On Minimal Assumptions for Sender-Deniable Public Key Encryption

Dana Dachman-Soled

University of Maryland
danadach@ece.umd.edu

Abstract. The primitive of deniable encryption was introduced by Canetti et al. (CRYPTO, 1997). Deniable encryption is an encryption scheme with the added feature that after transmitting a message m, both sender and receiver may produce random coins showing that the transmitted ciphertext was an encryption of any message m' in the message space. Deniable encryption is a key tool for constructing incoercible protocols, since it allows a party to send one message and later provide apparent evidence to a coercer that a different message was sent. In addition, deniable encryption may be used to obtain *adaptively*-secure multiparty computation (MPC) protocols and is secure under *selective-opening* attacks. Different flavors such as sender-deniable and receiver-deniable encryption, where only the sender or receiver produce fake random coins, have been considered.

Recently, over 15 years after the primitive was first introduced, Sahai and Waters (IACR Cryptology ePrint Archive, 2013), gave the first construction of sender-deniable encryption schemes with super-polynomial security, where an adversary has negligible advantage in distinguishing real and fake openings. Their construction is based on the construction of an indistinguishability obfuscator for general programs recently introduced in a breakthrough result of Garg et al. (FOCS, 2013). Although feasibility has now been demonstrated, the question of determining the *minimal* assumptions necessary for sender-deniable encryption with super-polynomial security remains open.

The primitive of simulatable public key encryption (PKE), introduced by Damgård and Nielsen (CRYPTO, 2000), is a public key encryption scheme with additional properties that allow oblivious sampling of public keys and ciphertexts. It is one of the low-level primitives used to construct adaptively-secure MPC protocols and was used by O'Neill et al. in their construction of bi-deniable encryption in the multi-distributional model (CRYPTO, 2011). Moreover, the original construction of sender-deniable encryption with polynomial security given by Canetti et al. can be instantiated with simulatable PKE. Thus, a natural question to ask is whether it is possible to construct sender-deniable encryption with *super-polynomial security* from simulatable PKE.

In this work, we investigate the possibility of constructing sender-deniable public key encryption from simulatable PKE in a black-box manner. We show that there is no black-box construction of sender-deniable public key encryption with super-polynomial security from simulatable PKE. This indicates that improving on the original construction of Canetti et al. requires the use of non-black-box techniques, stronger assumptions, or interaction, thus giving some evidence that strong assumptions such as those used by Sahai and Waters are necessary.

Keywords: sender-deniable encryption, simulatable PKE, black-box separation.

H. Krawczyk (Ed.): PKC 2014, LNCS 8383, pp. 574–591, 2014.

1 Introduction

Deniable encryption was first introduced by Canetti et al. [3]. In its strongest form, called bi-deniable encryption, this primitive allows a sender and receiver to communicate via a public key encryption scheme (sending some message m) and then later allows both parties to produce apparent evidence (i.e. secret key and random coins) that the ciphertext sent/received was actually an encryption of any message m' in the message space. Deniable encryption is useful for designing protocols that resist coercion (c.f. [5]) as well as for designing *adaptively*-secure protocols. Moreover, deniable encryption is is secure under *selective-opening* attacks. As a concrete example, consider a voting scheme where parties encrypt their votes using the voting authority's public key and send the ciphertext to the voting authority over a public channel. The voting authority is then trusted to decrypt and tally the votes[1]. In the voting scheme described, voters can carry away a *receipt*, the ciphertext sent to the authority along with the random coins used to encrypt, which can later be used to prove to a third party that a particular vote was cast. Although obtaining a receipt may seem desirable, it also means that voters or the voting authority can later be coerced by some third party to reveal the vote cast by a particular ciphertext. Thus, such a voting scheme is highly susceptible to coercion. However, using a bi-deniable encryption scheme instead of a regular public key encryption scheme allows both the voters and the authority to claim that a specific ciphertext corresponds to a vote for a particular candidate regardless of the actual effective vote. One may also consider weaker versions of bi-deniable encryption such as sender-deniable encryption and receiver-deniable encryption, where only the sender (resp. receiver) can produce fake coins.

Constructing deniable encryption schemes seems difficult due to two conflicting goals: Parties must be able to communicate effectively with each other, but if coerced, both parties must be able to produce seemingly correctly distributed randomness and/or secret keys consistent with *any* message m in the message space. Now it seems that surely deniability must interfere with effective communication since the receiver cannot tell which message m was the intended message and the sender cannot be assured that his intended message m was received. Indeed, it was shown by [2] that (non-interactive) receiver-deniable encryption (with negligible distinguishing advantage), and thus (non-interactive) bi-deniable encryption is impossible to achieve.

The case of *sender*-deniable encryption, however, is more optimistic. Indeed, very recently, Sahai and Waters [27], gave the first construction of sender-deniable encryption schemes with super-polynomial security, where an adversary has negligible advantage in distinguishing real and fake openings. Their construction is based on the construction of an indistinguishability obfuscator for general programs recently introduced in a breakthrough result of Garg et al. [12], and thus inherits the same non-standard hardness assumptions.

Prior to the result of [27], there were known constructions of deniable encryption (c.f. [3]) with *non-negligible* distinguishing advantage, where an adversary may distinguish real and fake openings of ciphertexts with probability $1/\operatorname{poly}$ for some polynomial.

[1] Alternatively, the voting authority may be required to give a zero-knowledge proof that the final tally is consistent with the transmitted ciphertexts.

We say that such schemes have *polynomial security*. As discussed in more detail below, the construction of [3] can be based on the existence of *simulatable* public key encryption, which can in turn be based on standard assumptions such as DDH and RSA.

This leaves open the following important question:

> What are the minimal assumptions required for sender-deniable public-key encryption with super-polynomial security?

Relationship to Adaptive Security and Simulatable Public Key Encryption

There is a strong link between deniable encryption and another primitive known as *non-committing encryption* [4]. The main difference between the two is that a Non-Committing Encryption scheme consists of two sets of Key Generation and Encryption algorithms–one for honest players and one for the simulator. Moreover, only honest parties need to communicate effectively, while only the simulator needs to equivocate ciphertexts. Both deniable encryption and non-committing encryption can be used to achieve *adaptively* secure multiparty computation and both are secure under *selective opening* attacks. One of the standard low-level assumptions used to construct non-committing encryption is a primitive known as *simulatable public key encryption (PKE)* introduced by Damgård and Nielsen[9][2]. Loosely speaking, a simulatable public key encryption scheme is an encryption scheme with special algorithms for obliviously sampling public keys and random ciphertexts without learning the corresponding secret keys and plaintexts; in addition, both of these oblivious sampling algorithms should be eficiently invertible. Simulatable public key encryption schemes can be based on the assumptions of DDH and RSA[3].

Simulatable public key encryption has been a useful tool for constructing variants of deniable encryption. O'Neill et al. showed how to use simulatable PKE to construct bi-deniable encryption in the multi-distributional model [24]. Moreover, it is not hard to see that the original construction of sender-deniable public key encryption given by [3] can be instantiated with simulatable PKE instead of trapdoor permutations, although in their paper they do not explicitly use simulatable PKE.

Thus, a natural and imperative direction to explore is whether it is possible to construct sender-deniable encryption with super-polynomial security from simulatable PKE.

Our Results

We consider the possibility of constructing non-interactive sender-deniable encryption, known as *sender-deniable public key encryption*, with super-polynomial security in a *black-box manner* from simulatable PKE. We provide a negative answer to the above question by showing the following:

Theorem 1 (Main Theorem, Informal). *There is no (fully) black-box reduction of sender-deniable public key encryption with super-polynomial security to simulatable PKE.*

[2] In fact, an even weaker primitive called *trapdoor-simulatable PKE* [6] is sufficient for non-committing encryption.

[3] Trapdoor-simulatable PKE can be constructed from these assumptions as well as hardness of factoring.

In particular, we show that every black-box construction of a sender-deniable public key encryption scheme from simulatable PKE which makes $m = m(n)$ queries to the simulatable PKE scheme cannot achieve security better than $O(m^4(n))$. Our results indicate that improving upon the original scheme of [3] requires the use of non-black-box techniques, stronger underlying assumptions or interaction thus giving some evidence that strong assumptions such as those used by Sahai and Waters [27] are necessary.

