; additionally, given a random transcript, this witness will be valid with a high probability (if the transcript is accepting).

In what follows, we present a sketch of the protocol and the proof and refer the reader to Appendix C for the full text of the protocol description and the proof.

C.1 An Adaptively Secure Witness Indistinguishable Proof of Knowledge

In this section, we describe a three-round witness-hiding proof of knowledge (WIPOK) Π, which is used as a building block in the non-malleable ZK protocol, and state or prove some facts about the WIPOK. The protocol is a parallelization of the 3-round zero-knowledge proof ρ for the $\mathcal{NP}$-complete language of Hamiltonicity [8, 17], with the only change that the commitment scheme used in the proof is adaptively secure.

In fact, we construct a family of protocols Π_{tag}, indexed by tag. The protocols Π_{tag} has an adaptive WI property which, roughly stated, means that the transcripts of the protocol when the prover uses two different witnesses w_1 and w_2 are computationally indistinguishable, even if the distinguisher has access to a magic oracle that inverts all commitments $\textsc{Com}_{\mathsf{tag}'}$, where $\mathsf{tag}' \neq \mathsf{tag}$.

The protocol Π_{tag} is simply a basic 3-round WI protocol $\tilde{\Pi}_{\mathsf{tag}}$ repeated k times in parallel (where k is a security parameter), where the prover and the verifier choose independent random bits for each instance of $\tilde{\Pi}_{\mathsf{tag}}$. We now describe how $\tilde{\Pi}_{\mathsf{tag}}$ works. The common input is a graph G on n vertices and the string tag. The auxiliary input to the prover is a Hamiltonian cycle in G. The protocol uses an adaptively secure family of commitment schemes $\{\textsc{Com}_{\mathsf{tag}}\}_{\mathsf{tag}\in\{0,1\}^k}$. The messages in the three rounds of the protocol $\tilde{\Pi}_{\mathsf{tag}}$ will be denoted A, C and Z respectively. The messages in the i^{th} copy of $\tilde{\Pi}_{\mathsf{tag}}$ will be denoted A_i, C_i and Z_i. Thus, the transcript of the protocol Π_{tag} consists of $\{(A_i, C_i, Z_i)\}_{i\in[k]}$.

1. **(Round** 1**)** $P \rightarrow V$: P chooses a random n-cycle Ψ. Let $\Psi_{i,j}$ denote the $(i,j)^{th}$ entry in the adjacency matrix of Ψ. Compute n^2 commitments using $\textsc{Com}_{\mathsf{tag}}$, one for each entry of the adjacency matrix $\Psi_{i,j}$. Send the commitments to the verifier V (this message is denoted A).

2. **(Round** 2**)** $V \rightarrow P$: Sends a random bit, denoted C, to the prover P.

3. **(Round** 3**)** $P \rightarrow V$: If $C = 0$, P opens all the commitments it sent in the first round. If $C = 1$, P sends a random permutation $\pi : [n] \mapsto [n]$ that maps the cycle Ψ to the Hamiltonian cycle in G, and the decommitment of all the entries $\Psi_{i,j}$ of M such that $(\pi(i), \pi(j))$ is not an edge of the graph G. Denote the third-round message by Z.

4. **(Verifier’s Local Computation)** If $C = 0$, V checks that Z contains valid decommitments of all the commitments in A, and that the resulting decommitments form an n-node cycle. If $C = 1$, V checks that the decommitments correspond to all the non-edges of $\pi^{-1}(G)$.

Let $\mathcal{O}(\mathsf{tag}, \cdot, \cdot)$ denote an oracle that on input tag' and a commitment, inverts the commitment if $\mathsf{tag}' \neq \mathsf{tag}$, and outputs $\perp$ otherwise. The lemma below shows that the protocol Π_{tag} is witness-indistinguishable even if the distinguisher has access to the oracle $\mathcal{O}(\mathsf{tag}, \cdot, \cdot)$.

