Lower and Upper Bounds for Deniable Public-Key Encryption

Rikke Bendlin[1,⋆], Jesper Buus Nielsen[1,⋆,⋆⋆], Peter Sebastian Nordholt[1,⋆], and Claudio Orlandi[2,⋆⋆⋆]

[1] Aarhus University, Denmark, {rikkeb, jbn, psn}@cs.au.dk
[2] Bar-Ilan University, Israel, claudio.orlandi@biu.ac.il

Abstract. A deniable cryptosystem allows a sender and a receiver to communicate over an insecure channel in such a way that the communication is still secure even if the adversary can threaten the parties into revealing their internal states after the execution of the protocol. This is done by allowing the parties to change their internal state to make it look like a given ciphertext decrypts to a message different from what it really decrypts to. Deniable encryption was in this way introduced to allow to deny a message exchange and hence combat coercion.

Depending on which parties can be coerced, the security level, the flavor and the number of rounds of the cryptosystem, it is possible to define a number of notions of deniable encryption.

In this paper we prove that there does not exist any non-interactive receiver-deniable cryptosystem with better than polynomial security. This also shows that it is impossible to construct a non-interactive bi-deniable public-key encryption scheme with better than polynomial security. Specifically, we give an explicit bound relating the security of the scheme to how efficient the scheme is in terms of key size. Our impossibility result establishes a lower bound on the security.

As a final contribution we give constructions of deniable public-key encryption schemes which establishes upper bounds on the security in terms of key length. There is a gap between our lower and upper bounds, which leaves the interesting open problem of finding the tight bounds.

1 Introduction

Alice and Bob live in a country ruled by an evil dictator, Eve. If Alice wants to communicate with Bob, standard public-key cryptography can be used by Alice if she wants to keep Eve from learning the subject of her communication with Bob. However, if Eve controls the network she will be able to observe that

[⋆] Supported by the Danish National Research Foundation and The National Science Foundation of China (under the grant 61061130540) for the Sino-Danish Center for the Theory of Interactive Computation.

[⋆⋆] Supported by a Sapere Aude grant from The Danish Council for Independent Research.

[⋆⋆⋆] Supported by the European Research Council as part of the ERC project LAST.

a ciphertext is traveling from Alice to Bob. Once the evil Eve knows that a conversation took place, she might get suspicious and force Bob to reveal the content of the conversation. Can cryptography offer any help to Alice and Bob against such a powerful adversary? To solve this problem Canetti, Dwork, Naor and Ostrovsky [CDNO97] introduced the notion of deniable encryption as a tool to combat coercion.

Using a deniable cryptosystem Alice and Bob can communicate over an insecure channel in a way such that even if Eve records the transcript of the communication and later coerces Alice (resp. Bob, or both) to reveal their internal state (secret keys, randomness, ...), then Alice (resp. Bob, or both) has an efficient strategy to produce an alternative internal state that is consistent with the transcript and with a message different than the original one.

Threat model: First note that deniable encryption does not help if Eve has physical access to Alice and Bob's computers. In this case nothing can prevent Eve from seeing everything that Bob sees and therefore learn the encrypted message—since we want Alice and Bob to actually communicate information between them, this is unavoidable. On the other hand, if Alice and Bob can *erase* their secret information, they could simply lie about the content of a ciphertext: the standard indistinguishability security requirement implies that Eve cannot check whether the ciphertext is really an encryption of the message that Alice and Bob claim it to be. Therefore, as in [CDNO97], we consider the case where the parties hand their private keys and randomness to Eve, who can then check that the revealed message is in fact consistent with the ciphertext she observed earlier. If the parties are able to produce a reasonable explanation for the ciphertext that Eve observes, this is enough to fight this kind of coercion.

Sender/Receiver/Bi-Deniability: We distinguish between three kinds of deniability, according to which parties can be coerced by Eve. Note that, up to the number of rounds required by the protocol, sender and receiver deniability are equivalent: Bob can use a sender-deniable scheme to send a random key K to Alice, who can use it to encrypt the message M using a one-time pad and send back $C = M \oplus K$. Now if Bob is coerced he can claim to have received a different message M' by using the sender-deniable property and explain the transcript as if it contained a different K'.