Black-Box Separations

Impagliazzo and Rudich [19] were the first to develop a technique to rule out the existence of an important class of reductions between primitives known as black-box reductions. Indeed, most known reductions between cryptographic primitives are black-box (see the works of [28,16,26,17,15,20,18,23,22] for a small sampling). Intuitively, black-box reductions are reductions where the primitive is treated as an oracle or a "black-box". There are actually several flavors of black-box reductions (fully black-box, semi black-box and weakly black-box [25]). In our work, we only deal with fully black-box reductions, and so we will focus on this notion here. Informally, a fully black-box reduction from a primitive $\mathcal{Q}$ to a primitive $\mathcal{P}$ is a pair of *oracle* $\mathcal{PPT}$ Turing machines (G, S) such that the following two properties hold:

Correctness: For every implementation f of primitive $\mathcal{P}$, $g = G^f$ implements $\mathcal{Q}$.

Security: For every implementation f of primitive $\mathcal{P}$, and every adversary A, if A breaks G^f (as an implementation of $\mathcal{Q}$) then $S^{A,f}$ breaks f. (Thus, if f is "secure", then so is G^f.)

We remark that an *implementation* of a primitive is any specific scheme that meets the requirements of that primitive (e.g., an implementation of a public-key encryption scheme provides samplability of key pairs, encryption with the public-key, and decryption with the private key). Correctness thus states that when G is given oracle access to any valid implementation of $\mathcal{P}$, the result is a valid implementation of $\mathcal{Q}$. Furthermore, security states that any adversary breaking G^f yields an adversary breaking f. The reduction here is *fully* black-box in the sense that the adversary S breaking f uses A in a black-box manner.

Our Techniques

Following the paradigm introduced by [19], we define an oracle O and consider constructions of simulatable PKE and sender-deniable public key encryption relative to this oracle. The oracle O that we use is similar to the by now standard oracle first introduced by [13]. This oracle implements an ideal trapdoor function with the important property that it is difficult to obliviously sample from the range of the function. Namely, it is hard to find an image in the range of the function without first sampling the corresponding preimage.

Relative to the oracle O, we show the following:

- There exists a simulatable PKE scheme, $\mathcal{E}_{\mathsf{Sim}}$ secure against all (computationally unbounded) adversaries $\mathcal{A}$ making at most polynomial number of queries.

– For every implementation $\mathcal{E}$ of a sender-deniable public key encryption scheme relative to O, there exists an adversary $\mathcal{A}$ making at most polynomial number of queries such that $\mathcal{A}$ breaks $\mathcal{E}$.

The above is sufficient to imply that there is no fully black-box construction of sender-deniable public key encryption from simulatable PKE.

Now, recall that a sender-deniable public key encryption scheme is a public key encryption scheme with an additional algorithm, Fake, which takes an honestly generated sender's view $\mathsf{View}_{\mathsf{S}_0}$ encrypting a bit b and returns a fake view, $\mathsf{View}_{\mathsf{S}_1} = \mathsf{Fake}(\mathsf{View}_{\mathsf{S}_0})$, encrypting the bit $1-b$. A simple but key observation is the following: If the distributions over the corresponding views, $\mathsf{View}_{\mathsf{S}_0}$ and $\mathsf{View}_{\mathsf{S}_1}$ are indistinguishable, then one should be able to now compute $\mathsf{View}_{\mathsf{S}_2} = \mathsf{Fake}(\mathsf{View}_{\mathsf{S}_1})$ obtaining a fake view encrypting the bit b and such that the distributions over the views, $\mathsf{View}_{\mathsf{S}_1}$ and $\mathsf{View}_{\mathsf{S}_2}$ are again indistinguishable. We note that somewhat similar arguments were used in [2]. In general, in any sender-deniable public key encryption scheme with negligible distinguishing advantage, one must be able to run Fake iteratively on the output of the previous Fake invocation for any (unbounded) polynomial number of times. Otherwise, if there is a fixed polynomial upper bound $p(n)$ on the number of times that Fake can be applied to a fresh ciphertext (before failure), then we can distinguish $\mathsf{View}_{\mathsf{S}_0}$ from $\mathsf{View}_{\mathsf{S}_{p(n)}} = \perp = \mathsf{Fake}^{p(n)}(\mathsf{View}_{\mathsf{S}_0})$ (where by $\mathsf{Fake}^{p(n)}$ we denote the composition of Fake, $p(n)$ times). So by a hybrid argument there must be some i such that $\mathsf{Fake}^{i}(\mathsf{View}_{\mathsf{S}_0})$, $\mathsf{Fake}^{i+1}(\mathsf{View}_{\mathsf{S}_0})$ can be distinguished with probability $1/p(n)$. Finally, this means that real and fake openings $\mathsf{View}_{\mathsf{S}_0}$ and $\mathsf{View}_{\mathsf{S}_1}$ can be distinguished, contradicting the security of the sender-deniable public key encryption scheme[4]. Thus, in order to prove the lower bound it is sufficient to show that relative to our oracle, Fake can be repeatedly applied only a fixed polynomial number of times before failure.

To gain some intuition for why this is the case, it is instructive to recall the construction of [3][5]. Let $\{F_{pk}\}$ be a family of trapdoor functions with pseudorandom range such that given the secret key sk of F_{pk}, one can distinguish between elements y in the range of F_{pk} and random elements, but given only pk, random elements in the range of F_{pk} are indistinguishable from random strings. In [3], the secret key of the sender-deniable public key encryption scheme is the secret key sk of the trapdoor function F. The public key pk is the public key of F. Each ciphertext consists of m number of strings $s_1, \ldots, s_m$. To encrypt a 1, choose an a set of indeces $I \subseteq [m]$ of odd cardinality; otherwise choose a set $I \subseteq [m]$ of even cardinality. Compute m strings in the following way: For the i-th string, if $i \in I$, choose a random x_i and compute $y_i = f(x_i)$. If $i \notin I$, choose y_i to be a random string. The sender sends these m strings to the receiver. The receiver then checks which of the m strings $y_1, \ldots, y_m$ are valid images. If an odd number of strings are valid, output 1. Otherwise, output 0. It is not hard to see that the Fake algorithm works by having the sender claim that a pseudorandom string is really random (but note that the sender cannot claim the reverse).

Clearly, the Fake algorithm described above can be run iteratively at most m times for a given ciphertext, since the sender claims to have made one less query each time Fake is run and there are at most m queries total. Unfortunately, our analysis is more

[4] Simply run Fake iteratively i number of times on $\mathsf{View}_{\mathsf{S}_0}$ and then use the distinguisher above.

[5] We simplify their construction here somewhat.

complicated since we must also consider candidate schemes where the Fake algorithm might *add* queries to the outputted view. It may seem at first glance that it is impossible for Fake to add new queries to the sender's view that were not in the original view since it would seem to require inverting a random image y without access to the corresponding secret key. However, this is not necessarily the case (see the full version [7] for a toy example where this occurs).

Thus, we must show that even for candidate schemes whose Fake algorithms may both remove and add queries, Fake can be repeatedly applied only a fixed polynomial number of times before failure. Intuitively, the reason we can handle such schemes is that it is infeasible to add an unbounded number of new queries to the fake view, since many queries must be removed from the previous view for each new query that is added. In order to show that this intuition indeed holds, we leverage the fact that in our oracle, with overwhelming probability, random strings are not valid images of the trapdoor function. Much of the technical part of the proof is in showing that the above intuition holds for *all* possible constructions of sender-deniable public key encryption schemes relative to our oracle.