Lemma 16 *Let τ_w denote a random transcript of the protocol Π_{tag} between the prover P and the verifier V on common input x and where the prover has auxiliary input w and the verifier has auxiliary input z. Then,* *for every x, z and w, w' such that $R_L(w) = R_L(w') = 1$, and for every PPT machine D with oracle access to $\mathcal{O}(\mathsf{tag}, \cdot, \cdot)$, the following quantity is negligible in k:*

$$\big| \Pr[D^{\mathcal{O}(\mathsf{tag},\cdot,\cdot)}(x, z, w, w', \tau_w) = 1] - \Pr[D^{\mathcal{O}(\mathsf{tag},\cdot,\cdot)}(x, z, w, w', \tau_{w'}) = 1] \big|$$

Proof: We first show that the basic subprotocol $\tilde{\Pi}_{\mathsf{tag}}$ is zero-knowledge, even if the distinguisher has access to $\mathcal{O}(\mathsf{tag}, \cdot, \cdot)$. Witness-indistinguishability follows by a straightforward hybrid argument. The simulator for zero-knowledge is exactly the classical GMW zero-knowledge simulator for $\tilde{\Pi}_{\mathsf{tag}}$. The adaptive security of COM immediately implies that the output distribution of the simulator is indistinguishable from the real interaction even if the distinguisher has access to $\mathcal{O}(\mathsf{tag}, \cdot, \cdot)$. □

Now, we define the notion of a *witness implicit in the transcript* of an instance of $\tilde{\Pi}_{\mathsf{tag}}$, and show that the implicit witness in $\tilde{\Pi}_{\mathsf{tag}}$ is computable given access to $\mathcal{O}(\mathsf{tag}', \cdot, \cdot)$ for any $\mathsf{tag}' \neq \mathsf{tag}$. Consider an accepting transcript (A, C, Z) of the protocol $\tilde{\Pi}_{\mathsf{tag}}$. Note that the decommitment of the n^2 commitments in A uniquely defines an n-node graph (even though not it is not computable in polynomial-time). We observe that if (A, C, Z) is an accepting transcript, then A has to be a commitment to one of the following two graphs: either (1) G_0, an n-cycle, or (2) G_1, a (permuted) subgraph of G. Moreover, an accepting transcript where the first message is a commitment of G_0 (resp. G_1) and the challenge-bit C is 1 (resp. 0) uniquely defines a a Hamiltonian cycle in G; call this the witness implicit in the transcript. When the first message is a commitment to G_0 (resp. G_1) and the challenge-bit is 0 (resp. 1), the witness implicit in the transcript is $\perp$. Furthermore, we can recover the implicit witness, given oracle access to $\mathcal{O}(\mathsf{tag}', \cdot, \cdot)$.

Lemma 17 *Given oracle access to the commitment-inversion oracle $\mathcal{O}(\mathsf{tag}', \cdot, \cdot)$ for an $\mathsf{tag}' \neq \mathsf{tag}$, the witness implicit in any accepting transcript of $\tilde{\Pi}_{\mathsf{tag}}$ can be computed in polynomial time*

C.2 The Non-Malleable Zero-Knowledge Argument System

The non-malleable ZK protocol consists of two instances of the protocol Π_{tag} running in conjunction, one of them initiated by the verifier and the other initiated by the prover. We will denote the copy of Π_{tag} initiated by the verifier as Π^V_{tag} and the one initiated by the prover as Π^P_{tag}. We will use the notation from the previous subsection to describe the messages in these protocols – the messages in the protocol Π^V_{tag} (resp. Π^P_{tag}) appear with a superscript of V (resp. P).

Theorem 18 *Assume that* COM *is a non-interactive adaptively secure commitment scheme. Then, the protocol in Figure 3 is a 4-round non-malleable zero-knowledge argument system.*

Proof: Completeness, soundness and zero-knowledge properties of the protocol follow directly from the corresponding properties of the Feige-Shamir protocol. In Lemma 19, we show that the protocol non-malleable.

In other words, for every man-in-the-middle adversary A that interacts with the prover P_{tag} on a statement x and convinces the verifier $V_{\mathsf{tag}'}$ (for a $\mathsf{tag}' \neq \mathsf{tag}$) in a right-interaction on a statement x' (possibly the same as x), we construct a stand-alone prover that convinces the verifier on x' with the same probability as A, but *without access to the left-interaction*. The construction of the stand-alone prover (see Lemma 19) relies on the adaptive security of the commitment scheme $\text{COM}_{\mathsf{tag}}$. It is important to note that the stand-alone prover itself runs in classical polynomial-time, and in particular does not use any oracles. Access to the commitment-inversion oracle is used only to show that the stand-alone prover works as expected (and in particular, that it convinces the verifier with the same probability as does the man-in-the-middle adversary). □

Non-Malleable Zero-Knowledge Argument $\mathrm{NM}_{\mathsf{tag}}$

COMMON INPUT: An instance $x \in \{0,1\}^n$, presumably in the language L.

