Chosen-Ciphertext Security via Correlated Products

Alon Rosen[1]* and Gil Segev[2]

[1] Efi Arazi School of Computer Science,
Herzliya Interdisciplinary Center (IDC), Herzliya 46150, Israel.
alon.rosen@idc.ac.il
[2] Department of Computer Science and Applied Mathematics,
Weizmann Institute of Science, Rehovot 76100, Israel.
gil.segev@weizmann.ac.il

Abstract. We initiate the study of one-wayness under *correlated products*. We are interested in identifying necessary and sufficient conditions for a function f and a distribution on inputs $(x_1, \ldots, x_k)$, so that the function $(f(x_1), \ldots, f(x_k))$ is one-way. The main motivation of this study is the construction of public-key encryption schemes that are secure against chosen-ciphertext attacks (CCA). We show that any collection of injective trapdoor functions that is secure under a very natural correlated product can be used to construct a CCA-secure encryption scheme. The construction is simple, black-box, and admits a direct proof of security. We provide evidence that security under correlated products is achievable by demonstrating that lossy trapdoor functions (Peikert and Waters, STOC '08) yield injective trapdoor functions that are secure under the above mentioned correlated product. Although we currently base security under correlated products on existing constructions of lossy trapdoor functions, we argue that the former notion is potentially weaker as a general assumption. Specifically, there is no fully-black-box construction of lossy trapdoor functions from trapdoor functions that are secure under correlated products.

1 Introduction

The construction of secure public-key encryption schemes lies at the heart of cryptography. Following the seminal work of Goldwasser and Micali [20], increasingly strong security definitions have been formulated. The strongest notion to date is that of semantic security against a chosen-ciphertext attack (CCA) [27, 32], which protects against an adversary that is given access to decryptions of ciphertexts of her choice.

Constructions of CCA-secure public-key encryption schemes have followed several structural approaches. These approaches, however, either result in rather complicated schemes, or rely only on specific number-theoretic assumptions. Our

* Research supported in part by BSF grant 2006317.

goal in this paper is to construct a simple CCA-secure public-key encryption scheme based on general computational assumptions.

The first approach for constructing a CCA-secure encryption scheme was put forward by Naor and Yung [27], and relies on any semantically secure public-key encryption scheme and non-interactive zero-knowledge (NIZK) proof system for $\mathcal{NP}$. Their approach was later extended by Dolev, Dwork and Naor [11] for a more general notion of chosen-ciphertext attack, and subsequently simplified by Sahai [35] and by Lindell [26]. Schemes resulting from this approach, however, are somewhat complicated and impractical due to the use of generic NIZK proofs.

An additional approach was introduced by Cramer and Shoup [10], and is based on "smooth hash proof systems", which were shown to exist based on several number-theoretic assumptions. Elkind and Sahai [12] observed that both the above approaches can be viewed as special cases of a single paradigm in which ciphertexts include "proofs of well-formedness". Even though in some cases this paradigm leads to elegant and efficient constructions [9], the complexity of the underlying notions makes the general framework somewhat cumbersome.

Recently, Peikert and Waters [31] introduced the intriguing notion of lossy trapdoor functions, and demonstrated that such functions can be used to construct a CCA-secure public-key encryption scheme in a black-box manner. Their construction can be viewed as an efficient and elegant realization of the "proofs of well-formedness" paradigm. Lossy trapdoor functions seem to be a very powerful primitive. In particular, they were shown to also imply oblivious transfer protocols and collision-resistant hash functions[3]. It is thus conceivable that CCA-secure encryption can be realized based on weaker primitives.

A different approach was suggested by Canetti, Halevi and Katz [6] (followed by [3–5]) who constructed a CCA-secure public-key encryption scheme based on any identity-based encryption (IBE) scheme. Their construction is elegant, black-box, and essentially preserves the efficiency of the underlying IBE scheme. However, IBE is a rather strong cryptographic primitive, which is currently realized only based on a small number of specific number-theoretic assumptions.

1.1 Our Contributions

Motivated by the task of constructing a simple CCA-secure public-key encryption scheme, we initiate the study of one-wayness under *correlated products*. The main question in this context is to identify necessary and sufficient conditions for a collection of functions $\mathcal{F}$ and a distribution on inputs $(x_1, \dots, x_k)$ so that the function $(f_1(x_1), \dots, f_k(x_k))$ is one-way, where $f_1, \dots, f_k$ are independently chosen from $\mathcal{F}$. Our results are as follows:

1. We show that any collection of injective trapdoor functions that is secure under a very natural correlated product can be used to construct a CCA-secure public-key encryption scheme. The construction is simple, black-box,

[3] We note that the constructions of CCA-secure encryption and collision-resistant hash functions presented in [31] require lossy trapdoor functions that are "sufficiently lossy" (i.e., they rely on lossy trapdoor functions with sufficiently good parameters).

and admits a direct proof of security. Arguably, both the underlying assumption and the proof of security are simple enough to be taught in an undergraduate course in cryptography.

2. We demonstrate that any collection of lossy trapdoor functions (with appropriately chosen parameters) yields a collection of injective trapdoor functions that is secure under the correlated product that is required by our encryption scheme. In turn, existing constructions of lossy trapdoor functions [1, 31, 34] imply that our encryption scheme can be based on the hardness of the decisional Diffie-Hellman problem, and of Paillier's decisional composite residuosity problem.
3. We argue that security under correlated products is potentially weaker than lossy trapdoor functions as a general computational assumption. Specifically, we prove that there is no fully-black-box construction of lossy trapdoor functions from trapdoor functions (and even from enhanced trapdoor permutations) that are secure under correlated products.

Following our work Peikert [30] and Goldwasser and Vaikuntanathan [21] recently showed that security under correlated products is achievable also under the worst-case hardness of lattice problems (although these assumptions are currently not known to imply lossy trapdoor functions with the appropriately chosen parameters that are required for our transformation). Their constructions result in new CCA-secure public-key encryption schemes that are based on lattices, and this demonstrates that the correlated products approach for chosen-ciphertext security is fruitful, and that security under correlated products is achievable under a variety of number-theoretic assumptions.

In the remainder of this section we provide a high-level overview of our contributions, and then turn to describe the related work.

1.2 Security Under Correlated Products

It is well known that for every collection of one-way functions $\mathcal{F} = \{f_s\}_{s \in S}$ and polynomially-bounded $k \in \mathbb{N}$, the collection $\mathcal{F}_k = \{f_{s_1,\ldots,s_k}\}_{(s_1,\ldots,s_k) \in S^k}$, whose members are defined as

$$f_{s_1,\ldots,s_k}(x_1,\ldots,x_k) = (f_{s_1}(x_1),\ldots,f_{s_k}(x_k))$$

is also one-way. Moreover, such a direct product amplifies the one-wayness of $\mathcal{F}$ [19, 37], and this holds even when considering a single function (i.e., when $s_1 = \cdots = s_k$).