Technical Overview of Proof. The high-level approach of the proof will be to consider the distribution $\mathcal{D}_{\mathsf{Fake}}^{10m^2(n)}$, where $m(n)$ is the maximum number of queries made by sender and receiver, and a draw from $\mathcal{D}_{\mathsf{Fake}}^{10m^2(n)}$ is obtained in the following way:

- Draw an oracle O and original views, $\mathsf{View}_{\mathsf{S}_0}, \mathsf{View}_{\mathsf{R}}$, for sender and receiver from the correct distributions.
- For $1 \leq i \leq 10m^2(n)$, set $\mathsf{View}_{\mathsf{S}_i} = \mathsf{Fake}^{\mathsf{O}}(\mathsf{View}_{\mathsf{S}_{i-1}})$.
- Output $\mathsf{O}, \mathsf{View}_{\mathsf{R}}, \mathsf{View}_{\mathsf{S}_0}, \ldots, \mathsf{View}_{\mathsf{S}_{10m^2(n)}}$

In our analysis, we will look at the properties of sequences of fake openings $\mathsf{View}_{\mathsf{S}_0}$, $\ldots$, $\mathsf{View}_{\mathsf{S}_{10m^2(n)}}$ drawn from this distribution. Note that for any sender-deniable public key encryption scheme it should (at the very least) be the case that w.v.h.p. for every consecutive $i, i+1$, $\mathsf{View}_{\mathsf{S}_i}$ and $\mathsf{View}_{\mathsf{S}_{i+1}}$ are valid encryptions of bits b_i and $b_{i+1} = 1 - b_i$, respectively. Furthermore, we show that if a public key encryption scheme has the deniability property then with high probability a sequence drawn from $\mathcal{D}_{\mathsf{Fake}}^{10m^2(n)}$ will have several additional properties. However, we will also argue that it is impossible for a sequence of fake openings of length $10m^2(n)$ to satisfy all of the required properties simultaneously. Thus, a sequence drawn from $\mathcal{D}_{\mathsf{Fake}}^{10m^2(n)}$ will with high probability not satisfy at least one of the required properties. This leads to contradiction and so we conclude that the encryption scheme is not sender-deniable.

In what follows, we give a slightly innacurate but intuitive overview of what these properties are and the techniques we use to prove that with high probability a sequence of fake openings will possess these properties.

First, note that a fake opening is simply a view $\mathsf{View}_{\mathsf{S}_i}$ of the sender which consists of a transcript, W (i.e. a public key, PK, and ciphertext c), and a set of queries $Q(\mathsf{S}_i)$ made by the sender. We also consider the set $Q(E)_i$ which, intuitively, is a set of queries that includes all queries the honest sender (with view $\mathsf{View}_{\mathsf{S}_i}$) believes may have been made by both him and the receiver. The set of queries in $Q(E)_i$ can be found by running

an algorithm that is very similar to the Eve algorithm of [1], which finds intersection queries based only on the transcript (and does not depend on the sender's view, as in our case). During the execution of the Eve algorithm, Eve finds pairs (pk^*, y^*) such that it is likely the sender queried $F(pk^*, x) = y^*$ for some x. If Eve identifies a such a pair (pk^*, y^*) and, indeed, a corresponding $F(pk^*, x^*) = y^*$ is found in $\mathsf{View}_{\mathsf{S}_i}$, then the query is "added" and placed in Q_i^{made}. If Eve identifies a such a pair (pk^*, y^*), however, and no corresponding $F(pk^*, x^*) = y^*$ is found in $\mathsf{View}_{\mathsf{S}_i}$, then the query is "removed" and placed in Q_i^{skipped}.

Now for each fake opening $\mathsf{View}_{\mathsf{S}_i}$ we consider two types of queries "A" type queries and "B" type queries. Intuitively, "A" type queries are those queries that were originally in $\mathsf{View}_{\mathsf{S}_0}$ and have either not been removed in some Q_j^{skipped} set (for $j \leq i$), or were removed and then added again in some Q_k^{made} set (for $j < k \leq i$). "B" type queries are new queries that do not appear in the original view $\mathsf{View}_{\mathsf{S}_0}$, were added in some Q_j^{made} set (for $j \leq i$) and have not been subsequently removed in a Q_k^{skipped} set (for $j < k \leq i$). Thus, each view $\mathsf{View}_{\mathsf{S}_i}$ is associated with a set, A^i, of "A" type queries and a set, B^i, of "B" type queries.

We will show that with high probability a draw of fake openings $\mathsf{View}_{\mathsf{S}_0}, \ldots, \mathsf{View}_{\mathsf{S}_{10m^2(n)}}$ and corresponding sequence $(A^0, B^0), \ldots, (A^{10m^2(n)}, B^{10m^2(n)})$ must satisfy the following properties:

- $(\mathsf{View}_{\mathsf{S}_0}, \mathsf{View}_{\mathsf{S}_1}, \ldots, \mathsf{View}_{\mathsf{S}_{10m^2(n)}})$ are valid openings.
- $A^i \subseteq A^{i-1}$ for $1 \leq i \leq 10m^2(n)$
- $(A^{i-1}, B^{i-1}) \neq (A^i, B^i)$ for $1 \leq i \leq 10m^2(n)$
- If the same set A^* appears consecutively β times within the sequence above, and all corresponding consecutive B sets are different, then $\beta \leq 10m(n)$.

Much of the technical portion of this work is dedicated to showing that these properties hold (see Claim 2, Lemma 4 and Lemma 5). Then, we will show that it is, in fact, impossible to realize all of the above properties simultaneously (see the end of Section 5).

Related Work

In their seminal paper, Canetti et al. [3] introduce the primitive of deniable encryption and present constructions. However, for the strongest form of deniable encryption which assumes that the same key generation and encryption algorithms are always used, [3] achieve only sender-deniable and receiver-deniable schemes with polynomial security. [3] also rule out the existence of a specific type of sender-deniable encryption scheme with negligible distinguishing advantage (or super-polynomial security) called *separable* schemes (which, roughly speaking, are a generalization of the scheme of [3]). Our impossibility result is incomparable to theirs since ours rules out a larger class of reductions (black-box reductions), but only rules out reductions to the specific primitive of simulatable PKE.

O'Neill et al. [24] recently constructed a bi-deniable encryption scheme in the multi-distributional model, in which the parties run alternative key-generation and encryption algorithms for equivocable communication, but claim under coercion to have run the

prescribed algorithms. This weaker model was also initially considered by [3]. Although useful in some settings, the multi-distributional model does not achieve the strongest form of deniability which we consider in this work. We note that it is essential for our impossibility result that the *same* encryption algorithm is run for both real and equivocable communication, which is why our result does not contradict the work of [24].

Recently, Dürmuth and Freeman announced a fully-deniable (receiver/sender)-deniable interactive cryptosystem with negligible security [10]. However their result was later showed to be incorrect by Peikert and Waters (see [11] for details). The protocol constructed by [10] was both interactive and utilized the fact that for the trapdoor function used, a random element in the range could be sampled obliviously. We note that in our analysis it is essential both that the schemes we consider are non-interactive and that the trapdoor function implemented by our oracle does not allow oblivious sampling of the range. Thus, an interesting open question is whether removing these two restrictions can help achieve fully-deniable encryption schemes.

Subsequently, [2] showed, using an information-theoretic argument, that (non-interactive) receiver-deniable encryption with negligible distinguishing advantage do not exist, unconditionally. We note, however, that the work of [2] does not address the case of sender-deniable encryption and it does not seem that their techniques may be applied to our case.

Recently, Sahai and Waters [27] showed how to construct sender-deniable encryption from indistinguishability ofuscation. In a breakthrough result, a candidate construction of an indistinguishability obfuscator for general programs was put forward by Garg et al. [12]. In their followup paper, [27] show that indistinguishability obfuscation can be used to achieve sender-deniable encryption[6] We note that the candidate construction of [12] is based on newly introduced hardness assumptions such as "multilinear jigsaw puzzles". Thus, the construction of [27] also requires these non-standard assumptions.

Organization

In Section 2 we formally define sender-deniable public key encryption and simulatable PKE as well as the notion of a black-box construction of sender-deniable public key encryption from simulatable PKE. In Section 3 we define our oracle and in Section 4 we define some additional useful notations, algorithms and corresponding properties which will be used in the main result. Finally, in Section 5 we prove our main theorem, with some technical parts deferred to the Appendix.

2 Definitions

Definition 1 (Sender-Deniable Public Key Encryption). *A* sender-deniable (bit) public key encryption scheme *is a tuple of algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake})$ *defined as follows:*

- *The key-generation, encryption and decryption algorithms* $\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}$ *are defined as usual for public-key encryption.*

[6] Simply called "deniable encryption" in their work.

– *The* sender faking algorithm $\mathsf{Fake}(\mathrm{PK}, r_S, b)$*, given a public key* PK*, original coins* r_S *and bit* b *of* Enc*, outputs faked random coins* r_S^* *for* Enc *and the bit* $1-b$*.*

We require the following properties:

Correctness. $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *forms a correct public-key encryption scheme*[7]*.*

Deniability. For $b \in \{0,1\}$*, we require that the following two probability ensembles are computationally indistinguishable:*

– $\{(\mathrm{PK}, c, r_S) | \mathrm{PK} \leftarrow \mathsf{Gen}(1^n; r_G), c \leftarrow \mathsf{Enc}(\mathrm{PK}, b; r_S)\}_n$
– $\{(\mathrm{PK}, c, r_S^*) | \mathrm{PK} \leftarrow \mathsf{Gen}(1^n; r_G), c \leftarrow \mathsf{Enc}(\mathrm{PK}, 1-b; r_S), r_s^* \leftarrow \mathsf{Fake}(\mathrm{PK}, r_S, b)\}_n$

It follows from the definition that a sender-deniable public key encryption scheme is also semantically secure.