PROVER INPUT: A witness w such that $(x, w) \in R_L$.

ROUND 1: **(Verifier)** Pick w_1 and w_2 at random and compute $x_i = f(w_i)$ for $i \in \{1, 2\}$.

Let the $\mathcal{NP}$-relation $R_V = \{((x_1, x_2), w') \mid \text{ either } f(w') = x_1 \text{ or } f(w') = x_2\}$.

Initiate the WI protocol Π^V_{tag} with the statement $(x_1, x_2) \in L_V$. In particular,

$\mathbf{V} \rightarrow \mathbf{P}$: Send (x_1, x_2) to P. Send $A^V_1, A^V_2, \ldots, A^V_n$ to P.

ROUND 2: **(Prover)** Let the $\mathcal{NP}$-relation R_P be

$$\{((x, x_1, x_2), w) \mid \text{either } (x, w) \in R_L \text{ or } f(w) = x_1 \text{ or } f(w) = x_2\}$$

Initiate a WI protocol Π^P_{tag} with common input (x, x_1, x_2). Also, send the second-round messages of the protocol Π^V_{tag}. In particular,

(2a) $\mathbf{P} \rightarrow \mathbf{V}$: Send $A^P_1, A^P_2, \ldots, A^P_n$ to V.

(2b) $\mathbf{P} \rightarrow \mathbf{V}$: Send $C^V_1, C^V_2, \ldots, C^V_n$ to V.

ROUND 3: **(Verifier)** Send round-2 challenges of the protocol Π^P_{tag} and round-3 responses of Π^V_{tag}.

(3a) $\mathbf{V} \rightarrow \mathbf{P}$: Send $C^P_1, \ldots, C^P_n$ to P.

(3b) $\mathbf{V} \rightarrow \mathbf{P}$: Send $Z^V_1, \ldots, Z^V_n$ to P.

ROUND 4: **(Prover)** P verifies that the transcript $\{(A^V_i, C^V_i, Z^V_i)\}_{i \in [n]}$ is accepting for the subprotocol Π^V_{tag}. If not, abort and send nothing to V. Else,

$\mathbf{P} \rightarrow \mathbf{V}$: Send $Z^P_1, \ldots, Z^P_n$ to V.

V accepts iff the transcript $\{(A^P_i, C^P_i, Z^P_i)\}_{i \in [n]}$ is accepting for the subprotocol Π^P_{tag}.

Figure 3: NON-MALLEABLE ZERO-KNOWLEDGE PROTOCOL $\mathrm{NM}_{\mathsf{tag}}$ FOR A LANGUAGE L

Lemma 19 *The protocol* $\mathrm{NM}_{\mathsf{tag}}$ *in Figure 3 is non-malleable.*

Proof: For every man-in-the-middle adversary A, we construct a stand-alone prover S that convinces the verifier with essentially the same probability that A does. Very roughly, the construction of the stand-alone prover S proceeds in two steps.

1. Run the adversary A with "honestly generated" verifier-messages on the right interaction, and extract the witness for the WIPOK Π^V_{tag} that the adversary initiates on the left interaction.

2. Use the witness thus obtained to simulate the left-interaction of the adversary A and rewind the WI proof of knowledge $\Pi^P_{\mathsf{tag}'}$ it initiates on the right interaction to extract the witness for the statement x'.

Carrying out this agenda involves a number of difficulties. We first describe how to accomplish Step 1. This is done by invoking the simulator for the Feige-Shamir protocol, and is described below. Informally, S extracts the witness w' that the MIM A uses in the subprotocol Π^V_{tag} in the left-interaction. Then, S acts as the honest prover using the witness w' in the protocol Π^P_{tag}. More precisely,

1. S runs the MIM adversary A until the end of round 3 in the left-interaction (namely, until the subprotocol Π^V_{tag} finishes). In doing so, S feeds A with the messages of an honest verifier on the right-interaction.