In general, however, the one-wayness of $\mathcal{F}_k$ is guaranteed only when the inputs are independently chosen, and when the inputs are correlated no such guarantee can exist. A well-known example for insecurity under correlated products is Håstad's attack [2, 23] on plain-broadcast RSA: there is an efficient algorithm that is given as input $x^3 \bmod N_1$, $x^3 \bmod N_2$, and $x^3 \bmod N_3$, and outputs x. More generally, it is rather easy to show that if collections of one-way functions exist, then there exists a collection of one-way functions $\mathcal{F} = \{f_s\}_{s \in S}$ such that

$f_{s_1,s_2}(x, x) = (f_{s_1}(x), f_{s_2}(x))$ is not one-way. However, this does not rule out the possibility of constructing a collection of one-way functions whose product remains one-way even when the inputs are correlated.

Informally, given a collection $\mathcal{F}$ of functions and a distribution $\mathcal{C}_k$ of inputs $(x_1, \ldots, x_k)$, we say that $\mathcal{F}$ is *secure under a $\mathcal{C}_k$-correlated product* if $\mathcal{F}_k$ is one-way when the inputs $(x_1, \ldots, x_k)$ are distributed according to $\mathcal{C}_k$ (a formal definition is provided in Section 2). The main goal in this setting is to characterize the class of collections $\mathcal{F}$ and distributions $\mathcal{C}_k$ that satisfy this notion.

We motivate the study of security under correlated products by relating it to the study of chosen-ciphertext security. Specifically, we show that any collection of injective trapdoor functions that is secure under a very natural correlated product can be used to construct a CCA-secure public-key encryption scheme. The simplest form of distribution $\mathcal{C}_k$ on inputs that is sufficient for our construction is the *uniform k-repetition distribution* that outputs k copies of a uniformly chosen input x. We note that although this seems to be a strong requirement, we demonstrate that it can be based on various number-theoretic assumptions.

More generally, our construction can rely on any distribution $\mathcal{C}_k$ with the property that any $(x_1, \ldots, x_k)$ in the support of $\mathcal{C}_k$ can be reconstructed given any $t = (1 - \epsilon)k$ entries from $(x_1, \ldots, x_k)$, for some constant $0 < \epsilon < 1$. For example, $\mathcal{C}_k$ may be a distribution that evaluates a random polynomial of degree at most $t-1$ on a set of k points (in this case the x_i's are t-wise independent, but other choices which do not guarantee such a strong property are also possible).

1.3 Chosen-Ciphertext Security via Correlated Products

Consider the following, very simple, public-key encryption scheme. The public-key consists of an injective trapdoor function f, and the secret-key consists of its trapdoor td. Given a message $m \in \{0, 1\}$, the encryption algorithm chooses a random input x and outputs the ciphertext $(f(x), m \oplus h(x))$, where h is a hard-core predicate of f. The decryption algorithm uses the trapdoor to retrieve x and then extracts m. In what follows we frame our approach as a generalization of this fundamental scheme.

The above scheme is easily proven secure against a chosen-plaintext attack. Any adversary $\mathcal{A}$ that distinguishes between an encryption of 0 and an encryption of 1 can be used to construct an adversary $\mathcal{A}'$ that distinguishes between $h(x)$ and a randomly chosen bit with exactly the same probability. Specifically, $\mathcal{A}'$ receives a function f, a value $y = f(x)$, and a bit w (which is either $h(x)$ or a uniformly chosen bit), and emulates $\mathcal{A}$ with f as the public-key and $(y, m \oplus w)$ as the challenge ciphertext for a random message m. This scheme, however, fails to be proven secure against a chosen-ciphertext attack (even when considering only CCA1 security). There is a conflict between the fact that $\mathcal{A}'$ is required to answer decryption queries, and the fact that $\mathcal{A}'$ does not have the trapdoor for inverting f.

The following simplified variant of our scheme is designed to resolve this conflict. The public-key consists of k pairs of functions $(f_1^0, f_1^1), \ldots, (f_k^0, f_k^1)$, where each function is sampled independently from a collection $\mathcal{F}$ of injective trapdoor functions[4]. The secret-key consists of the trapdoors $(td_1^0, td_1^1), \ldots, (td_k^0, td_k^1)$, where each td_i^b is the trapdoor of the function f_i^b. Given a message $m \in \{0,1\}$, the encryption algorithm chooses a random $v = v_1 \cdots v_k \in \{0,1\}^k$, a random input x, and outputs the ciphertext

$$E_{PK}(m; v, x) = (v, f_1^{v_1}(x), \ldots, f_k^{v_k}(x), m \oplus h(x)) \ ,$$

where h is a hard-core predicate of $\mathcal{F}_k$ with respect to the uniform k-repetition distribution. The decryption algorithm acts as follows: given a ciphertext of the form $(v, y_1, \ldots, y_k, z)$ it inverts $y_1, \ldots, y_k$ to obtain $x_1, \ldots, x_k$, and if $x_1 = \cdots = x_k$ then it outputs $h(x_1) \oplus z$ (otherwise it outputs $\perp$).

In order to prove the CCA1 security of this scheme, we show that any adversary $\mathcal{A}$ that breaks the CCA1 security of the scheme can be used to construct an adversary $\mathcal{A}'$ that distinguishes between $h(x)$ and a randomly chosen bit with exactly the same probability. The adversary $\mathcal{A}'$ receives as input k functions $f_1, \ldots, f_k \in \mathcal{F}$, k values $y_1 = f_1(x), \ldots, y_k = f_k(x)$, and a bit w (which is either $h(x)$ or a uniformly chosen bit). $\mathcal{A}'$ simulates the CCA1 interaction to $\mathcal{A}$ by choosing a random value $v^* = v_1^* \cdots v_k^* \in \{0,1\}^k$, and for each pair (f_i^0, f_i^1) it sets $f_i^{vk_i^*} = f_i$ and samples $f_i^{1-vk_i^*}$ together with its trapdoor from $\mathcal{F}$. Note that now $\mathcal{A}'$ is able to answer decryption queries as long as none of them contain the value v^*, and in this case we claim that essentially no information on v^* is revealed. The challenge ciphertext is then computed as $(v^*, y_1, \ldots, y_k, m \oplus w)$ for a random message m. If $\mathcal{A}$ guesses the bit m correctly then $\mathcal{A}'$ outputs that $w = h(x)$, and otherwise $\mathcal{A}'$ outputs that w is a random bit.

Our scheme can be viewed as a realization of the Naor-Yung paradigm [27] in which a message is encrypted using several independently chosen keys, and ciphertexts include "proofs of well-formedness". In our scheme, however, the decryption algorithm can verify "well-formedness" of ciphertexts without any additional "proof": given any one of the trapdoors it is possible to verify that the remaining values are consistent with the same input x.

Our scheme is inspired also by the one based on lossy trapdoor functions [31], and specifically, by the generic construction of *all-but-one* lossy trapdoor functions from lossy trapdoor functions. However, the proof security of our construction is simpler than that of [31] due to the additional hybrids resulting from using both lossy trapdoor functions and all-but-one trapdoor functions. In addition, our construction only relies on *computational* hardness, whereas the construction of [31] relies on the *statistical* properties of lossy trapdoor functions.