When we consider bi-deniability, the case where Eve can coerce both Alice and Bob, the only coordination that we allow between Alice and Bob is to agree on which message to fake the ciphertext to. In particular this means that the parties cannot communicate to each other their internal states, when they have to produce a fake explanation. This seems to be the only meaningful definition: if Alice and Bob could communicate this information through a channel not controlled by Eve, why would they not use this channel to communicate the original message in the first place?

Fully-Deniable vs. Multi-Distributional: In a multi-distributional deniable cryptosystem a ciphertext produced with a "fake" encryption algorithm $\mathsf{E_F}$ can be later explained as an encryption of any message under the "standard" encryption algorithm E. In other words, for any m, m' it is possible to find appropriate randomness for $\mathsf{E}, \mathsf{E}_\mathsf{F}$ such that $\mathsf{E}(m') = \mathsf{E}_\mathsf{F}(m)$. Note however, that Eve might not believe that the ciphertext was produced using E and ask to see the internal state for E_F and in this case the parties have no efficient strategy to lie about the content of the ciphertext. A fully-deniable scheme is a scheme where $\mathsf{E} = \mathsf{E}_\mathsf{F}$ and therefore does not present this issue.

Public-key vs. Interactive Cryptosystems: A (receiver/sender/bi)-deniable *public-key* cryptosystem is a public-key cryptosystem that is (receiver/sender/bi)-deniable. I.e., the cryptosystem consist of a public key known by the sender and the communication protocol consists of sending a ciphertext to the receiver. A generic, or interactive, cryptosystem might involve arbitrary interaction.

Security Level: All notions of deniability can be quantified by $\varepsilon : \mathbb{N} \to \mathbb{R}_+$ which measures how indistinguishable the faked states are from the honest states. As an example, an ε-receiver-deniable public-key cryptosystem is one in which the faked secret key is ε-indistinguishable from the honest secret key to a computationally bounded distinguisher. We will distinguish between schemes where ε is a negligible function and where ε is of the form $1/p$, for some polynomial p. We will idiosyncratically say that the former kind has negligible security and the latter polynomial security.

Prior Work, Our Contributions and Open Questions: Deniable encryption was first introduced and defined in [CDNO97]. They constructed a sender-deniable public-key cryptosystem with polynomial security, and therefore a receiver-deniable interactive cryptosystem. In [OPW11] O'Neill, Peikert and Waters showed how to construct multi-distributional bi-deniable public-key encryption with negligible security. This is the first scheme that achieves any kind of deniability when both parties are corrupted. Recently, Dürmuth and Freeman announced a fully-deniable (receiver/sender)-deniable interactive cryptosystem with negligible security [DF11]. However their result was later showed to be incorrect by Peikert and Waters.

Our contribution to the state of the art on deniable-encryption is to derive upper and lower bounds on how secure a deniable public-key encryption scheme can be as a function of the key-size.

Lower bounds: As for lower bounds, we have the following results.

- **Receiver:** We show that any public-key cryptosystem with σ-bit keys can be at most $\frac{1}{2}(\sigma + 1)^{-1}$-receiver-deniable.
- **Sender:** We do not know of a non-trivial lower bound for sender-deniable public-key encryption.
- **Bi:** Since bi-deniable public-key encryption with σ-bit keys implies receiver-deniable public-key encryption with σ-bit keys, any public-key cryptosystem with σ-bit keys can be at most $\frac{1}{2}(\sigma + 1)^{-1}$-bi-deniable.

Upper bounds: We show three upper bounds.

Receiver: If we let κ denote the length of the secret key of the best multi-distributional receiver-deniable public-key encryption scheme, then there exists a $1/n$-receiver-deniable public-key encryption scheme with key length $\sigma = O(n^2\kappa)$.