Remark 1. In this work, we also consider constructions of deniable public key encryption schemes that do not achieve negligible distinguishing advantage. We say that a deniable encryption scheme has security $p(n)$ for some polynomial $p(\cdot)$ if correctness holds and every probabilistic polynomial time adversary $\mathcal{A}$ distinguishes the following two probability ensembles with advantage at most $1/p(n)$:

– $\{(\mathrm{PK}, c, r_S) | \mathrm{PK} \leftarrow \mathsf{Gen}(1^n; r_G), c \leftarrow \mathsf{Enc}(\mathrm{PK}, b; r_S)\}_n$
– $\{(\mathrm{PK}, c, r_S^*) | \mathrm{PK} \leftarrow \mathsf{Gen}(1^n; r_G), c \leftarrow \mathsf{Enc}(\mathrm{PK}, 1-b; r_S), r_s^* \leftarrow \mathsf{Fake}(\mathrm{PK}, r_S, b)\}_n$.

We note that in this case semantic security does not follow from deniability and is an additional requirement.

Definition 2 (Simulatable PKE). *A* ℓ-bit simulatable encryption scheme *consists of an encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *augmented with* $(\mathsf{oGen}, \mathsf{oRndEnc}, \mathsf{rGen}, \mathsf{rRndEnc})$*. Here,* oGen *and* oRndEnc *are the oblivious sampling algorithms for public keys and ciphertexts, and* rGen *and* rRndEnc *are the respective inverting algorithms,* rGen *(resp.* rRndEnc*) takes* r_{G} *(resp.* $(\mathrm{PK}, r_{\mathrm{E}}, m)$*) as the trapdoor information. We require that, for all messages* $m \in \{0,1\}^\ell$*, the following distributions are computationally indistinguishable:*

$$\{\mathsf{rGen}(r_{\mathrm{G}}), \mathsf{rRndEnc}(\mathrm{PK}, r_{\mathrm{E}}, m), \mathrm{PK}, c \mid (\mathrm{PK}, \mathrm{SK}) = \mathsf{Gen}(1^k; r_{\mathrm{G}}), c = \mathsf{Enc}_{\mathrm{PK}}(m; r_{\mathrm{E}})\}$$
$$\text{and } \{\hat{r}_{\mathrm{G}}, \hat{r}_{\mathrm{E}}, \hat{\mathrm{PK}}, \hat{c} \mid (\hat{\mathrm{PK}}, \perp) = \mathsf{oGen}(1^k; \hat{r}_{\mathrm{G}}), \hat{c} = \mathsf{oRndEnc}_{\hat{\mathrm{PK}}}(1^k; \hat{r}_{\mathrm{E}})\}$$

It follows from the definition that a simulatable encryption scheme is also semantically secure.

Definition 3 (Sender-Deniable Public Key Encryption from Simulatable PKE). *For oracle algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake})$ *we call* $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake})$ *a black-box construction of sender-deniable public key encryption based on simulatable PKE if the following properties hold:*

– **Implementation:** *The algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake})$ *get oracle access to simulatable PKE scheme* $\mathcal{E}_{\mathsf{Sim}}$ *and* $\mathcal{E}$ *is an implementation of sender-deniable public key encryption.*

[7] Note that *perfect* correctness is not possible.

– **Security:** *There is a polynomial-time oracle algorithm S with the following property. For any simulatable PKE $\mathcal{E}_{\mathsf{Sim}} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{oGen}, \mathsf{oRndEnc}, \mathsf{rGen}, \mathsf{rRndEnc})$, given as oracle, if $\mathcal{A}$ breaks the security of $\mathcal{E}$ then $S^{\mathcal{E}_{\mathsf{Sim}}, \mathcal{A}}$ breaks the security of $\mathcal{E}_{\mathsf{Sim}}$.*

3 Oracle

The oracle O consists of three functions G, F, F^{-1} defined below for every security parameter n.

– $G : \{0,1\}^n \to \{0,1\}^{3n}$ is an injective function taking inputs sk of length n bits to outputs pk of length $3n$ bits.
– $F : \{0,1\}^{4n} \to \{0,1\}^{12n}$ is an injective function taking inputs pk, x of length $4n$ bits to outputs y of length $12n$ bits.
– $F^{-1} : \{0,1\}^{13n} \to \{0,1\}^n$ takes inputs of the form sk, y where $sk \in \{0,1\}^n$ and $y \in \{0,1\}^{12n}$. F^{-1} returns $x \in \{0,1\}^n$ if $G(sk) = pk$ and $F(pk, x) = y$ and $\perp$ otherwise.

Note that the oracle above behaves like a trapdoor function, where G is the key generation functionality, F evaluates the trapdoor function and F^{-1} is the inversion function. Additionally, note that we may easily construct a simulatable PKE scheme relative to this oracle.

We denote by Υ the uniform distribution over all possible oracles O.

Lemma 1. *There is a construction of a simulatable PKE scheme $\mathcal{E}_{\mathsf{Sim}}$ relative to oracle O, such that for every unbounded adversary $\mathcal{A}$, making a polynomial number of queries to O:*

$$\Pr_{\mathsf{O} \sim \Upsilon}[\mathcal{A}^{\mathsf{O}} \textit{ breaks } \mathcal{E}^{\mathsf{O}}_{\mathsf{Sim}}] \leq \mathrm{neg}(n).$$

The proof of the Lemma above is by now standard (c.f. [13,14]) and so we omit it.

4 Preliminaries

In this section we introduce some useful notation, algorithms and properties of sender-deniable public key encryption schemes.

Given a deniable public key encryption scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake})$, we will consider the natural two-message protocol $\langle \mathsf{S}, \mathsf{R} \rangle$ between a *Receiver*, R (who sends a public key in the first message) and a *Sender*, S (who sends a ciphertext in the second message).

The view of the Receiver (resp. Sender) consists of the transcript W, random tape, r_R (resp. r_S) and queries made to the oracle along with the responses. The view of the Receiver, denoted by $\mathsf{View}_\mathsf{R} = (\mathsf{View}_G, \mathsf{View}_D)$, consists of two parts where View_G includes queries and responses made during Gen and View_D includes queries and responses made during Dec. The view of the Sender, denoted by View_S includes queries and responses made during Enc. We denote the queries to O in View_R by $Q(\mathsf{R}) = Q(G) \cup Q(D)$. We denote the queries to O in View_S by $Q(\mathsf{S})$.

We assume without loss of generality that:

- No queries to F^{-1} are made during Gen. This is WLOG since with overwhelming probability either the corresponding query to F was already made, or F^{-1} returns $\perp$.
- Each party queries $G(sk) = pk$ before querying $F^{-1}(sk, y)$.
- Either Fake returns a valid opening or returns $\perp$ and $\mathsf{Fake}(\perp) = \perp$.

Additionally, relative to our oracle O, we assume WLOG that Fake takes $\mathsf{View_S}$ and returns another $\mathsf{View_S}$ with the *same* public key and ciphertext but *different* randomness and input bit (i.e. $\mathsf{View}_{\mathsf{S}_{i+1}} = \mathsf{Fake}^{\mathsf{O}}(\mathsf{View}_{\mathsf{S}_i})$). By $\mathsf{Fake}^{\mathsf{O},i}$ we denote composing Fake with itself i times.

4.1 Useful Distributions

Distribution $\mathcal{D}$: $\mathcal{D}$ is a distribution over tuples $(\mathsf{View_S}, \mathsf{View_R})$ resulting from an execution of $\langle \mathsf{S}, \mathsf{R} \rangle$. A draw from $\mathcal{D}$ is obtained as follows:

- Draw $\mathsf{O} \sim \Upsilon$, $b \leftarrow \{0,1\}$, $\mathsf{r_R}, \mathsf{r_S} \leftarrow \{0,1\}^{p(n)}$, for some polynomial $p(\cdot)$ and execute $\langle \mathsf{S}, \mathsf{R} \rangle$ with $\mathsf{O}, \mathsf{r_R}, \mathsf{r_S}$ and input bit b.
- Output: The views $(\mathsf{View_S}, \mathsf{View_R})$ resulting from the execution of $\langle \mathsf{S}, \mathsf{R} \rangle$ above.