Finally, we note that our proof of security is rather similar to that of the IBE-based schemes [4–6]. The value v^* can be viewed as the challenge identity, for which $\mathcal{A}'$ does not have the secret key, and is therefore not able to decrypt ciphertexts for this identity. For any other identity $v \neq v^*$, $\mathcal{A}'$ has sufficient information to decrypt ciphertexts.

---

[4] For CCA1 security any $k = \omega(\log n)$ is sufficient, where n is the security parameter. For our more generalized construction that guarantees CCA2 security, any $k = n^\epsilon$ for some constant $0 < \epsilon < 1$ is sufficient.

In some sense, our approach enjoys "the best of both worlds" in that both the underlying assumption and the proof of security are simpler than those of previous approaches.

1.4 A Black-Box Separation

Although we currently base security under correlated products on lossy trapdoor functions, we argue that security under correlated products is potentially weaker than lossy trapdoor functions as a general computational assumption. Specifically, we prove that there is no fully-black-box construction of lossy trapdoor functions from trapdoor functions that are secure under correlated products. We present an oracle relative to which there exists a collection of injective trapdoor functions (and even of enhanced trapdoor permutations) that is secure under a correlated product with respect to the above mentioned uniform k-repetition distribution, but there is no collection of lossy trapdoor functions. The oracle is essentially the collision-finding oracle due to Simon [36], and the proof follows the approach of Haitner et al. [22] while overcoming several technical difficulties.

Informally, consider a circuit A which is given as input $(f_1(x), \ldots, f_k(x))$, and whose goal is to retrieve x. The circuit A is provided access to an oracle Sam that receives as input a circuit C and outputs random w and w' such that $C(w) = C(w')$. As in the approach of Haitner et al. the idea underlying the proof is to distinguish between two cases: one in which A obtains information on x via one of its Sam-queries, and the other in which none of A's Sam-queries provides information on x. The proof consists of two modular parts dealing with these two cases separately. In first part we generalize an argument of Haitner et al. (who in turn generalized the reconstruction lemma of Gennaro and Trevisan [14]) to deal with the product of several functions. We show that the probability that $\mathcal{A}$ retrieves x in the first case is exponentially small. In the second part we show that the second case can essentially be reduced to the first case. This part of the proof is simpler than the corresponding argument of Haitner et al. that considers a more interactive setting.

1.5 Related Work

Much research has been devoted for the construction of CCA-secure public-key encryption schemes. A significant part of this research was already mentioned in the previous sections, and here we mainly focus on results regarding the possibility and limitations of basing such schemes on general assumptions.

Pass, shelat and Vaikuntanathan [28] constructed a public-key encryption scheme that is non-malleable against a chosen-plaintext attack from any semantically secure one (building on the scheme of Dolev, Dwork and Naor [11]). Their technique was later shown by Cramer et al. [8] to also imply non-malleability against a weak notion of chosen-ciphertext attack, in which the number of decryption queries is bounded. These approaches, however, are rather impractical due to the use of generic (designated verifier) NIZK proofs. Very recently, Choi et al. [7] showed that the latter notions of security can in fact be elegantly realized in a black-box manner based on the same assumptions. The reader is referred to [11, 29] for classifications of the different notions of security.

Impagliazzo and Rudich [24] introduced a paradigm for proving impossibility results for cryptographic constructions. They showed that there are no black-box constructions of key-agreement protocols from one-way permutations, and substantial additional work in this line followed (see, for example [13, 15, 17, 25, 36] and many more). The reader is referred to [33] for a comprehensive discussion and taxonomy of black-box constructions. In the context of public-key encryption schemes, most relevant to our result is the work of Gertner, Malkin and Myers [16], who addressed the question of whether or not semantically secure public-key encryption schemes imply the existence of CCA-secure schemes. They showed that there are no black-box constructions in which the decryption algorithm of the proposed CCA-secure scheme does not query the encryption algorithm of the semantically secure one.

1.6 Paper Organization

The remainder of the paper is organized as follows. In Section 2 we provide a formal treatment of security under correlated products, which is shown to be satisfied by lossy trapdoor functions. In Section 3 we describe a simplified version of our encryption scheme which already illustrates the main ideas underlying our approach. In Section 4 we prove that there is no fully-black-box construction of lossy trapdoor functions from trapdoor functions secure under correlated products. Due to space limitation we refer the reader to the full version for a more generalized version of the encryption scheme, and for a complete proof of the black-box separation.

2 Security Under Correlated Products

In this section we formally define the notion of security under correlated products, and demonstrate that the notion is satisfied by any collection of lossy trapdoor functions (with appropriately chosen parameters) for a very natural and useful correlation. We then discuss the exact parameters that are required for our encryption scheme, and the number-theoretic assumptions that are currently known to guarantee such parameters.

A collection of functions is represented as a pair of algorithms $\mathcal{F} = (G, F)$, where G is a generation algorithm used for sampling a description of a function, and F is an evaluation algorithm used for evaluating a function on a given input. The following definition formalizes the notion of a k-wise product which introduces a collection $\mathcal{F}_k$ consisting of all k-tuples of functions from $\mathcal{F}$.

Definition 2.1 (k-wise product). *Let $\mathcal{F} = (G, F)$ be a collection of efficiently computable functions. For any integer k, we define the* k-wise product *$\mathcal{F}_k = (G_k, F_k)$ as follows:*

– *The generation algorithm G_k on input 1^n invokes $G(1^n)$ for k times independently and outputs $(s_1, \ldots, s_k)$. That is, a function is sampled from $\mathcal{F}_k$ by independently sampling k functions from $\mathcal{F}$.*
– *The evaluation algorithm F_k on input $(s_1, \ldots, s_k, x_1, \ldots, x_k)$ invokes F to evaluate each function s_i on x_i. That is,*

$$F_k(s_1, \ldots, s_k, x_1, \ldots, x_k) = (F(s_1, x_1), \ldots, F(s_k, x_k)) \ .$$

The notion of a one-way function asks for a function that is efficiently computable but is hard to invert given the image of a uniformly chosen input. More generally, one can naturally extend this notion to consider one-wayness under any specified input distribution, not necessarily the uniform distribution. That is, informally, we say that a function is one-way with respect to an input distribution $\mathcal{I}$ if it is efficiently computable but hard to invert given the image of a random input sampled according to $\mathcal{I}$.

In the context of k-wise products, a standard argument shows that for any collection $\mathcal{F}$ which is one-way with respect to some input distribution $\mathcal{I}$, the k-wise product $\mathcal{F}_k$ is one-way with respect to the input distribution which samples k independent inputs from $\mathcal{I}$. The following definition formalizes the notion of security under correlated products, where the inputs for $\mathcal{F}_k$ may be correlated.