Sender: If we let κ denote the length of the sender randomness in the best multi-distributional sender-deniable public-key encryption scheme, then there exists a $1/n$-sender-deniable public-key encryption scheme where the sender randomness has length $\sigma = O(n\kappa)$.

Bi: If we let κ denote the length of the secret key of the best multi-distributional bi-deniable public-key encryption scheme, then there exists a $1/n$-bi-deniable public-key encryption scheme with key length $\sigma = O(n^4\kappa)$.

We phrase the upper bounds in terms of the upper bounds for multi-distributional schemes. The reason for this is that we do not know of any assumption which allows to construct deniable public-key encryption with polynomial security, which does not also allow to construct multi-distributional deniable encryption. And, we do not know of any direct construction of deniable public-key encryption with polynomial security which is more efficient than going via a multi-distributional scheme. It therefore seems that multi-distributional schemes are the natural building block for deniable public-key encryption with polynomial security.

Our upper bounds for receiver-deniability and sender-deniability are similar to bounds which can be derived from constructions in [OPW11]. Our upper bound for bi-deniability is new. In [OPW11] a construction of a bi-deniable public-key encryption scheme is hinted, but no explicit construction is given which makes it impossible to estimate the complexity. The hinted construction is, however, different from the one we give here.

Our lower bound for receiver-deniability is a generalization of a result in [CDNO97], where a similar bound was proven for any so-called *separable* public-key encryption scheme. An encryption scheme being separable is, however, a very strong structural requirement, so it was unclear if the bound in [CDNO97] should hold for any scheme. In fact, we have not been able to find even a conjecture in the more than a decade of literature between [CDNO97] and the present result that polynomial security should be optimal in general. Our proof technique is completely different from the one in [CDNO97], as we cannot make any structural assumption about the encryption scheme in question.

Our work leaves a number of interesting open problems.

1. Our proof of the upper bounds are via black-box constructions of deniable public-key encryption with polynomial security from multi-distributional deniable public-key encryption. This shows that multi-distributional deniable public-key encryption is stronger than deniable public-key encryption with polynomial security. Is it *strictly* stronger, or does there exist a black-box construction of multi-distributional deniable public-key encryption from deniable public-key encryption with polynomial security?

Notion	Security	Interaction	Sender	Receiver	Bi
Full-Deniability	Negligible	Interactive	?		?
		Public-key	?	✗	✗
	Polynomial	Public-key	✓	✓	✓
Multi-Distributional	Negligible	Public-key	✓	✓	✓

Table 1. The current state of the art for deniable encryption. The first column distinguishes between fully-deniable schemes and schemes with multi-distributional deniability. The Sender/Receiver/Bi columns contains "✓" if any construction is known; a "✗" indicates an impossibility result; a "?" marks a question that is still open.

2. Our lower bounds do not apply to sender-deniable public-key encryption. Is it possible to construct sender-deniable public-key encryption with better than polynomial security?
3. Our lower bounds do not apply to interactive encryption schemes. Is it possible to construct deniable encryption schemes with better than polynomial security when arbitrary interaction is allowed?
4. There is a gap between our upper and lower bounds of at least a factor κ. Since κ itself is typically, for practical purposes, a rather large number (multi-distributional schemes are not simple objects on themselves), this gap is important in practice. What are the tight bounds on the security of a deniable public-key encryption scheme? We conjecture that the bound is in the order of σ^{-1}.

Non-committing encryption: Canetti, Feige, Goldreich and Naor introduced the notion of a non-committing cryptosystem, which is similar to the notion of a bi-deniable cryptosystem, but it is only required that the faking can be done by a simulator. This simulator is allowed to use public keys with a different distribution than those in the protocol. This is needed when showing adaptive security in simulation-based models. It is known [CFGN96] how to implement non-committing encryption with negligible security. Several improvements over the original scheme (both in terms of efficiency and assumptions) have been published in [Bea97,DN00,KO04,GWZ09,CDSMW09].