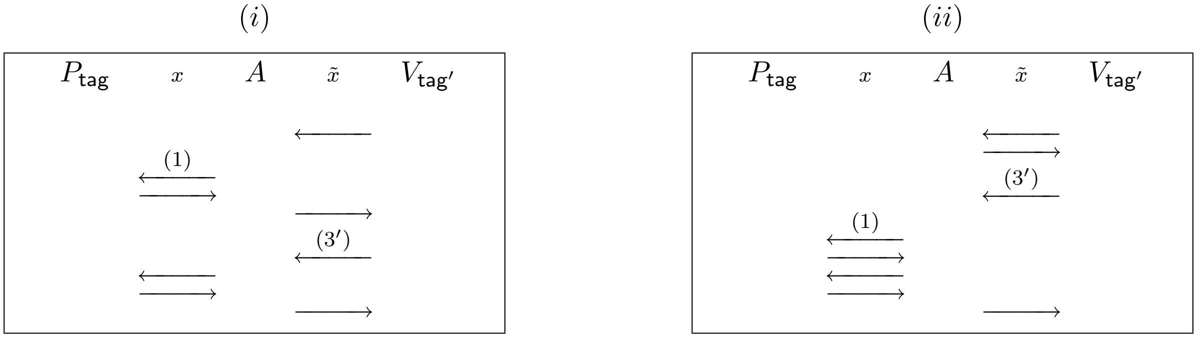
2. S rewinds A to the beginning of Round 2, that is immediately after A sends the Round 1 message in the subprotocol Π^V_{tag}. It continues running A until it produces another accepting transcript with a different sequence of challenges for Π^V_{tag} in round $2b$.

   At this point, S can extract a witness w' for the statement A uses in Π^V_{tag}. Note that during the extraction process, S sends the messages in rounds $2a$ and $3a$ as an honest prover would, and this does not require knowledge of the witness w for the statement x.

3. S now rewinds A to the beginning of Round 2, and uses w' as the witness in the subprotocol Π^P_{tag} starting from Round 2, while sending the messages in the subprotocol Π^V_{tag} as an honest prover would. In particular, S runs the WI protocol Π^P_{tag} with w' as the witness.

We now describe how to carry out Step 2 of the agenda, and show that at the end of Step 2, S extracts a witness for the statement x' that the MIM adversary A uses in the right-interaction with essentially the same probability that A convinces the verifier on the right-interaction.

S starts by running the protocol in the left-interaction using the witness w' it extracted using the strategy in Step 1 of the agenda. Consider the moment when A outputs the first message on the left (that is, the first message in the subprotocol Π^V_{tag}). Now, consider two cases.


(i)

(ii)

Figure 4: Two scheduling strategies.

1. At this time, A has not yet received the round-3 messages in the right interaction (that is, the challenges in the subprotocol $\Pi^P_{\mathsf{tag}'}$). This situation is illustrated in Figure 4(i). In this case, the Round-1 message that A sends on the left interaction is independent of the Round-3 message in the right interaction.

   Now, S proceeds as follows: S runs the left-interaction as a normal prover P_{tag} would with the fake-witness w', and rewinds the protocol $\Pi^P_{\mathsf{tag}'}$ on the right-interaction to extract a witness for the statement $\tilde{x}$. Since the rewinding process does not change the messages in the right-interaction before round 3, S can use w' to produce the left-interaction just as an honest prover with witness w' would.

2. A has already received the challenges in the subprotocol $\Pi^P_{\mathsf{tag}'}$ in the right interaction. Such a situation is illustrated in Figure 4(ii). In this case, trying to rewind in the WIPOK $\Pi^P_{\mathsf{tag}'}$ on the right is

problematic, since A could change the first message on the left, every time it is fed with a different challenge in round-3 on the right-interaction.

In this case, S proceeds as follows: Every time the extractor for the WIPOK $\Pi^P_{\mathsf{tag}'}$ in the right-interaction rewinds, S repeats the entire procedure in Step 1 of the agenda to extract a witness w' corresponding to the (potentially new) Round-1 message in the left interaction. S then simulates the left-interaction with the witness thus extracted.