Distribution $\mathcal{D}^i$: $\mathcal{D}^i$ is a distribution over tuples $(\mathsf{View}_{\mathsf{S}_i}, \mathsf{View_R})$ as before, but here we begin to use the Fake algorithm. A draw from $\mathcal{D}^i$ is obtained as follows:

- Draw $\mathsf{O} \sim \Upsilon$, $b \leftarrow \{0,1\}$, $\mathsf{r_R}, \mathsf{r_S} \leftarrow \{0,1\}^{p(n)}$. and execute $\langle \mathsf{S}, \mathsf{R} \rangle$ with $\mathsf{O}, \mathsf{r_R}, \mathsf{r_S}$ and input bit b.
- Let $\mathsf{View}_{\mathsf{S}_0} = \mathsf{View_S}, \mathsf{View_R}$ containing $\mathrm{PK}, c, b, \mathsf{r_S}$ be the resulting views from the execution of $\langle \mathsf{S}, \mathsf{R} \rangle$. Compute $\mathsf{View}_{\mathsf{S}_i} = \mathsf{Fake}^{\mathsf{O},i}(\mathsf{View_S})$.
- Output: O and the views $(\mathsf{View}_{\mathsf{S}_i}, \mathsf{View_R})$.

For every fixed polynomial $p(\cdot)$, we additionally define the following distribution:

Distribution $\mathcal{D}^{p(n)}_{\mathsf{Fake}}$: $\mathcal{D}^{p(n)}_{\mathsf{Fake}}$ is a distribution over tuples $(\mathsf{O}, \mathsf{View_S}, \mathsf{View}_{\mathsf{S}_1}, \ldots, \mathsf{View}_{\mathsf{S}_{p(n)}})$. A draw from $\mathcal{D}^{p(n)}_{\mathsf{Fake}}$ is obtained as follows:

- Draw $\mathsf{O} \sim \Upsilon$, $b \leftarrow \{0,1\}$, $\mathsf{r_R}, \mathsf{r_S} \leftarrow \{0,1\}^{p(n)}$. and execute $\langle \mathsf{S}, \mathsf{R} \rangle$ with $\mathsf{O}, \mathsf{r_R}, \mathsf{r_S}$ and input bit b.
- Let $\mathsf{View}_{\mathsf{S}_0} = \mathsf{View_S}, \mathsf{View_R}$ containing $\mathrm{PK}, c, b, \mathsf{r_S}$ be the resulting views from the execution of $\langle \mathsf{S}, \mathsf{R} \rangle$.
- Output: $(\mathsf{O}, \mathsf{View_R}, \mathsf{View}_{\mathsf{S}_0} = \mathsf{View_S}, \mathsf{View}_{\mathsf{S}_1} = \mathsf{Fake}^{\mathsf{O}}(\mathsf{View_S}), \mathsf{View}_{\mathsf{S}_2} = \mathsf{Fake}^{\mathsf{O}}(\mathsf{View}_{\mathsf{S}_1}), \ldots, \mathsf{View}_{\mathsf{S}_{p(n)}} = \mathsf{Fake}^{\mathsf{O}}(\mathsf{View}_{\mathsf{S}_{p(n)-1}}))$.

4.2 Algorithms for Finding Likely Queries

As in [19,1,8,13,21], we will be concerned with finding *intersection queries*, or common information about the oracle shared by S and R. We note that in our setting there are two ways to get an intersection query:

- One party makes a query of the form $G(sk) = pk$, $F(pk, x) = y$, or $F^{-1}(sk, y)$ and the other party makes the same query.
- One of the parties queries both $G(sk)$, $F^{-1}(sk, y) = x$ and the other party queries $F(pk, x) = y$.

We now (informally) define the Eve algorithm: For a more formal specification, see the full version [7]. Eve runs the following algorithm, using threshold $\varepsilon = \varepsilon_1 = 1/m^{16}$ during the first pass (before S sends its message) and using threshold $\varepsilon = \varepsilon_2 = 1/m^6$ during the second pass (after S sends its message).

(0) *Eve queries F on all possible inputs up to length $4\hat{n} = 4\log(10m^{34})$ and adds all queries and responses to E.*

(1) *As long as there exists a query q of the form $G(sk)$, $F(pk, x)$, or $F^{-1}(sk, y)$ that was previously made by* S *or* R *with probability at least ε (conditioned on* Eve*'s current knowledge, E), then ask q from the oracle and add q paired with its answer to E.*

(2) *As long as there exists a pair (pk^*, y^*) such that $G(sk) = pk^* \in Q(E)$, $F(pk^*, x) = y^* \notin Q(E)$ and with probability at least ε,* R *made a query of the form $F(pk^*, x) = y^*$ for some x (conditioned on* Eve*'s current knowledge, E), then query the oracle on $F^{-1}(sk, y^*)$. If $F^{-1}(sk, y^*)$ returns some value x, then add $F(pk^*, x) = y^*$ to E. If $F^{-1}(sk, y^*)$ returns $\perp$ then add $F^{-1}(sk, y^*) = \perp$ to E.*

(3) *As long as there exists a pair (pk^*, y^*) such that $F(pk^*, x) = y^* \notin Q(E)$ and with probability at least ε,* S *made a query of the form $F(pk^*, x) = y^*$ for some x (conditioned on* Eve*'s current knowledge, E), then if $F(pk^*, x) = y^* \in Q(\mathsf{S})$, add q paired with its answer to E and add (pk^*, y^*) to Q^{made}. Otherwise, add (pk^*, y^*) to Q^{skipped}.*

We denote by $Q(E)_G$ the Eve queries made after the first message is sent from R to S and denote by $Q(E)_{\mathsf{S}}$ the Eve queries made after the second message is sent from S to R. Thus $Q(E) = Q(E)_G \cup Q(E)_{\mathsf{S}}$.

The following Lemma appeared in [8], but there was proven with respect to a random oracle.

Lemma 2. *Let $\langle \mathsf{S}, \mathsf{R} \rangle$ be a protocol as specified above in which the Sender and Receiver ask at most $2m$ queries each from the oracle* O*. Then there is a universal constant c such that on input parameter ε:*

- (cm/ε)**-Efficiency:** *Eve is deterministic and, over the randomness of the oracle and* S *and* R*'s private randomness, the expected number of Eve queries from the oracle* O *is at most cm/ε_1.*
- $(c\sqrt{m\varepsilon})$**-Security:** *Let W be the transcript of messages sent between* R *and* S *so far, and let E be the additional information that Eve has learned till the* end *of the i'th round. We denote by $Q(E)$ the oracle query/answer pairs that Eve has asked. Let $\mathcal{D}(W, E)$ be the* joint *distribution over the views $(\mathsf{View}_{\mathsf{S}}, \mathsf{View}_{\mathsf{R}})$ of* S *and* R only *conditioned on (W, E). By $\mathcal{D}_{\mathsf{R}}(\cdot, \cdot)$ and $\mathcal{D}_{\mathsf{S}}(\cdot, \cdot)$ we refer to the projections of $\mathcal{D}(W, E)$ over its first or second components.*
  With probability at least $1 - c\sqrt{m\varepsilon}$ over the randomness of S*,* R*, and the random oracle* O *the following holds at* all *moments during the protocol when Eve is*

done with her learning phase in that round: There are independent *distributions* $\mathcal{S}(W,E), \mathcal{R}(W,E)$ *such that:*

1. *The statistical distance between* $\mathcal{S}(W,E) \times \mathcal{R}(W,E)$ *and* $\mathcal{D}(W,E)$ *is at most* $\Delta(\mathcal{S}(W,E) \times \mathcal{R}(W,E), \mathcal{D}(W,E)) \leq c\sqrt{m\varepsilon}$.
2. *For every oracle query* $q \notin Q(E)$ *it holds that* $\Pr_{(\mathsf{View_S} \sim \mathcal{S}(W,E), \mathsf{View_R} \sim \mathcal{R}(W,E)}$ $[q \in Q(\mathsf{S}) \cup Q(\mathsf{R})] \leq \varepsilon$.