In [Nie02] it was shown that non-interactive non-committing encryption is impossible. This does not imply the negative result we are proving here, as receiver-deniable public-key encryption does not imply non-committing encryption. In non-committing encryption both sides have to be faked. In receiver-deniable encryption, only the receiver has to be faked. In this sense non-committing encryption is a stronger notion than receiver-deniable encryption. But, in fact, the notions are incomparable, as receiver-deniable encryption on other axes is stronger than non-committing encryption. As an example, it can be shown that if a public-key encryption scheme is receiver-deniable, then the parallel composition of the scheme where the same public key is used to encrypt many massages is also receiver-deniable. This is a property which non-committing encryption provably does not have. And, in fact, this self composition property is crucial in the proof of our lower bound. Also, the result in [Nie02] addresses the case of perfect non-committing encryption (the real-world and the simulated world must be indistinguishable). We are interested in the exact level of security which can be obtained i.e., given a public-key encryption scheme with a certain secret-key length, how deniable can the scheme be?

Structure: In Section 2 we formally define the different flavors of deniable public-key encryption. In Section 3 we show that receiver-deniability is maintained under parallel self-composition with at most a linear security loss. We use that fact to derive our lower bounds giving us the impossibility result of fully-receiver deniable encryption. Finally, section 4 contains our results on poly-deniable encryption schemes.

2 Deniable Public-Key Encryption

In this section we define three different notions of deniable public-key encryption schemes. These notions correspond respectively to an adversary with the ability to coerce the receiver, the sender or both parties simultaneously. We model coercion by letting the adversary request the secret information used in the encryption scheme by the coerceable parties. Deniability is obtained by letting the coerceable parties supply fake secret information.

Basic Scheme. All schemes are defined based on the following definition of a standard public-key encryption scheme consisting of three probabilistic polynomial-time algorithms $(\mathsf{G}, \mathsf{E}, \mathsf{D})$:

- $\mathsf{G}(1^\kappa)$ generates a key-pair (pk, sk), where pk is the public key, sk is the secret key and κ is the security parameter. Note that we consider sk to be the randomness used in $\mathsf{G}(1^\kappa)$.
- $\mathsf{E}_{pk}(m; r)$ generates a ciphertext c which is an encryption under the public key pk of message $m \in \{0,1\}^\ell$ using randomness r. We sometimes write $\mathsf{E}_{pk}(m)$ to make the randomness be implicit.
- $\mathsf{D}_{sk}(c)$ outputs the message $m \in \{0,1\}^\ell$ contained in the ciphertext c.

Let negl : $\mathbb{N} \to \mathbb{R}_+$ be a negligible function. For all notions defined below we require correctness, i.e., we require that $\Pr[\mathsf{D}_{sk}(\mathsf{E}_{pk}(m)) = m] > 1 - \mathrm{negl}(\kappa)$, and IND-CPA security i.e., we require that $\forall$ PPT $(A_1, A_2), \exists\, \mathrm{negl}(\cdot)$:

$$\begin{aligned}\Pr[&(pk, sk) \leftarrow \mathsf{G}(1^\kappa), (m_0, m_1, \mathsf{st}) \leftarrow A_1(pk),\\ &c = \mathsf{E}_{pk}(m_b), b' \leftarrow A_2(c, \mathsf{st}) : b = b'] < 1/2 + \mathrm{negl}(\kappa)\ .\end{aligned}$$

Multi-distributional Encryption. We define a general form of deniable public-key encryption called multi-distributional deniable public-key encryption. Such a scheme essentially consists of two standard public-key schemes sharing a common decryption algorithm.

- The honest scheme $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ does not provide deniability in itself.
- The fakeable scheme $(\mathsf{G}_\mathsf{F}, \mathsf{E}_\mathsf{F}, \mathsf{D})$ provides deniability in the sense that, for a ciphertext c fake secret information can be generated. The faked secret information will make c appear as an encryption of any chosen message m' in the honest scheme. How this is done depends on the notion of deniability as defined below.