Definition 2.2 (Security under correlated products). *Let $\mathcal{F} = (G, F)$ be a collection of efficiently computable functions, and let $\mathcal{C}_k$ be a distribution where $\mathcal{C}_k(1^n)$ is distributed over $\{0,1\}^{k \cdot n}$ for some integer $k = k(n)$. We say that $\mathcal{F}$ is* secure under a $\mathcal{C}_k$-correlated product *if $\mathcal{F}_k$ is one-way with respect to the input distribution $\mathcal{C}_k$.*

Correlated products security based on lossy trapdoor functions. We conclude this section by demonstrating that, for an appropriate choice of parameters, any collection of lossy trapdoor functions yields a collection of injective trapdoor functions that is secure under a $\mathcal{C}_k$-correlated product. The input distribution under consideration, $\mathcal{C}_k$, samples a uniformly random input x and outputs k copies of x. We refer to this distribution as the *uniform k-repetition distribution*, and this distribution is the one required for the simplified variant of our encryption scheme, presented in Section 3.

Specifically, given a collection of lossy trapdoor functions $\mathcal{F} = (G, F, F^{-1})$ we define a collection $\mathcal{F}_{\mathsf{inj}}$ of injective trapdoor functions by restricting $\mathcal{F}$ to its injective functions. That is, $\mathcal{F}_{\mathsf{inj}} = (G_{\mathsf{inj}}, F, F^{-1})$ where $G_{\mathsf{inj}}(1^n) = G(1^n, \mathsf{injective})$. We prove the following theorem:

Theorem 2.1. *Let $\mathcal{F} = (G, F, F^{-1})$ be a collection of (n, ℓ)-lossy trapdoor functions. Then, for any integer $k < \frac{n - \omega(\log n)}{n - \ell}$, for any probabilistic polynomial-time algorithm $\mathcal{A}$ and polynomial $p(\cdot)$, it holds that*

$$\Pr\left[\mathcal{A}(1^n, s_1, \ldots, s_k, F(s_1, x), \ldots, F(s_k, x)) = x\right] < \frac{1}{p(n)} \ ,$$

for all sufficiently large n, where the probability is taken over the choices of $s_1 \leftarrow G_{\mathsf{inj}}(1^n), \ldots, s_k \leftarrow G_{\mathsf{inj}}(1^n)$, $x \leftarrow \{0,1\}^n$, and over the internal coin tosses of $\mathcal{A}$.

Proof. Peikert and Waters [31] proved that any collection of $(n, \omega(\log n))$-lossy trapdoor functions is in particular a collection of one-way functions. Thus, it is sufficient to prove that $\mathcal{F}_k$ is a collection of $(n, \omega(\log n))$-lossy trapdoor functions. For any k functions $s_1, \ldots, s_k$ sampled according to $G_{\mathsf{inj}}(1^n)$, the function $F_k(s_1, \ldots, s_k, x_1, \ldots, x_k) = (F(s_1, x_1), \ldots, F(s_k, x_k))$ is clearly injective. For any k functions $s_1, \ldots, s_k$ sampled according to $G_{\mathsf{lossy}}(1^n)$, the function $F_k(s_1, \ldots, s_k, x_1, \ldots, x_k) = (F(s_1, x_1), \ldots, F(s_k, x_k))$ obtains at most $2^{k(n-\ell)}$ values, which is upper bounded by $2^{n-\omega(\log n)}$ for any $k < \frac{n-\omega(\log n)}{n-\ell}$. Finally, note that a standard hybrid argument shows that the distribution obtained by independently sampling k functions according to $G_{\mathsf{inj}}(1^n)$ is computationally indistinguishable from the distribution obtained by independently sampling k functions according to $G_{\mathsf{lossy}}(1^n)$. Thus, $\mathcal{F}_k$ is a collection of $(n, \omega(\log n))$-lossy trapdoor functions. □

The required parameters for our scheme. The assumption underlying our encryption scheme asks for $k(n) = \omega(\log n)$ for CCA1 security, and for $k(n) = n^{\epsilon}$ (for some constant $0 < \epsilon < 1$) for CCA2 security. In turn, existing constructions of lossy trapdoor functions guaranteing these parameters [1, 31, 34] imply that our encryption scheme can be realized under the hardness of the decisional Diffie-Hellman problem, and of Paillier's decisional composite residuosity problem. We note that the lattice-based construction of Peikert and Waters [31] guarantees only a constant $k(n)$ that is not sufficient for our encryption scheme. However, Peikert [30] and Goldwasser and Vaikuntanathan [21] recently showed that security under correlated products (with sufficiently large $k(n)$) is nevertheless achievable under the worst-case hardness of lattice problems, although these are currently known to imply lossy trapdoor functions with only a relatively small amount of loss.

3 A Simplified Construction

In this section we describe a simplified version of our construction which already illustrates the main ideas underlying our approach. The encryption scheme presented in the current section is a simplification in the sense that it relies on a seemingly stronger computational assumption than the more generalized construction which is presented in the full version. In addition, we first present the scheme as encrypting only one bit messages, and then demonstrate that it naturally extends to multi-bit messages. In what follows we state the computational assumption, describe the encryption scheme, prove its security, and describe the extension to multi-bit messages.

The underlying computational assumption. The computational assumption underlying the simplified scheme is that there exists a collection $\mathcal{F}$ of injective trapdoor functions and an integer function $k = k(n)$ such that $\mathcal{F}$ is secure under a $\mathcal{C}_k$-correlated product, where $\mathcal{C}_k$ is the uniform k-repetition distribution (i.e., outputs k copies of a uniformly distributed input x). Specifically, our scheme uses a hard-core predicate $h : \{0,1\}^* \rightarrow \{0,1\}$ for $\mathcal{F}_k$ with respect to $\mathcal{C}_k$. That is, the underlying computational assumption is that for any probabilistic polynomial-time predictor $\mathcal{P}$ it holds that

$$\left| \Pr\left[\mathcal{P}(1^n, s_1, \ldots, s_k, F(s_1, x), \ldots, F(s_k, x)) = h(s_1, \ldots, s_k, x)\right] - \frac{1}{2} \right|$$

is negligible in n, where the probability is taken over the choices of $s_1 \leftarrow G(1^n), \ldots, s_k \leftarrow G(1^n)$, $x \leftarrow \{0,1\}^n$, and over the internal coin tosses of $\mathcal{P}$.

The integer function $k(n)$ should correspond to the bit-length of verification keys of some one-time strongly-unforgeable signature scheme $(\mathsf{KG_{sig}}, \mathsf{Sign}, \mathsf{Ver})$. By applying a universal one-way hash function to the verification keys (as in [11]) it suffices that the above assumption holds for $k(n) = n^\epsilon$ for a constant $0 < \epsilon < 1$. For simplicity, however, when describing our scheme we do not apply a universal one-way hash function to the verification keys. We also note that for an even more simplified version which is only CCA1-secure (the one described in Section 1.3), any $k(n) = \omega(\log n)$ suffices.

The construction. The following describes our simplified encryption scheme given by the triplet (KG, E, D).