– **Robustness.** *The learning algorithm is robust to the input parameter* ε *in the following sense. If the parameter* ε *changes in the interval* $\varepsilon \in [\varepsilon_1, \varepsilon_2]$ *arbitrarily during the learner's execution (even inside a learning phase of a specific round), it still preserves* $O(cm/\varepsilon_1)$*-efficiency and* $(c\sqrt{m\varepsilon_2})$*-security.*

See the full version [7] for the proof of Lemma 2 which is based on the proofs found in [1,8,21].

Remark 2. Note that the Eve algorithm as described above requires knowledge of $\mathsf{View_S}$ but not of $\mathsf{View_R}$. Thus, Eve can only be simulated by a party who has knowledge of $\mathsf{View_S}$. This is a key difference between our results and the results of [13]. Note that we can actually implement oblivious transfer relative to our oracle, since although it is hard to sample valid public keys without knowing the corresponding secret key, a party can call $F(pk, \cdot)$ with any string pk and receive a value y indistinguishable from a "valid" image. In contrast, [13] show that oblivious transfer does not exist relative to their oracle. The fact that only S can simulate Eve but not R is the reason that our results do not contradict those of [13].

Remark 3. Note that since the expected number of Eve queries is at most cm/ε, we may consider a modified algorithm Eve' which simulates Eve but aborts if Eve makes more than cm/ε^2 number of queries. By Markov's inequality, this occurs with probability at most $O(\varepsilon)$ and so executions of Eve and Eve' are identical with probability $1 - O(\varepsilon)$. Thus, all properties stated above for Eve hold also for Eve'. In the following, we assume that we run Eve', making at most $N = O(m^{33}) = \mathrm{poly}(n)$ number of queries, to generate the sets $E, Q(E)$. We additionally assume that $N \leq 2^{\hat{n}}/1600m^2$.

4.3 Properties of Fake Openings

Definition 4 (Iterative Indistinguishability). *Let* $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake})$ *be an implementation of a sender-deniable public key encryption scheme relative to oracle* O. *We say that* $\mathcal{E}$ *is* iteratively indistinguishable up to $p(n)$*, where* $p(\cdot)$ *is some polynomial, if for every* i *where* $1 \leq i \leq p(n)$*, and every adversary* $\mathcal{A}$ *making at most a polynomial number of oracle queries we have:*

$$\Pr_{\mathsf{View_S} \sim \mathcal{D}_\mathsf{S}}[\mathcal{A}^{\mathsf{O}}(\mathsf{View_S}) \textit{ outputs } 1] - \Pr_{\mathsf{View}_{\mathsf{S}_i} \sim \mathcal{D}^i_\mathsf{S}}[\mathcal{A}^{\mathsf{O}}(\mathsf{View}_{\mathsf{S}_i}) \textit{ outputs } 1] \leq i/80p(n).$$

In what follows, we split the queries found in a given view $\mathsf{View}_{\mathsf{S}_i}$ into two types: "A" type queries and "B" type queries. Informally, "A" type queries are queries that were also made in the original $\mathsf{View}_{\mathsf{S}_0} = \mathsf{View_S}$. "B" type queries are new queries that were added which do not appear in $\mathsf{View}_{\mathsf{S}_0}$. Details follow.

For a given draw $(\mathsf{O}, \mathsf{View}_{\mathsf{R}}, \mathsf{View}_{\mathsf{S}_0}, \mathsf{View}_{\mathsf{S}_1}, \ldots, \mathsf{View}_{\mathsf{S}_{p(n)}}) \sim \mathcal{D}^{p(n)}_{\mathsf{Fake}}$, we consider a run of the Eve' algorithm with $(\mathsf{O}, \mathsf{View}_{\mathsf{R}}, \mathsf{View}_{\mathsf{S}_0})$ yielding sets $Q(E), Q^{\mathsf{made}}, Q^{\mathsf{skipped}}$ and a run of the Eve' algorithm with $(\mathsf{O}, \mathsf{View}_{\mathsf{R}}, \mathsf{View}_{\mathsf{S}_i})$ for each $1 \leq i \leq p(n)$ yielding sets $Q(E)_i, Q^{\mathsf{made}}_i, Q^{\mathsf{skipped}}_i$.

We define the sets A^0, B^0 corresponding to $(\mathsf{View}_{\mathsf{R}}, \mathsf{View}_{\mathsf{S}_0})$ as follows: $A^0 = Q(\mathsf{S}_0)$, $B^0 = \emptyset$. For $i \geq 1$, we define the sets A^i, B^i corresponding to $(\mathsf{View}_{\mathsf{R}}, \mathsf{View}_{\mathsf{S}_i})$ as follows [8]:

$$A^i = \left(A^{i-1} \setminus Q^{\mathsf{skipped}}_i\right) \cup \left(Q^{\mathsf{made}}_i \cap Q(\mathsf{S}_0)\right),\ B^i = \left(B^{i-1} \setminus Q^{\mathsf{skipped}}_i\right) \cup \left(Q^{\mathsf{made}}_i \setminus Q(\mathsf{S}_0)\right).$$

Note that every draw $(\mathsf{O}, \mathsf{View}_{\mathsf{R}}, \mathsf{View}_{\mathsf{S}_0}, \mathsf{View}_{\mathsf{S}_1}, \ldots, \mathsf{View}_{\mathsf{S}_{p(n)}}) \sim \mathcal{D}^{p(n)}_{\mathsf{Fake}}$, is associated with a unique sequence $(A^0, B^0), (A^1, B^1), \ldots, (A^{p(n)}, B^{p(n)})$.

Definition 5 (Well-formed Sequences). *Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake})$ be an implementation of a sender-deniable public key encryption scheme relative to oracle O. We say that an opening $(\mathsf{O}, \mathsf{View}_{\mathsf{R}}, \mathsf{View}_{\mathsf{S}_0}, \mathsf{View}_{\mathsf{S}_1}, \ldots, \mathsf{View}_{\mathsf{S}_{p(n)}}) \sim \mathcal{D}^{p(n)}_{\mathsf{Fake}}$ is* well-formed *if it has the following properties:*

(1) $(\mathsf{View}_{\mathsf{S}_0}, \mathsf{View}_{\mathsf{S}_1}, \ldots, \mathsf{View}_{\mathsf{S}_{p(n)}})$ *are valid openings.*
(2) $\left(Q(G) \cap \bigcup_{i=1}^{p(n)} Q(\mathsf{S}_i)\right) \setminus Q(E)_G = \emptyset$.
(3) $A^i \subseteq A^{i-1}$ *for* $1 \leq i \leq p(n)$.
(4) *For every query of the form $F(pk, x) = y$ that appears in $Q(E)_i$ for some $1 \leq i \leq p(n)$, the pair (pk, y) does not appear in Q^{skipped}_j for all $1 \leq j \leq i$.*

Claim 2. *Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake})$ be an implementation of a sender-deniable public key encryption scheme relative to oracle O and let $m = m(n)$ be the maximum number of queries made by $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake})$. If $\mathcal{E}$ is iteratively indistinguishable up to $10m^2(n)$ then $(\mathsf{O}, \mathsf{View}_{\mathsf{R}}, \mathsf{View}_{\mathsf{S}_0}, \mathsf{View}_{\mathsf{S}_1}, \ldots, \mathsf{View}_{\mathsf{S}_{10m^2(n)}}) \sim \mathcal{D}^{10m^2(n)}_{\mathsf{Fake}}$ is well-formed with probability $9/10$.*

We defer the proof to the full version [7].

5 Analysis

In this section, we prove our main theorem:

Theorem 3 (Main Theorem, Formal). *Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake})$ be a black-box construction of sender-deniable public key encryption from simulatable PKE and let $m = m(n)$ be the maximum number of queries made by $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake})$. Then $\mathcal{E}$ has security at most $O(m^4)$.*

We first present the following Lemma, which will be our main technical Lemma:

[8] By the notation below, we mean to remove from A^{i-1} all queries of the form $F(pk, x) = y$ such that the pair $(pk, y) \in Q^{\mathsf{skipped}}$. The same holds for the following definitions.

Lemma 3. *Let* $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake})$ *be an implementation of a sender-deniable public key encryption scheme relative to oracle* O *and let* $m = m(n)$ *be the maximum number of queries made by* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake})$. *Then* $\mathcal{E}$ *is* not *iteratively indistinguishable up to* $10m^2 = 10m^2(n)$.