For a multi-distributional deniable public-key encryption scheme to be correct we require standard correctness of all public-key schemes $(\mathsf{G}', \mathsf{E}', \mathsf{D})$ where $\mathsf{G}' \in \{\mathsf{G}, \mathsf{G}_\mathsf{F}\}$ and $\mathsf{E}' \in \{\mathsf{E}, \mathsf{E}_\mathsf{F}\}$.

The idea behind having two different schemes is to use the fakeable scheme to encrypt a message m on which the parties would like to have deniability. When coerced the parties simply claim that they used the honest scheme to encrypt the fake message m'. This approach has two disadvantages. First, the parties must decide beforehand whether they later want to deny. Secondly, is the question of why a coercer should believe the parties, when they claim to have used the honest scheme. Note that we cannot guarantee deniability, if the coercer insists on getting the secret information used in the faking process.

Fully-deniable Encryption. An important special case of multi-distributional deniable public-key encryption is fully-deniable public-key encryption (or just deniable public-key encryption). This notion addresses the disadvantages of multi-distributional encryption mentioned above. For a fully-deniable public-key encryption scheme we have that $(\mathsf{G}, \mathsf{E}, \mathsf{D}) = (\mathsf{G}_\mathsf{F}, \mathsf{E}_\mathsf{F}, \mathsf{D})$, that is there are no special faking key generation and encryption algorithms. We will often omit the prefix 'fully' for simplicity.

Receiver-Deniability. A multi-distributional receiver-deniable public-key encryption scheme consists of five probabilistic polynomial-time algorithms $(\mathsf{G}, \mathsf{G}_\mathsf{F}, \mathsf{E}, \mathsf{D}, \mathsf{F}_\mathsf{R})$. Here $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ is the honest scheme and $(\mathsf{G}_\mathsf{F}, \mathsf{E}, \mathsf{D})$ is the fakeable scheme. Notice that the honest and fakeable encryption algorithm are the same since faking is only done on the receiver's side. The faking algorithm F_R is defined as follows:

- For $(pk, sk) \leftarrow \mathsf{G}_\mathsf{F}(1^\kappa)$ and $c \leftarrow \mathsf{E}_{pk}(m)$, $\mathsf{F}_\mathsf{R}(sk, c, m')$ generates an alternative secret key sk' such that $\mathsf{D}_{sk'}(c) = m'$.

Sender-Deniability. A multi-distributional sender-deniable public-key encryption scheme consists of five probabilistic polynomial-time algorithms $(\mathsf{G}, \mathsf{E}, \mathsf{E}_\mathsf{F}, \mathsf{D}, \mathsf{F}_\mathsf{S})$. Here $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ is the honest scheme and $(\mathsf{G}, \mathsf{E}_\mathsf{F}, \mathsf{D})$ is the fakeable scheme. The faking algorithm F_S is defined as follows:

- $\mathsf{F}_\mathsf{S}(pk, m, r, m')$ generates alternative randomness r' such that $\mathsf{E}_{\mathsf{F}pk}(m; r) = \mathsf{E}_{pk}(m'; r')$.

Bi-Deniability. We assume here to be in a setting where receiver and sender have individual faking algorithms. This models the fact that, after an initial stage where the parties can agree on which message to fake to, the sender and the receiver cannot communicate over a channel that is not controlled by the adversary—otherwise they could be using this channel to communicate the message m in the first place.

A multi-distributional bi-deniable public-key encryption scheme consists of seven probabilistic polynomial-time algorithms $(\mathsf{G}, \mathsf{G}_\mathsf{F}, \mathsf{E}, \mathsf{E}_\mathsf{F}, \mathsf{D}, \mathsf{F}_\mathsf{R}, \mathsf{F}_\mathsf{S})$. The faking algorithms F_R and F_S are defined similar to the receiver-deniable and sender-deniable notions respectively, that is:

- For $(pk, sk) \leftarrow \mathsf{G}_\mathsf{F}(1^\kappa)$ and $c \leftarrow \mathsf{E}_{\mathsf{F}pk}(m)$, $\mathsf{F}_\mathsf{R}(sk, c, m')$ generates an alternative secret key sk' such that $\mathsf{D}_{sk'}(c) = m'$.
- $\mathsf{F}_\mathsf{S}(pk, m, r, m')$ generates alternative randomness r' such that $\mathsf{E}_{\mathsf{F}pk}(m; r) = \mathsf{E}_{pk}(m'; r')$.