- **Key generation:** On input 1^n the key generation algorithm invokes $G(1^n)$ for $2k$ times independently to obtain $2k$ descriptions of functions denoted $(s_1^0, s_1^1), \ldots, (s_k^0, s_k^1)$ with trapdoors $(td_1^0, td_1^1), \ldots, (td_k^0, td_k^1)$. The public-key and secret-key are defined as follows:

$$\begin{aligned} PK &= \left(\left(s_1^0, s_1^1\right), \ldots, \left(s_k^0, s_k^1\right)\right) \\ SK &= \left(\left(td_1^0, td_1^1\right), \ldots, \left(td_k^0, td_k^1\right)\right) \ . \end{aligned}$$

- **Encryption:** On input a message $m \in \{0,1\}$ and a public key PK, the algorithm samples $(vk, sk) \leftarrow \mathsf{KG_{sig}}(1^n)$ where $vk = vk_1 \circ \cdots \circ vk_k \in \{0,1\}^k$, chooses a uniformly distributed $x \in \{0,1\}^n$, and outputs the ciphertext

$$(vk, y_1, \ldots, y_k, c_1, c_2) \ ,$$

  where

$$\begin{aligned} y_i &= F\left(s_i^{vk_i}, x\right) \quad \forall i \in [k] \\ c_1 &= m \oplus h\left(s_1^{vk_1}, \ldots, s_k^{vk_k}, x\right) \\ c_2 &= \mathsf{Sign}(sk, (y_1, \ldots, y_k, c_1)) \ . \end{aligned}$$

- **Decryption:** On input a ciphertext $(vk, y_1, \ldots, y_k, c_1, c_2)$ and a secret-key SK, the algorithm acts as follows. If $\mathsf{Ver}(vk, (y_1, \ldots, y_k, c_1), c_2) = 0$, it outputs $\perp$. Otherwise, for every $i \in [k]$ it computes $x_i = F^{-1}\left(td_i^{vk_i}, y_i\right)$. If $x_1 = \cdots = x_k$ then it outputs $c_1 \oplus h\left(s_1^{vk_1}, \ldots, s_k^{vk_k}, x_1\right)$, and otherwise it outputs $\perp$.

The following theorem establishes the security of the scheme.

Theorem 3.1. *Assuming that $\mathcal{F}$ is secure under a $\mathcal{C}_k$-correlated product, where $\mathcal{C}_k$ is the uniform k-repetition distribution, and that* $(\mathsf{KG_{sig}}, \mathsf{Sign}, \mathsf{Ver})$ *is one-time strongly unforgeable, the encryption scheme (KG, E, D) is CCA2-secure.*

Proof. Let $\mathcal{A}$ be a probabilistic polynomial-time CCA2-adversary. We denote by Forge the event in which for one of $\mathcal{A}$'s decryption queries $(vk, y_1, \ldots, y_k, c_1, c_2)$ during the CCA2 interaction it holds that $vk = vk^*$ (where vk^* is given in the secret key) and $\mathsf{Ver}(vk, (y_1, \ldots, y_k, c_1), c_2) = 1$. We first argue that the event Forge has a negligible probability due to the security of the one-time signature scheme. Then, assuming that the event Forge does not occur, we construct a probabilistic polynomial-time algorithm $\mathcal{P}$ that predicts the hard-core predicate h while preserving the advantage of $\mathcal{A}$.

More formally, we denote by $\mathsf{Success}$ the event in which $\mathcal{A}$ successfully guesses the bit b used for encrypting the challenge ciphertext. Then, the advantage of $\mathcal{A}$ in the CCA2 interaction is bounded as follows:

$$\begin{aligned}\left|\Pr\left[\mathsf{Success}\right] - \frac{1}{2}\right| &= \left|\Pr\left[\mathsf{Success} \wedge \mathsf{Forge}\right] + \Pr\left[\mathsf{Success} \wedge \overline{\mathsf{Forge}}\right] - \frac{1}{2}\right| \\ &\leq \Pr\left[\mathsf{Forge}\right] + \left|\Pr\left[\mathsf{Success} \wedge \overline{\mathsf{Forge}}\right] - \frac{1}{2}\right| .\end{aligned}$$

The theorem follows from the following two claims:

Claim 3.2. $\Pr\left[\mathsf{Forge}\right]$ *is negligible.*

Proof. We show that any probabilistic polynomial-time adversary $\mathcal{A}$ for which $\Pr\left[\mathsf{Forge}\right]$ is non-negligible, can be used to construct a probabilistic polynomial-time adversary $\mathcal{A}'$ that breaks the security of the one-time signature with the same probability. The adversary $\mathcal{A}'$ is given a verification key vk^* sampled using $\mathsf{KG_{sig}}(1^n)$ and simulates the CCA2 interaction to $\mathcal{A}$ as follows. $\mathcal{A}'$ begins by invoking the key generation algorithm on input 1^n and using vk^* for forming the public and secret keys. In the decryption phases, whenever $\mathcal{A}$ submits a decryption query $(vk, y_1, \ldots, y_k, c_1, c_2)$, $\mathcal{A}'$ acts as follows. If $vk = vk^*$ and $\mathsf{Ver}(vk, (y_1, \ldots, y_k, c_1), c_2) = 1$, then $\mathcal{A}'$ outputs $((y_1, \ldots, y_k, c_1), c_2)$ as the forgery and halts. Otherwise, $\mathcal{A}'$ invokes the decryption procedure. In the challenge phase, upon receiving two message m_0 and m_1, $\mathcal{A}'$ chooses $b \in \{0, 1\}$ and $x \in \{0, 1\}^n$ uniformly at random, and computes

$$\begin{aligned} y_i &= F\left(s_i^{vk_i^*}, x\right) \quad \forall i \in [k] \\ c_1 &= m_b \oplus h\left(s_1^{vk_1^*}, \ldots, s_k^{vk_k^*}, x\right) .\end{aligned}$$

Then, it obtains a signature c_2 on $(y_1, \ldots, y_k, c_1)$ with respect to vk^* (recall that $\mathcal{A}'$ is allowed to ask for a signature on one message). Finally, it sends $(vk^*, y_1, \ldots, y_k, c_1, c_2)$ to $\mathcal{A}$. We note that during the second decryption phase, if $\mathcal{A}$ submits the challenge ciphertext as a decryption query, then $\mathcal{A}'$ responds with $\perp$.

Note that prior to the first decryption query in which Forge occurs (assuming that Forge indeed occurs), the simulation of the CCA2 interaction is perfect. Therefore, the probability that $\mathcal{A}'$ breaks the security of the one-time signature scheme is exactly $\Pr[\mathsf{forge}]$. The security of the signature scheme implies that this probability is negligible. ☐

Claim 3.3. $\left|\Pr\left[\mathsf{Success} \wedge \overline{\mathsf{Forge}}\right] - \frac{1}{2}\right|$ *is negligible.*

Proof. Given any efficient adversary $\mathcal{A}$ for which $\left|\Pr\left[\mathsf{Success} \wedge \overline{\mathsf{Forge}}\right] - \frac{1}{2}\right|$ is non-negligible, we construct a predictor $\mathcal{P}$ that breaks the security of the hard-core predicate h. That is,

$$\left|\Pr\left[\mathcal{P}(1^n, s_1, \ldots, s_k, F(s_1, x), \ldots, F(s_k, x)) = h(s_1, \ldots, s_k, x)\right] - \frac{1}{2}\right|$$

is non-negligible, where $s_1 \leftarrow G(1^n), \ldots, s_k \leftarrow G(1^n)$ independently, and the probability is taken over the uniform choice of $x \in \{0,1\}^n$, and over the internal coin tosses of both G and $\mathcal{P}$.