Note that the round-2 message of A in the WIPOK $\Pi^P_{\mathsf{tag}'}$ on the right-interaction is fixed, and independent of the rewindings. Thus, the extraction procedure on the right-interaction is unaffected by the rewinding on the left. The views generated in the left-interaction in this process are identically distributed and thus, the probability that $V_{\mathsf{tag}'}(\tilde{x})$ accepts on any two such views is the same. Thus, sampling sufficiently many times ensures that extraction succeeds.

Correctness. First, we show that the view generated by S following Step 1 of the agenda is indistinguishable from the view of A in a real interaction, even to a distinguisher that has access to the oracle $\mathcal{O}(\mathsf{tag}, \cdot, \cdot)$ that inverts $\textsc{Com}_{\mathsf{tag}'}$ for any $\mathsf{tag}' \neq \mathsf{tag}$ (Claim 1) Then, we use this to show that the *implicit witness* in the transcript of the subprotocol $\Pi^P_{\mathsf{tag}'}$ in the right-interaction is indistinguishable between the simulated and the real execution (Claim 2). This means that (1) the extraction on the right succeeds with essentially the same probability that A manages to convince the verififer $V_{\mathsf{tag}'}$ in the right-interaction, and moreover, (2) the witness that S extracts from the right interaction of A is computationally indistinguishable from the witness that A uses in the real interaction. Together, these claims imply the correctness of the stand-alone prover S.

Claim 1 *The views generated by S in the left-interaction following the strategy in Step 1 of the agenda is indistinguishable from a real left-interaction, even to a distinguisher that has access to a commitment-inversion oracle $\mathcal{O}(\mathsf{tag}, \cdot, \cdot)$.*

Proof: The view generated by S is identical to the view of A in the real interaction, except that S uses the fake witness it extracted from the subprotocol Π^V_{tag}, instead of the a witness for the statement $x \in L$. Thus the transcript of the left-interaction corresponds to the interaction with a prover with witness w in the real left-interaction, whereas it corresponds to a prover with witness w' in the simulated left-interaction. These two ensembles are indistinguishable, even to an adversary with access to $\mathcal{O}(\mathsf{tag}, \cdot, \cdot)$, by the adaptive WI property of Π_{tag} (Lemma 16). □

The protocol $\Pi^P_{\mathsf{tag}'}$ consists of n copies of a smaller subprotocol $\tilde{\Pi}^P_{\mathsf{tag}'}$, running in parallel. Define the *witness-tuple* implicit a transcript of $\Pi^P_{\mathsf{tag}'}$ to be the n-tuple $(w_1, \ldots, w_n)$ where w_i is the witness implicit in the i^{th} copy of the smaller subprotocol $\tilde{\Pi}^P_{\mathsf{tag}'}$. We now show that the witness-tuple is computationally indistinguishable between the real and the simulated interactions. This means that the witness extracted by S in the right-interaction is computationally indistinguishable from the one that the MIM A uses in the real-interaction. By the soundness of the protocol, this has to be the witness for the statement $\tilde{x}$.

Claim 2 *The witness-tuple implicit in the transcript of the subprotocol $\Pi^P_{\mathsf{tag}'}$ (in the right-interaction) of the real and simulated executions are computationally indistinguishable.*

Proof: Suppose, for contradiction, that the two witness-tuples are distinguishable. Then, we show that the real and simulated interactions are distinguishable, given access to a commitment-inversion oracle $\mathcal{O}(\mathsf{tag}, \cdot, \cdot)$, which is impossible by Claim 1. By Lemma 17, it is easy to compute the implicit witness given a transcript of the subprotocol $\tilde{\Pi}^P_{\mathsf{tag}'}$. Thus, it is easy to compute the witness-tuple implicit in $\tilde{\Pi}^P_{\mathsf{tag}'}$, given oracle-access to $\mathcal{O}(\mathtt{tag},\cdot,\cdot)$. Thus, if the witness-tuples in the real and simulated executions are distinguishable, then the transcripts of the real and simulated executions themselves are distinguishable given oracle access to $\mathcal{O}(\mathtt{tag},\cdot,\cdot)$. □

Running Time. Let X_1 be the random variable representing the number of times S has to rewind the protocol $\Pi^V_{\mathtt{tag}}$ in the left-interaction to extract a fake witness. Similarly, let X_2 be the random variable representing the number of times the extraction procedure on the subprotocol $\Pi^P_{\mathtt{tag}'}$ on the right interaction has to rewind to extract a witness.