2.1 Security Notions

The security notions of the three schemes above, are defined in terms of the following experiments performed with an adversary $A = (A_1, A_2)$, where $m, m' \in \{0, 1\}^\ell$.

Honest Game (Receiver)	Faking Game (Receiver)
$(pk, sk) \leftarrow \mathsf{G}(1^\kappa)$	$(pk, sk) \leftarrow \mathsf{G}_\mathsf{F}(1^\kappa)$
$(m, m', \mathsf{st}) \leftarrow A_1(pk)$	$(m, m', \mathsf{st}) \leftarrow A_1(pk)$
$c \leftarrow \mathsf{E}_{pk}(m'; r)$	$c \leftarrow \mathsf{E}_{pk}(m; r)$
	$sk' \leftarrow \mathsf{F}_\mathsf{R}(sk, c, m')$
$b \leftarrow A_2(\mathsf{st}, c, sk)$	$b \leftarrow A_2(\mathsf{st}, c, sk')$

Honest Game (Sender)	Faking Game (Sender)
$(pk, sk) \leftarrow \mathsf{G}(1^\kappa)$	$(pk, sk) \leftarrow \mathsf{G}(1^\kappa)$
$(m, m', \mathsf{st}) \leftarrow A_1(pk)$	$(m, m', \mathsf{st}) \leftarrow A_1(pk)$
$c \leftarrow \mathsf{E}_{pk}(m'; r)$	$c \leftarrow \mathsf{E}_{\mathsf{F}pk}(m; r)$
	$r' \leftarrow \mathsf{F}_\mathsf{S}(pk, m, r, m')$
$b \leftarrow A_2(\mathsf{st}, c, r)$	$b \leftarrow A_2(\mathsf{st}, c, r')$

Honest Game (Bi)	Faking Game (Bi)
$(pk, sk) \leftarrow \mathsf{G}(1^\kappa)$	$(pk, sk) \leftarrow \mathsf{G}_\mathsf{F}(1^\kappa)$
$(m, m', \mathsf{st}) \leftarrow A_1(pk)$	$(m, m', \mathsf{st}) \leftarrow A_1(pk)$
$c \leftarrow \mathsf{E}_{pk}(m'; r)$	$c \leftarrow \mathsf{E}_{\mathsf{F}pk}(m; r)$
	$sk' \leftarrow \mathsf{F}_\mathsf{R}(sk, c, m')$
	$r' \leftarrow \mathsf{F}_\mathsf{S}(pk, m, r, m')$
$b \leftarrow A_2(\mathsf{st}, c, sk, r)$	$b \leftarrow A_2(\mathsf{st}, c, sk', r')$

Let $h_A(\kappa)$ and $f_A(\kappa)$ be the random variables describing b when running the honest game and faking game respectively with security parameter κ. The advantage of A is

$$\mathrm{Adv}_A(\kappa) = |h_A(\kappa) - f_A(\kappa)| \ .$$

We say that a scheme is $\mathsf{(receiver/sender/bi)\text{-}deniable}$ if Adv_A is negligible in κ for any efficient A. Let $\varepsilon : \mathbb{N} \to \mathbb{R}_+$. We say that a scheme is ε-$\mathsf{(receiver/sender/bi)\text{-}deniable}$ if $\mathrm{Adv}_A(\kappa) \leq \varepsilon(\kappa) + \mathrm{negl}(\kappa)$.