For simplicity, we first construct an efficient distinguisher $\mathcal{A}'$ which receives input of the form $(1^n, s_1, \ldots, s_k, F(s_1, x), \ldots, F(s_k, x))$ and a bit $w \in \{0,1\}$ which is either $h(s_1, \ldots, s_k, x)$ or a uniformly random bit, and is able to distinguish between the two cases with non-negligible probability. The distinguisher $\mathcal{A}'$ acts by simulating the CCA2 interaction to $\mathcal{A}$. More specifically, on input $(1^n, s_1, \ldots, s_k, y_1, \ldots, y_k)$ and a bit w, the distinguisher $\mathcal{A}'$ first creates a pair (PK, SK) as follows. It samples $(vk^*, sk^*) \leftarrow \mathsf{KG}_{\mathsf{sig}}(1^n)$, where $vk^* = vk_1^* \circ \cdots \circ vk_k^* \in \{0,1\}^k$, and for every $i \in [k]$ sets $s_i^{vk_i^*} = s_i$ and samples $\left(s_i^{1-vk_i^*}, td_i^{1-vk_i^*}\right) \leftarrow G(1^n)$. Then, $\mathcal{A}'$ outputs the public-key

$$PK = \left(\left(s_1^0, s_1^1\right), \ldots, \left(s_k^0, s_k^1\right)\right) \ .$$

Whenever $\mathcal{A}$ submits a decryption query of the form $(vk, y_1, \ldots, y_k, c_1, c_2)$, $\mathcal{A}'$ acts as follows. If $vk = vk^*$ or $\mathsf{Ver}(vk, (y_1, \ldots, y_k, c_1), c_2) = 0$, it outputs $\perp$ and halts. Otherwise, it picks some $i \in [k]$ for which $vk_i \neq vk_i^*$ and computes $x = F^{-1}\left(td_i^{vk_i}, y_i\right)$. If for every $j \in [k]$ it holds that $y_j = F\left(s_j^{vk_j}, x\right)$, it outputs $c_1 \oplus h\left(s_1^{vk_1}, \ldots, s_k^{vk_k}, x\right)$, and otherwise it outputs $\perp$.

In the challenge phase, given two messages m_0 and m_1, $\mathcal{A}'$ chooses a random bit $b \in \{0,1\}$ and replies with the challenge ciphertext

$$c = (vk^*, y_1, \ldots, y_k, c_1, c_2) \ ,$$

where $c_1 = m_b \oplus w$, and $c_2 = \mathsf{Sign}(sk^*, (y_1, \ldots, y_k, c_1))$. We note that during the second decryption phase, if $\mathcal{A}$ submits the challenge ciphertext as a decryption query, then $\mathcal{A}'$ responds with $\perp$. At the end of this interaction $\mathcal{A}$ outputs a bit b'. If $b' = b$ then $\mathcal{A}'$ outputs 1, and otherwise $\mathcal{A}'$ outputs 0.

In order to compute the advantage of $\mathcal{A}'$ we observe the following:

1. If w is a uniformly random bit, then the challenge ciphertext in the simulated interaction is independent of b. Therefore, the probability that $\mathcal{A}'$ outputs 1 in this case is exactly 1/2.
2. If $w = h(s_1, \ldots, s_k, x)$, then as long as the event Forge does not occur, the simulated interaction is identical to the CCA2 interaction (a formal argument follows). Therefore, the probability that $\mathcal{A}'$ outputs 1 in this case is exactly $\Pr\left[\mathsf{Success} \wedge \overline{\mathsf{Forge}}\right]$.

   Note that the only difference between the CCA2 interaction and the simulated interaction is the distribution of the challenge ciphertext: In the CCA2 interaction the value vk in the challenge ciphertext is a randomly chosen verification key, and in the simulated interaction the value vk is chosen ahead of time by $\mathcal{A}$. In what follows we claim that as long as the event Forge does not occur, the distribution of vk in the challenge ciphertext is identical in the two cases.

   Formally, denote by $vk_1, \ldots, vk_q$ the random variables corresponding to the value of vk in $\mathcal{A}$'s decryption queries (without loss of generality we assume that $\mathcal{A}$ always submits q queries, and that the signature verification never fails on these queries). In the CCA2 interaction, as long as the event Forge does not occur, it holds that the verification key used for the challenge ciphertext is a random verification key with the only restriction that it is different than $vk_1, \ldots, vk_q$. In the simulated interaction, given that $vk^* \notin \{vk_1, \ldots, vk_q\}$, we claim that from $\mathcal{A}$'s point of view, the value vk^* is also a random verification key which is different than $vk_1, \ldots, vk_q$. That is, each $vk^* \notin \{vk_1, \ldots vk_q\}$ produces exactly the same transcript. Indeed, first note that the public key is independent of vk^*. Now consider a decryption query $(vk, y_1, \ldots, y_k, c_1, c_2)$ for some $vk \in \{vk_1, \ldots, vk_q\}$. For any $vk^* \neq vk$, if $y_1, \ldots, y_k$ have the same preimage x, then the decryption algorithm will always output $c_1 \oplus h\left(s_1^{vk_1}, \ldots, s_k^{vk_k}, x\right)$. In addition, for any $vk^* \neq vk$, if $y_1, \ldots, y_k$ do not have the same preimage, then the decryption algorithm will always output $\perp$.

The above observations imply that

$$\begin{aligned} &\left|\Pr\left[\mathcal{A}' \text{ outputs } 1 \mid w = h(s_1, \ldots, s_k, x)\right] - \Pr\left[\mathcal{A}' \text{ outputs } 1 \mid w \text{ is random}\right]\right| \\ &\quad = \left|\Pr\left[\mathsf{Success} \wedge \overline{\mathsf{Forge}}\right] - \frac{1}{2}\right| . \end{aligned}$$

A standard argument (see, for example, [18, Chapter 3.4]) can be applied to efficiently transform $\mathcal{A}'$ into a predictor $\mathcal{P}$ that predicts $h(s_1, \ldots, s_k, x)$ with the same probability. □

□

Encrypting any polynomial number of bits. For simplicity we presented the encryption scheme above for one-bit plaintexts. We now demonstrate that our approach extends to plaintexts of any polynomial length while relying on the same computational assumption[5].