Consider two cases. In the first case, corresponding to Figure 4(i), the rewinding on the left interaction is done once (until a fake witness is extracted) and the rewinding on the right interaction is done once. Thus, the total expected running time is proportional to $E[X_1 + X_2] = E[X_1] + E[X_2]$, which is polynomial.

In the second case, corresponding to Figure 4(ii), the total expected running time is $X = X_1X_2$, since every time the extraction procedure on the right rewinds, S has to extract a new fake witness in the left-interaction. Thus,

$$E[X_1X_2] = \sum_{a\in\mathbb{Z}^+} a\Pr[X_1 = a]E[X_2|X_1 = a]$$

However, the number of times the extractor needs to rewind on the right is *independent of* the number of times the simulator rewinds on the left-interaction and thus $E[X_2|X_1 = a] = E[X_2]$. Thus, $\mathbb{E}[X_1X_2] = \sum_{a\in\mathbb{Z}^+} a\Pr[X_1 = a]E[X_2] = E[X_1]E[X_2]$, which is polynomial. □

D CCA-Secure Encryption Scheme

D.1 Security Against Chosen-Ciphertext Attacks

Definition 6 (IND-CCA2 Security) *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be an encryption scheme and let the random variable* $\textsc{ind}_b(\Pi, A, k)$ *where* $b \in \{0, 1\}$, $A = (A_1, A_2)$ *and* $k \in \mathbb{N}$ *denote the result of the following probabilistic experiment:*

$\textsc{ind}_b(\Pi, A, k)$:
$(\textsc{pk}, \textsc{sk}) \leftarrow \mathsf{Gen}(1^k)$
$(m_0, m_1, s) \leftarrow A_1^{O_1}(\textsc{pk})$ *s.t.* $|m_0| = |m_1|$
$y \leftarrow \mathsf{Enc}_{\mathrm{PK}}(m_b)$
$z \leftarrow A_2^{O_2}(y, s)$
Output z.

where O_1 *and* O_2 *denote the decryption oracles, and* O_2 *decrypts all ciphertexts except* y.
$(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is IND-CCA2-secure if* $\forall$ *p.p.t. algorithms* $A = (A_1, A_2)$, *the following two ensembles are computationally indistinguishable:*

$$\Big\{\textsc{ind}_0(\Pi, A, k)\Big\}_{k\in\mathbb{N}} \overset{c}{\approx} \Big\{\textsc{ind}_1(\Pi, A, k)\Big\}_{k\in\mathbb{N}} \tag{1}$$

D.2 Assumptions

In this section, we exhibit a construction of an IND-CCA2-secure encryption scheme, assuming an adaptively secure variant of perfectly one-way hash functions (defined by Canetti [12]), and a family of trapdoor permutations that are uninvertible even with access to an oracle that inverts the perfectly one-way hash function. We make the assumptions more precise below.

The construction is a modification of an encryption scheme of Bellare and Rogaway [7], which they proved to be IND-CCA2-secure in the random oracle. We show that the construction is IND-CCA2-secure under well-defined complexity-theoretic assumptions, and in particular, without assuming random oracles. Our construction is essentially as efficient as the original construction of [7]. In particular, the secret and public keys are of size n, the security parameter. The ciphertext-size is $\ell + O(n)$ to encrypt a message of size n bits. First, we state the complexity assumption, and then present the scheme that is IND-CCA2-secure under the assumption.

We define the notion of perfectly one-way hashing, in the presence of auxiliary information and oracle access. The auxiliary information is an uninvertible function g evaluated on the input r. Informally, we require that $h = H(r; s)$ be indistinguishable from random for an adversary that is given $g(r)$ as auxiliary input, and gets access to an oracle that inverts every $h' \neq h$. Namely, the oracle, given $h' \neq h$, computes r' and s' such that $h' = H(r'; s')$. Let $\mathcal{O}(h, \cdot)$ denote such an oracle.