2.2 Full Bi-Deniablity implies Full Sender/Receiver-Deniability

Any fully bi-deniable scheme can trivially be turned into both a receiver-deniable and a sender-deniable scheme. On the surface this seems obvious, if both parties can fake then they should be able to fake individually as well. Surprisingly, however, this conclusion cannot be drawn in the multi-distributional setting—in [OPW11] the authors show that in this setting bi-deniability does imply sender deniability but not receiver deniability. As stated in Lemma 1 similar subtleties do not arise in the fully-deniable case. A proof of this can be found in the full version.

Lemma 1. *If* $(\mathsf{G}, \mathsf{E}, \mathsf{D}, \mathsf{F_R}, \mathsf{F_S})$ *is a fully* ε*-bi-deniable encryption scheme, then* $(\mathsf{G}, \mathsf{E}, \mathsf{D}, \mathsf{F_S})$ *is a fully* ε*-sender-deniable encryption scheme and* $(\mathsf{G}, \mathsf{E}, \mathsf{D}, \mathsf{F_R})$ *is a fully* ε*-receiver-deniable encryption scheme.*

3 Impossibility of Fully Receiver/Bi-Deniable Encryption

In this section we prove the impossibility of fully receiver-deniable and fully bi-deniable public-key encryption with better than inverse polynomial security. Since, by Lemma 1, any fully bi-deniable public-key encryption scheme is also a fully receiver-deniable public-key encryption scheme, it is sufficient to prove impossibility of fully receiver-deniable public-key encryption. It turns out that the impossibility follows readily from the fact that full receiver-deniability is preserved under parallel self-composition with only a linear security loss.

We will use a slightly modified definition of receiver-deniability. Recall that in the definition from section 2 the faking algorithm $\mathsf{F_R}$ is invoked as $\mathsf{F_R}(sk, c, m')$, especially it is not given the sender's randomness r. In this section we will allow $\mathsf{F_R}$ to have access to r, that is $\mathsf{F_R}$ is invoked as $\mathsf{F_R}(sk, m, r, m')$. Since we are proving an impossibility result, this does not weaken the result.

3.1 Security of Parallel Self-Composition

Let $(\mathsf{G}, \mathsf{E}, \mathsf{D}, \mathsf{F_R})$ be any receiver-deniable public-key cryptosystem. Let $n : \mathbb{N} \to \mathbb{N}$ be a polynomial in the security parameter κ. We define the parallel self-composition $(\mathsf{G}^n, \mathsf{E}^n, \mathsf{D}^n, \mathsf{F_R}^n)$ as follows:

$$\begin{aligned}
\mathsf{G}^n(1^\kappa) &= \mathsf{G}(1^\kappa) \\
\mathsf{E}^n_{pk}(m_1, \ldots, m_n; r_1, \ldots, r_n) &= (\mathsf{E}_{pk}(m_1; r_1), \ldots, \mathsf{E}_{pk}(m_n; r_n)) \\
\mathsf{D}^n_{sk}(c_1, \ldots, c_n) &= (\mathsf{D}_{sk}(c_1), \ldots, \mathsf{D}_{sk}(c_n)) \\
\mathsf{F_R}^n(sk, (m_1, \ldots, m_n), (r_1, \ldots, r_n), (m'_1, \ldots, m'_n)) &= sk' \ ,
\end{aligned}$$

where $sk_0 = sk$, $sk_i \leftarrow \mathsf{F_R}(sk_{i-1}, m_i, r_i, m'_i)$ for $i = 1, \ldots, n$ and $sk_n = sk'$.