Recall that the underlying computational assumption is the existence of a collection $\mathcal{F}$ of injective trapdoor functions such that $\mathcal{F}_k$ is one-way under the uniform k-repetition distribution (i.e., $x_1 = \cdots = x_k$ where x_1 is chosen uniformly at random). Specifically, the scheme uses a hard-core predicate $h : \{0,1\}^* \to \{0,1\}$ for $\mathcal{F}_k$ to mask the plaintext bit. This assumption clearly implies that for any polynomial $T = T(n)$ there exists a collection $\mathcal{F}'$ of injective trapdoor functions such that $\mathcal{F}'$ is one-way under the uniform k-repetition distribution, and has a hard-core *function* $h' : \{0,1\}^* \to \{0,1\}^T$ that can be used in our scheme to mask T-bit plaintexts. Specifically, the collection $\mathcal{F}'$ is defined as follows: for every function $f : \{0,1\}^n \to \{0,1\}^m$ in $\mathcal{F}$ define a function $f' : \{0,1\}^{Tn} \to \{0,1\}^{Tm}$ by $f'(x_1, \dots x_T) = (f(x_1), \dots, f(x_T))$. The security proof of the T-bit encryption scheme is essentially identical to the proof of Theorem 3.1 by showing that any successful CCA-adversary can be used to either break the one-time signature scheme or to break the pseudorandomness of h'.

4 A Black-Box Separation

In this section we show that there is no fully-black-box construction of lossy trapdoor functions (with even a single bit of lossiness) from injective trapdoor functions that are secure under correlated products. We show that this holds for the seemingly strongest form of correlated product, where independently chosen functions are evaluated on the same input (i.e., we consider the uniform k-repetition distribution).

Our proof consists of constructing an oracle $\mathcal{O}$ relative to which there exists a collection of injective trapdoor functions that are permutations secure under a correlated product[6], but there are no collections of lossy trapdoor functions. In what follows, we describe the oracle $\mathcal{O}$, and show that it breaks the security of any collection of lossy trapdoor functions.

The oracle. The oracle $\mathcal{O}$ is of the form $(\tau, \mathsf{Sam}^\tau)$, where τ is a collection of trapdoor permutations, and Sam^τ is an oracle that samples random collision. Specifically, Sam receives as input a description of a circuit C (which may contain τ-gates), chooses a random input w, and then samples a uniformly distributed $w' \in C^{-1}(C(w))$.

We now explain how exactly Sam samples w and w'. We provide Sam with a collection of permutations $\mathcal{F}$, where for every possible circuit C the collection

[5] It is well-known that for semantic security under a chosen-plaintext attack it is straightforward to construct a multi-bit encryption scheme from any one-bit encryption scheme by independently encrypting the individual bits of the plaintext. For semantic security under a chosen-ciphertext attack, however, this approach fails in general.

[6] These functions are in fact enhanced trapdoor permutations, but we note that this is not essential for our result.

$\mathcal{F}$ contains two permutations f_C^1 and f_C^2 over the domain of C. Given a circuit $C : \{0,1\}^m \to \{0,1\}^{\ell(m)}$, for some m and $\ell(m)$, the oracle Sam uses f_C^1 to compute $w = f_C^1(0^m)$. Then, it computes $w' = f_C^2(t)$ for the lexicographically smallest $t \in \{0,1\}^m$ such that $C(f_C^2(t)) = C(w)$. Note that whenever the permutations f_C^1 and f_C^2 are chosen uniformly at random, and independently of all other permutations in $\mathcal{F}$, then w is uniformly distributed over $\{0,1\}^m$, and w' is uniformly distributed over $C^{-1}(C(w))$. In the remainder of the proof, whenever we consider the probability of an event over the choice of the collection $\mathcal{F}$, we mean that for each circuit C, two permutations f_C^1 and f_C^2 are chosen uniformly at random and independently of all other permutations. A complete and formal description of the oracle is provided in Figure 1.

On input a circuit $C : \{0,1\}^m \to \{0,1\}^{\ell(m)}$, the oracle $\mathsf{Sam}^{\tau,\mathcal{F}}$ acts as follows:

1. Compute $w = f_C^1(0^m)$.
2. Compute $w' = f_C^2(t)$ for the lexicographically smallest $t \in \{0,1\}^m$ such that $C(f_C^2(t)) = C(w)$.
3. Output (w, w')

Figure 1: The oracle Sam.

Distinguishing between injective functions and lossy functions. The oracle Sam can be easily used to distinguish between the injective mode and the lossy mode of any collection of $(n,1)$-lossy functions. Consider the following distinguisher A: given a circuit C (which may contain τ-gates[7]), which is a description of either an injective function or a lossy function (with image size at most 2^{n-1}), A queries Sam with C. If Sam returns (w, w') such that $w = w'$, then A outputs 1, and otherwise A outputs 0. Clearly, if C corresponds to an injective function, then always $w = w'$ and A outputs 1. In addition, if C corresponds to a lossy function, then with probability at least $1/4$ it holds that $w \neq w'$, where the probability is taken over the randomness of Sam (i.e., over the collection $\mathcal{F}$).

Outline of the proof. For simplicity we first consider only two permutations. Then, we extend our argument to more than two permutations, and to trapdoor permutations. Our goal is to upper bound the success probability of circuits having oracle access to Sam in the task of inverting $(\pi_1(x), \pi_2(x))$ for random permutations $\pi_1, \pi_2 \in \Pi_n$ and a random $x \in \{0,1\}^n$ (where Π_n is the set of all permutations over $\{0,1\}^n$). We prove the following theorem:

[7] We allow the circuits given as input to Sam to contain τ-gates, but we do not allow them to contain Sam-gates. This suffices, however, for ruling out fully-black-box constructions.

Theorem 4.1. *For any circuit A of size at most $2^{n/40}$ and for all sufficiently large n, it holds that*

$$\Pr_{\substack{\pi_1,\pi_2,\mathcal{F} \\ x \leftarrow \{0,1\}^n}} \left[A^{\pi_1,\pi_2,\mathsf{Sam}^{\pi_1,\pi_2,\mathcal{F}}}(\pi_1(x),\pi_2(x)) = x \right] \le \frac{1}{2^{n/40}} \ .$$

Consider a circuit A which is given as input $(\pi_1(x), \pi_2(x))$, and whose goal is to retrieve x. The idea underlying the proof is to distinguish between two cases: one in which A obtains information on x via one of its Sam-queries, and the other in which none of A's Sam-queries provides information on x. More specifically, we define:

Definition 4.1. *A Sam-query C produces a x-*hit *if Sam outputs (w, w') such that some π_1-gate or π_2-gate in the computations of $C(w)$ or $C(w')$ has input x.*

Given π_1, π_2, $\mathcal{F}$, a circuit A, and a pair $(\pi_1(x), \pi_2(x))$, we denote by SamHIT_x the event in which one of the Sam-queries made by A produces a x-hit. From this point on, the proof proceeds in two modular parts. In the first part of the proof, we consider the case that the event SamHIT_x does not occur, and generalize an argument of Haitner et al. [22] (who in turn generalized the reconstruction lemma of Gennaro and Trevisan [14]). We show that if a circuit A manages to invert $(\pi_1(x), \pi_2(x))$ for many x's, then π_1 and π_2 have a short representation given A. This enables us to prove the following lemma:

Lemma 4.1. *For any circuit A of size at most $2^{n/7}$ and for all sufficiently large n, it holds that*

$$\Pr_{\substack{\pi_1,\pi_2,\mathcal{F} \\ x \leftarrow \{0,1\}^n}} \left[A^{\pi_1,\pi_2,\mathsf{Sam}^{\pi_1,\pi_2,\mathcal{F}}}(\pi_1(x),\pi_2(x)) = x \ \wedge \ \overline{\mathsf{SamHIT}}_x \right] \le 2^{-n/8} \ .$$

In the second part of the proof, we show that the case where the event SamHIT_x does occur can be reduced to the case where the event SamHIT_x does not occur. Given a circuit A that tries to invert $(\pi_1(x), \pi_2(x))$, we construct a circuit M that succeeds almost as well as A, without M's Sam-queries producing any x-hits. This proof is a simpler case of a similar argument due to Haitner et al. [22]. The following theorem is proved:

Lemma 4.2. *For any circuit A of size $s(n)$, if*

$$\Pr_{\substack{\pi_1,\pi_2,\mathcal{F} \\ x \leftarrow \{0,1\}^n}} \left[A^{\pi_1,\pi_2,\mathsf{Sam}^{\pi_1,\pi_2,\mathcal{F}}}((\pi_1(x),\pi_2(x))) = x \right] \ge \frac{1}{s(n)}$$

for infinitely many values of n, then there exists a circuit M of size $O(s(n))$ such that

$$\Pr_{\substack{\pi_1,\pi_2,\mathcal{F} \\ x \leftarrow \{0,1\}^n}} \left[M^{\pi_1,\pi_2,\mathsf{Sam}^{\pi_1,\pi_2,\mathcal{F}}}((\pi_1(x),\pi_2(x))) = x \ \wedge \ \overline{\mathsf{SamHIT}}_x \right] \ge \frac{1}{s(n)^5}$$

for infinitely many values of n.

Due to space limitations the remainder of the proof is provided in the full version.

Acknowledgments

We thank Oded Goldreich, Moni Naor, Chris Peikert, and Omer Reingold for useful discussions. In particular, we thank Oded for suggesting the relaxation that led to the more generalized scheme.

References

1. A. Boldyreva, S. Fehr, and A. O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In *CRYPTO '08*, pages 335–359, 2008.
2. D. Boneh. Twenty years of attacks on the RSA cryptosystem. *Notices of the American Mathematical Society*, 46(2):203–213, 1999.
3. D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. *SIAM J. Comput.*, 36(5):1301–1328, 2007.
4. D. Boneh and J. Katz. Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In *CT-RSA '05*, pages 87–103, 2005.
5. X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based techniques. In *12th ACM CCS*, pages 320–329, 2005.
6. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT '04*, pages 207–222, 2004.
7. S. G. Choi, D. Dachman-Soled, T. Malkin, and H. Wee. Black-box construction of a non-malleable encryption scheme from any semantically secure one. In *5th TCC*, pages 427–444, 2008.
8. R. Cramer, G. Hanaoka, D. Hofheinz, H. Imai, E. Kiltz, R. Pass, A. Shelat, and V. Vaikuntanathan. Bounded CCA2-secure encryption. In *ASIACRYPT '07*, pages 502–518, 2007.
9. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO '98*, pages 13–25, 1998.
10. R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen-ciphertext attack. *SIAM J. Comput.*, 33(1):167–226, 2003.
11. D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000.
12. E. Elkind and A. Sahai. A unified methodology for constructing public-key encryption schemes secure against adaptive chosen-ciphertext attack. Cryptology ePrint Archive, Report 2002/042, 2002.
13. R. Gennaro, Y. Gertner, J. Katz, and L. Trevisan. Bounds on the efficiency of generic cryptographic constructions. *SIAM J. Comput.*, 35(1):217–246, 2005.
14. R. Gennaro and L. Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *41st FOCS*, pages 305–313, 2000.
15. Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan. The relationship between public key encryption and oblivious transfer. In *41st FOCS*, pages 325–335, 2000.
16. Y. Gertner, T. Malkin, and S. Myers. Towards a separation of semantic and CCA security for public key encryption. In *4th TCC*, pages 434–455, 2007.
17. Y. Gertner, T. Malkin, and O. Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates. In *42nd FOCS*, pages 126–135, 2001.

18. O. Goldreich. *Foundations of Cryptography – Volume 1: Basic Tools*. Cambridge University Press, 2001.
19. O. Goldreich, R. Impagliazzo, L. A. Levin, R. Venkatesan, and D. Zuckerman. Security preserving amplification of hardness. In *31st FOCS*, pages 318–326, 1990.
20. S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.
21. S. Goldwasser and V. Vaikuntanathan. New constructions of correlation-secure trapdoor functions and CCA-secure encryption schemes. Manuscript, 2008.
22. I. Haitner, J. J. Hoch, O. Reingold, and G. Segev. Finding collisions in interactive protocols – A tight lower bound on the round complexity of statistically-hiding commitments. In *48th FOCS*, pages 669–679, 2007.
23. J. Håstad. Solving simultaneous modular equations of low degree. *SIAM J. Comput.*, 17(2):336–341, 1988.
24. R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *21st STOC*, pages 44–61, 1989.
25. J. H. Kim, D. R. Simon, and P. Tetali. Limits on the efficiency of one-way permutation-based hash functions. In *40th FOCS*, pages 535–542, 1999.
26. Y. Lindell. A simpler construction of CCA2-secure public-key encryption under general assumptions. *J. Cryptology*, 19(3):359–377, 2006.
27. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd STOC*, pages 427–437, 1990.
28. R. Pass, A. Shelat, and V. Vaikuntanathan. Construction of a non-malleable encryption scheme from any semantically secure one. In *CRYPTO '06*, pages 271–289, 2006.
29. R. Pass, A. Shelat, and V. Vaikuntanathan. Relations among notions of non-malleability for encryption. In *ASIACRYPT '07*, pages 519–535, 2007.
30. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. Cryptology ePrint Archive, Report 2008/481, 2008.
31. C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *40th STOC*, pages 187–196, 2008.
32. C. Rackoff and D. R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *CRYPTO '91*, pages 433–444, 1991.
33. O. Reingold, L. Trevisan, and S. P. Vadhan. Notions of reducibility between cryptographic primitives. In *1st TCC*, pages 1–20, 2004.
34. A. Rosen and G. Segev. Efficient lossy trapdoor functions based on the composite residuosity assumption. Cryptology ePrint Archive, Report 2008/134, 2008.
35. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th FOCS*, pages 543–553, 1999.
36. D. R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In *EUROCRYPT '98*, pages 334–345, 1998.
37. A. C. Yao. Theory and applications of trapdoor functions. In *23rd FOCS*, pages 80–91, 1982.