We note that Canetti [12] (define and) use perfectly one-way hashing with auxiliary input to prove the *IND-CPA* security (semantic security) of the [7] construction.

Definition 7 (Adaptively Secure Perfectly One-way Hashing with Auxiliary Information) *A function* PHGen *is called a perfectly one-way hash function if for a random* $H \leftarrow \mathsf{PHGen}(1^k)$,

1. *H is public-coin; namely, for any input x, $H(x; r)$ contains the randomness r.*
2. *$\{H(x; r)\}_{r \in \{0,1\}^n}$ and $\{H(y; r)\}_{r \in \{0,1\}^n}$ are disjoint for all x, y, and*
3. *For every well-spread ensemble χ, for all uninvertible g and for all PPT distinguishers D, there is a negligible function μ such that*

$$\Big| \Pr_{x \leftarrow \chi; r \leftarrow U_n} [D^{\mathcal{O}(H(x;r), \cdot)}(g(x), H(x; r)) = 1] - \Pr_{x \leftarrow \chi; u \leftarrow U_n} [D^{\mathcal{O}(u, \cdot)}(g(x), u) = 1] \Big| \leq \mu(n)$$

D.3 The Construction

Theorem 20 *Assume that* TDPGen *is a family of trapdoor permutations that are uninvertible with access to the H-inverting oracle, and that* PHGen *is an adaptively secure perfectly one-way hash with auxiliary information. Then, there exists a IND-CCA2-secure encryption scheme.*

Proof: The encryption scheme is $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ as in Figure 5.

Let us fix some notations. A ciphertext $\mathbf{c}$ consists of components $[c_0, c_1, c_2, s_1, s_2]$. $\mathbf{c}$ implicitly defines the quantities $r = f^{-1}(c_0)$ and $m = H(r; s_1) \oplus c_1$.

Assume, for contradiction, that there is a PPT adversary A that wins in the IND-CCA2 game. Then, we construct a PPT adversary B that breaks the adaptive perfectly one-way hash function PHGen. Let $\mathbf{c}^*$ denote the challenge ciphertext, which defines quantities $c_0^*, c_1^*, c_2^*, s_1^*, s_2^*, r^*$ and m^* as above. B works as follows.

ANSWERING THE DECRYPTION QUERIES. Let the query be $\mathbf{c}$. Now, there are two cases:

1. $(c_2, s_2) = (c_2^*, s_2^*)$: Then, the probability that $c' \neq c'^*$ and $H(c'^*; s_2) = H(c^*; s_2)$ is negligible over the choice of $H \in \mathsf{PHGen}(1^k)$. Thus, $(r, c_1, s_1) = (r^*, c_1^*, s_1^*)$, except with negligible probability. If this is the case, then $\mathbf{c} = \mathbf{c}^*$, and thus, the query of A is invalid. B returns $\perp$ in this case.

2. $(c_2, s_2) \neq (c_2^*, s_2^*)$: Then, B can use the H-inversion oracle to compute c' such that $H(c'; s_2) = c_2$. Let $c' = (r', c_1', s_1')$. Check that $(c_1', s_1') = (c_1, s_1)$ and $f(r') = c_0$. If both checks pass, return $H(r'; s_1) \oplus c_1$ to A. Otherwise, return $\perp$.

$\mathsf{Gen}(1^n)$: Run $\mathsf{TDPGen}(1^n)$ and get a pair (f, f^{-1}). Run $\mathsf{PHGen}(1^k)$ to get a perfectly one-way hash function H. Let $\mathrm{PK} = (f, H)$ and $\mathrm{SK} = f^{-1}$.

$\mathsf{Enc}(\mathrm{PK}, m)$:

1. Pick random $r \leftarrow \{0,1\}^n$. Compute $c_0 = f(r)$ and $c_1 = m \oplus H(r; s_1)$ for random s_1.
2. Let $c' = (r, s_1, c_1)$. Compute $c_2 = H(c'; s_2)$ for random s_2.

Output the ciphertext $\mathbf{c} = (c_0, c_1, c_2, s_1, s_2)$.