We present the following corollary and use it to prove our main theorem:

Corollary 1. *Let* $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake})$ *be an implementation of a sender-deniable public key encryption scheme relative to oracle* O *and let* $m = m(n)$ *be the maximum number of queries made by* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake})$. *Then there exists an adversary* $\mathcal{A}$ *making a polynomial number of oracle queries such that*

$$\Pr_{\mathsf{View}_{\mathsf{S}} \sim \mathcal{D}}[\mathcal{A}^{\mathsf{O}}(\mathsf{View}_{\mathsf{S}}) \textit{ outputs } 1] - \Pr_{\mathsf{View}_{\mathsf{S}_1} \sim \mathcal{D}^1}[\mathcal{A}^{\mathsf{O}}(\mathsf{View}_{\mathsf{S}_1}) \textit{ outputs } 1] \geq 1/8000m^4.$$

Lemma 1 and Corollary 1 imply our main theorem:

Proof (Proof of Main Theorem using Lemma 1 and Corollary 1.). Assume towards contradiction that there is some fully black-box reduction $(\mathcal{E}, S)$ of sender-deniable public key encryption with distinguishing advantage $o(1/m^4)$ to simulatable PKE, where S is a probabilitic polynomial time reduction. Then, since there exists a construction of simulatable PKE relative to oracle O, we have that $\mathcal{E}$ is also a sender-deniable public key encryption scheme relative to O. Now, Corollary 1 implies that with probability at least $1/16000m^4(n)$ over $\mathsf{O} \sim \Upsilon$, there exists an adversary $\mathcal{A}$ making at most a polynomial number of oracle queries such that $\mathcal{A}$ distinguishes with probability at least $1/16000m^4(n)$. Thus, with probability at least $1/16000m^4(n)$ over $\mathsf{O} \sim \Upsilon$, $\mathcal{A}$ breaks $\mathcal{E}$. However, since S makes at most a polynomial number of calls to $\mathcal{A}$, $S^{\mathcal{A}}$ also makes at most polynomial number of queries and so Lemma 1 implies that with probability $1 - \mathrm{neg}(n)$ over $\mathsf{O} \sim \Upsilon$, $S^{\mathcal{A}}$ does not break $\mathcal{E}_{\mathsf{Sim}}$. Thus, there must exist some fixed O such that $\mathcal{A}$ breaks $\mathcal{E}$ with distinguishing advantage $\Omega(1/m^4)$, but $S^{\mathsf{O},\mathcal{A}}$ does not break $\mathcal{E}_{\mathsf{Sim}}$, which means that the reduction $(\mathcal{E}, S)$ fails and so we arrive at contradiction.

We now turn to proving Lemma 3. We define two events and prove they occur with small probability.

Event E_{rSets}*:* E_{rSets} is the event that a draw $(\mathsf{O}, \mathsf{View}_{\mathsf{S}_0}, \mathsf{View}_{\mathsf{S}_1}, \ldots, \mathsf{View}_{\mathsf{S}_{10m^2(n)}}) \sim \mathcal{D}_{\mathsf{Fake}}^{10m^2(n)}$ has the property that $(A^i, B^i) = (A^{i+1}, B^{i+1})$ for some $0 \leq i \leq 10m^2(n) - 1$.

Event E_{rA}*:* E_{rA} is the event that a draw $(\mathsf{O}, \mathsf{View}_{\mathsf{S}_0}, \mathsf{View}_{\mathsf{S}_1}, \ldots, \mathsf{View}_{\mathsf{S}_{10m^2(n)}}) \sim \mathcal{D}_{\mathsf{Fake}}^{10m^2(n)}$ has the property that for some A^* there are $\beta > 10m(n)$ number of consecutive pairs of the form $(A^*, B^j), \ldots, (A^*, B^{j+\beta-1})$ such that $B^{j+i} \neq B^{j+i+1}$ for $0 \leq i \leq \beta - 2$.

Lemma 4. *Let* $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake})$ *be an implementation of a sender-deniable public key encryption scheme relative to* O *and let* $m = m(n)$ *be the maximum number of queries made by* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake})$. *Let* $\mathcal{E}$ *be iteratively indistinguishable up to* $10m^2(n)$. *The probability that upon a draw* $(\mathsf{O}, \mathsf{View}_{\mathsf{S}_0}, \mathsf{View}_{\mathsf{S}_1}, \ldots, \mathsf{View}_{\mathsf{S}_{10m^2(n)}}) \sim \mathcal{D}_{\mathsf{Fake}}^{10m^2(n)}$ *Event* E_{rSets} *occurs is at most* $1/2$.

Next, we give some intuition for the proof of Lemma 4.

Proof Intuition for Lemma 4. We show that if for two consectutive views $\mathsf{View}_{\mathsf{S}_i}$, $\mathsf{View}_{\mathsf{S}_{i+1}}$, we have that $(A^i, B^i) = (A^{i+1}, B^{i+1})$, then the set of "intersection queries" $Q(E)$ found by the Eve' algorithm when it is run on $\mathsf{View}_{\mathsf{S}_i}$ and $\mathsf{View}_{\mathsf{S}_{i+1}}$ are the same.

Now, intuitively, Lemma 2 tells us that conditioned on the transcript W and intersection queries $Q(E)$, the views of S and R are independent. Since both the transcript (which cannot be changed by the Fake algorithm) and the intersection queries $Q(E)$ are the same for the i-th and $i+1$-th opening, this means that the views of the receiver conditioned on $\mathsf{View}_{\mathsf{S}_i}$ and $\mathsf{View}_{\mathsf{S}_{i+1}}$ should be distributed nearly identically. But note that $\mathsf{View}_{\mathsf{S}_i}$ is supposed to be an encryption of a bit b, while $\mathsf{View}_{\mathsf{S}_i}$ is supposed to be an encryption of the bit $1-b$. Thus, by the correctness of the encryption scheme, the views of the receiver should be statistically far when conditioning on $\mathsf{View}_{\mathsf{S}_i}$ and $\mathsf{View}_{\mathsf{S}_{i+1}}$. This leads to a contradiction.

Lemma 5. *Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake})$ be an implementation of a sender-deniable public key encryption scheme relative to O and let $m = m(n)$ be the maximum number of queries made by $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake})$. Let $\mathcal{E}$ be iteratively indistinguishable up to $10m^2(n)$. The probability that upon a draw $(\mathsf{O}, \mathsf{View}_{\mathsf{S}_0}, \mathsf{View}_{\mathsf{S}_1}, \ldots, \mathsf{View}_{\mathsf{S}_{10m^2(n)}}) \sim \mathcal{D}_{\mathsf{Fake}}^{10m^2(n)}$ Event E_{rA} occurs is at most 1/5.*

Next, we give some intuition for the proof of Lemma 5.

Proof Intuition for Lemma 5. We show that given $\mathsf{View}_{\mathsf{S}_0}$, $\mathsf{View}_{\mathsf{R}}$, oracle O, the set $A^* \subseteq Q(\mathsf{S}_0)$ plus some additional small amount of information we can reconstruct the entire sequence $(A^*, B^j), \ldots, (A^*, B^{j+\beta-1})$. The following is an imprecise description of the reconstruction algorithm:

1. Execute the two-message protocol $\langle \mathsf{S}, \mathsf{R} \rangle$ with Receiver's view $\mathsf{View}_{\mathsf{R}}$ and Sender's view $\mathsf{View}_{\mathsf{S}_0}$.
2. Use the transcript W generated above and begin running the Eve' algorithm to reconstruct set $B^{j+\beta-1}$. The only additional information necessary to reconstruct $B^{j+\beta-1}$ is upon encountering a pair (pk, y) whether to return $F^{-1}(sk, y) = x$ and add the query to $B^{j+\beta-1}$ or whether to add this query to $Q^{\mathsf{skipped}}_{j+\beta-1}$.
3. Continue to construct sets $B^{j+\beta-2}$ through B^j in the same way as above.

The additional information needed to reconstruct $(A^*, B^j), \ldots, (A^*, B^{j+\beta-1})$ can be encoded by a list of α elements. More specifically, when encountering the pair (pk, y) as the ℓ-th query in the run of the Eve' algorithm reconstructing the set B^{j+i}, the algorithm checks whether the index ℓ appears on the list. If it does, the reconstruction algorithm adds $F^{-1}(sk, y) = x$ to B^{j+i}. Otherwise, it adds (pk, y) to $Q^{\mathsf{skipped}}_{j+i}$.