$\mathsf{Dec}(\mathrm{SK}, c)$: Parse $\mathbf{c}$ as $(c_0, c_1, c_2, s_1, s_2)$.

1. Compute $r' = f^{-1}(c_0)$, and $m' = c_1 \oplus H(r'; s_1)$.
2. Let $c' = (r', s_1, c_1)$. Output m' if $H(c'; s_2) = c_2$. Otherwise output $\perp$.

Figure 5: AN IND-CCA2-SECURE ENCRYPTION SCHEME.

GIVING THE CHALLENGE CIPHERTEXT TO A. Consider the following three experiments.

1. **Experiment $\mathrm{IND}_b^{(1)}$:** The challenge ciphertext is computed as $c^* = [f(r), u \oplus m_b, u', s_1, s_2]$, where r, s_1, s_2, u, u' are randomly chosen from the appropriate domains.

2. **Experiment $\mathrm{IND}_b^{(2)}$:** The challenge ciphertext is computed as $c^* = [f(r), H(r; s_1) \oplus m_b, u', s_1, s_2]$, where r, s_1, s_2, u' are randomly chosen from the appropriate domains.

3. **Experiment $\mathrm{IND}_b^{(3)}$:** The challenge ciphertext is computed as $c^* = [f(r), H(r; s_1), H(c'; s_2), s_1, s_2]$, where r, s_1, s_2 are randomly chosen from the appropriate domains.

The decryption queries in each of the experiments are answered as above. Note that $\mathrm{IND}_b^{(3)}$ is the same as IND_b. We will show that

1. $\mathrm{IND}_0^{(1)} \equiv \mathrm{IND}_1^{(1)}$,

2. $\mathrm{IND}_b^{(1)} \overset{c}{\approx} \mathrm{IND}_b^{(2)}$, and

3. $\mathrm{IND}_b^{(2)} \overset{c}{\approx} \mathrm{IND}_b^{(3)}$

The claim follows, since $\mathrm{IND}_0 \equiv \mathrm{IND}_0^{(3)} \overset{c}{\approx} \mathrm{IND}_0^{(1)} \equiv \mathrm{IND}_1^{(1)} \overset{c}{\approx} \mathrm{IND}_1^{(3)} \equiv \mathrm{IND}_1$.

$\mathrm{IND}_0^{(1)} \equiv \mathrm{IND}_1^{(1)}$, since in both cases, the ciphertext is a random 5-tuple, independent of m_b. $\mathrm{IND}_b^{(1)} \overset{c}{\approx} \mathrm{IND}_b^{(2)}$ since otherwise, we could distinguish between $(f(r), H(r; s), s)$ and $(f(r), s', s)$ where r, s, s' are chosen at random. This contradicts either (1) the adaptively perfect one-wayness of H with the uninvertible auxiliary information function f, or (2) the uninvertibility of f with access to the H-inversion oracle. A similar argument shows that $\mathrm{IND}_b^{(2)} \overset{c}{\approx} \mathrm{IND}_b^{(3)}$, if the composite function $g(r, s) = f(r) \circ h(r; s) \circ s$ is uninvertible, which follows from Claim 3.

Claim 3 *The function $g(r, s) = f(r) \circ h(r; s) \circ s$ is uninvertible, even with oracle access to an inverting algorithm for h on inputs of length $3n$.*

Proof: Suppose there is a PPT adversary A that inverts g. Then, we show a PPT adversary B that distinguishes between the tuples $(f(r), h(r; s), s)$ and $(f(r), s', s)$ for randomly chosen r, s, s'. B works as follows: B is given a tuple (a, b, c), and it simply invokes A on (a, b, c). B answers the POWHF inversion queries of A using its own POWHF inversion oracle. Since all the queries of A are to invert $h(\tilde{r})$ where $\tilde{r}$ is of length $3n$, B never has to invoke the inversion oracle on b.

Finally, A computes a purported inverse r' and returns r' to B. It checks if $f(r') = a$ and $h(r'; c) = b$. If both checks pass, return 1, else return a random bit. If the input tuple is of the form $(f(r), h(r; s), s)$, then A will succeed with non-negligible probability in inverting it, otherwise A cannot produce an inverse. Thus, B will distinguish between the tuples with non-negligible probability. □ □