Now, since the Eve' algorithm is efficient and makes N queries (where $N \leq 2^{\hat{n}}/1600m^2$) to reconstruct each B set, we only need $\log N$ bits to encode each of the α elements of the list above. Thus, we need "additional information" of length at most $\alpha \cdot \log N$.

We use properties (2) and (4) of well-formed sequences (see Definition 5) to show that for almost all sequences, when a pair (pk, y) is encountered when running the Eve' algorithm to reconstruct set B^{j+i}, if the corresponding query ($F^{-1}(sk, y)$ or $F(pk, x) = y$) has already been made by the reconstruction algorithm, then (pk, y)

is always added to B^{j+i}. Thus, we do not need to include such pairs in the list at all. This implies that since $B^{j+i} \neq B^{j+i+1}$ for all i, we must have $\alpha \geq \beta$. Moreover, the above implies that at the point when a pair (pk, y) is encountered as the ℓ-th Eve' query and the index ℓ appears on the list then it must be that the corresponding query $F(pk, x) = y$ has not yet been made by the reconstruction algorithm.

This means that at the point where we encounter each of these α queries on the list, the probability that an oracle O chosen *conditioned on the view of the reconstruction algorithm thus far* has the string y in its image is at most $1/2^{\hat{n}}$. Thus, the probability that an O chosen conditioned only on $\mathsf{View}_{\mathsf{S}}, \mathsf{View}_{\mathsf{R}}$ has each of the α-many encountered strings $y_1, \ldots, y_\alpha$ in its image is at most $(1/2^{\hat{n}})^\alpha$.

Finally, taking a union bound over all sets $A^* \subseteq Q(\mathsf{S})$ and all sequences $\mathcal{S}$ we show that the probability that an oracle O chosen conditioned only on $\mathsf{View}_{\mathsf{S}_0}, \mathsf{View}_{\mathsf{R}}$ is consistent with *any* well-formed sequence corresponding to some set $A^* \subseteq Q(\mathsf{S}_0)$ and some and sequence $\mathcal{S}$ of length $\alpha \geq \beta$ is small.

We complete the proof of Lemma 3 using the above lemmas. We defer the proofs of Lemmas 4 and 5 to the full version [7].

Proof (Proof of Lemma 3 using Lemmas 4 and 5). Assume towards contradiction that there is some implementation of a sender-deniable public key encryption scheme, $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, relative to oracle O that is iteratively indistinguishable up to $10m^2 = 10m^2(n)$. By Claim 2, we may assume that, with probability at least $9/10$, a draw $(\mathsf{O}, \mathsf{View}_{\mathsf{R}}, \mathsf{View}_{\mathsf{S}_0}, \mathsf{View}_{\mathsf{S}_1}, \ldots, \mathsf{View}_{\mathsf{S}_{10m^2(n)}}) \sim \mathcal{D}^{10m^2(n)}_{\mathsf{Fake}}$ is well-formed. In particular, this implies that with probability at least $9/10$ over draws, Property (1) and (3) hold so we have that with probability $9/10$ the openings $(\mathsf{View}_{\mathsf{S}_1}, \ldots, \mathsf{View}_{\mathsf{S}_{10m^2(n)}})$ are all valid and $A^{i+1} \subseteq A^i$ for every $0 \leq i \leq 10m^2(n) - 1$. This implies that with probability $9/10$ over draws there must be some set A^* that appears at least $10m = 10m(n)$ times. Moreover, since Lemma 4 guarantees that event E_{rSets} occurs with probability at most $1/2$, we have that with probability at least $9/10 - 1/2 = 2/5$, there is some set A^* that appears at least $10m$ times consecutively and for this A^*, for all $0 \leq i \leq 10m - 2$, $B^{j+i} \neq B^{j+i+1}$. Now, by definition of Event E_{rA}, this means that with probability at least $2/5$ over draws $(\mathsf{O}, \mathsf{View}_{\mathsf{R}}, \mathsf{View}_{\mathsf{S}_0}, \mathsf{View}_{\mathsf{S}_1}, \ldots, \mathsf{View}_{\mathsf{S}_{10m^2(n)}}) \sim \mathcal{D}^{10m^2(n)}_{\mathsf{Fake}}$, we have that Event E_{rA} occurs. But by Lemma 5 we have that event E_{rA} occurs with probability at most $1/5$. Thus, we have arrived at contradiction and so the Lemma is proved.

References

1. Barak, B., Mahmoody-Ghidary, M.: Merkle puzzles are optimal — an $O(n^2)$-query attack on any key exchange from a random oracle. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 374–390. Springer, Heidelberg (2009)
2. Bendlin, R., Nielsen, J.B., Nordholt, P.S., Orlandi, C.: Lower and upper bounds for deniable public-key encryption. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 125–142. Springer, Heidelberg (2011)
3. Canetti, R., Dwork, C., Naor, M., Ostrovsky, R.: Deniable encryption. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 90–104. Springer, Heidelberg (1997)
4. Canetti, R., Feige, U., Goldreich, O., Naor, M.: Adaptively secure multi-party computation. In: STOC, pp. 639–648 (1996)
5. Canetti, R., Gennaro, R.: Incoercible multiparty computation (extended abstract). In: FOCS, pp. 504–513 (1996)

6. Choi, S.G., Dachman-Soled, D., Malkin, T., Wee, H.: Improved non-committing encryption with applications to adaptively secure protocols. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 287–302. Springer, Heidelberg (2009)
7. Dachman-Soled, D.: On the impossibility of sender-deniable public key encryption. IACR Cryptology ePrint Archive, 2012:727 (2012)
8. Dachman-Soled, D., Lindell, Y., Mahmoody, M., Malkin, T.: On the black-box complexity of optimally-fair coin tossing. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 450–467. Springer, Heidelberg (2011)
9. Damgård, I.B., Nielsen, J.B.: Improved non-committing encryption schemes based on a general complexity assumption. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 432–450. Springer, Heidelberg (2000)
10. Dürmuth, M., Freeman, D.M.: Deniable encryption with negligible detection probability: An interactive construction. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 610–626. Springer, Heidelberg (2011)
11. Dürmuth, M., Freeman, D.M.: Deniable encryption with negligible detection probability: An interactive construction. IACR Cryptology ePrint Archive, 2011:66 (2011)
12. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: FOCS, pp. 40–49 (2013)
13. Gertner, Y., Kannan, S., Malkin, T., Reingold, O., Viswanathan, M.: The relationship between public key encryption and oblivious transfer. In: FOCS, pp. 325–335 (2000)
14. Gertner, Y., Malkin, T., Myers, S.: Towards a separation of semantic and CCA security for public key encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 434–455. Springer, Heidelberg (2007)
15. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. J. ACM 33(4), 792–807 (1986)
16. Goldwasser, S., Micali, S.: Probabilistic encryption. J. Comput. Syst. Sci. 28(2), 270–299 (1984)
17. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM J. Comput. 28(4), 1364–1396 (1999)
18. Impagliazzo, R., Luby, M.: One-way functions are essential for complexity based cryptography (extended abstract). In: FOCS, pp. 230–235 (1989)
19. Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: STOC, pp. 44–61 (1989)
20. Luby, M., Rackoff, C.: How to construct pseudorandom permutations from pseudorandom functions. SIAM J. Comput. 17(2), 373–386 (1988)
21. Maji, H.: On Computational Intractability Assumptions in Cryptography. PhD thesis, University of Illinois at Urbana-Champaign, Champaign, Illinois (2011)
22. Naor, M.: Bit commitment using pseudorandomness. J. Cryptology 4(2), 151–158 (1991)
23. Naor, M., Yung, M.: Universal one-way hash functions and their cryptographic applications. In: STOC, pp. 33–43 (1989)
24. O'Neill, A., Peikert, C., Waters, B.: Bi-deniable public-key encryption. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 525–542. Springer, Heidelberg (2011)
25. Reingold, O., Trevisan, L., Vadhan, S.P.: Notions of reducibility between cryptographic primitives. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 1–20. Springer, Heidelberg (2004)
26. Rompel, J.: One-way functions are necessary and sufficient for secure signatures. In: STOC, pp. 387–394 (1990)
27. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: Deniable encryption, and more. IACR Cryptology ePrint Archive, 2013:454 (2013)
28. Yao, A.C.-C.: Theory and applications of trapdoor functions. In: FOCS, pp. 80–91 (1982)