Lemma 2. *If* $(\mathsf{G}, \mathsf{E}, \mathsf{D}, \mathsf{F_R})$ *is* ε*-receiver-deniable, then* $(\mathsf{G}^n, \mathsf{E}^n, \mathsf{D}^n, \mathsf{F_R}^n)$ *is* $n\varepsilon$*-receiver-deniable.*

Proof. Let $\mathrm{A}^n = (\mathrm{A}_1^n, \mathrm{A}_2^n)$ be any probabilistic polynomial-time attacker against $(\mathsf{G}^n, \mathsf{E}^n, \mathsf{D}^n, \mathsf{F_R}^n)$. For $h = 1, \ldots, n$ we construct from A^n a probabilistic polynomial-time attacker $\mathrm{A}_h = (\mathrm{A}_{h,1}, \mathrm{A}_{h,2})$ against $(\mathsf{G}, \mathsf{E}, \mathsf{D}, \mathsf{F_R})$. We can then describe the advantage of A^n in terms of the advantages of A_h for $h = 1, \ldots, n$. Since, by assumption on $(\mathsf{G}, \mathsf{E}, \mathsf{D}, \mathsf{F_R})$, we have a bound on the advantage of each A_h, this gives us the bound on the advantage of A^n. The attacker A_h runs as follows:

1. $\mathrm{A}_{h,1}$: Receives pk.
2. $\mathrm{A}_{h,1}$: Input pk to A_1^n and run A_1^n to obtain $(m_1, \ldots, m_n)$, $(m'_1, \ldots, m'_n)$ and state st_{A^n}.
3. $\mathrm{A}_{h,1}$: For $i = 1, \ldots, h-1$, sample $c_i \leftarrow \mathsf{E}_{pk}(m'_i)$.
4. $\mathrm{A}_{h,1}$: Output $(m_h, m'_h, st_{\mathrm{A}_h})$ where $st_{\mathrm{A}_h} = ((m_1, \ldots, m_n), (m'_1, \ldots, m'_n), st_{\mathrm{A}^n}, (c_1, \ldots, c_{h-1}))$.
5. $\mathrm{A}_{h,2}$: Receive $(st_{\mathrm{A}_h}, c, sk)$. Let $c_h = c$ and $sk_h = sk$.
6. $\mathrm{A}_{h,2}$: For $i = h+1, \ldots, n$, sample $c_i \leftarrow \mathsf{E}_{pk}(m_i; r_i)$ and $sk_i \leftarrow \mathsf{F_R}(sk_{i-1}, m_i, r_i, m'_i)$.
7. $\mathrm{A}_{h,2}$: Input $(st_{\mathrm{A}^n}, (c_1, \ldots, c_n), sk_n)$ to A^n and run it to obtain a bit $b \in \{0, 1\}$.
8. $\mathrm{A}_{h,2}$: Output b.

Let b_h^0 be the distribution of the bit b output by A_h when run in the honest game and let b_h^1 be the distribution of the bit b output by A_h when run in the faking game.

When A_h is run in the honest game, then sk_n is computed from an honest secret key sk_h as $sk_i \leftarrow \mathsf{F_R}(sk_{i-1}, m_i, r_i, m'_i)$ for $i = h+1, \ldots, n$. When A_h is run in the faking game, then sk_n is computed from an honest secret key sk_{h-1} as $sk_i \leftarrow \mathsf{F_R}(sk_{i-1}, m_i, r_i, m'_i)$ for $i = h, \ldots, n$, where the first computation $sk_h \leftarrow \mathsf{F_R}(sk_{h-1}, m_h, r_h, m'_h)$ is performed by the faking game before sk_h is input to A_h. It follows that when A_h is run in the honest game and A_{h+1} is run in the faking game, the values input to A^n have identical distributions, so $b_h^1 = b_{h-1}^0$. Let $\mathrm{Adv}_{\mathrm{A}_h}$ denote the advantage of A_h against $(\mathsf{G}, \mathsf{E}, \mathsf{D}, \mathsf{F_R})$ and $\mathrm{Adv}_{\mathrm{A}^n}$ be the advantage of A^n against $(\mathsf{G}^n, \mathsf{E}^n, \mathsf{D}^n, \mathsf{F_R}^n)$. We then have by definition $\mathrm{Adv}_{\mathrm{A}_h}(\kappa) = |b_h^0 - b_h^1|$ and by construction $\mathrm{Adv}_{\mathrm{A}^n}(\kappa) = |b_n^0 - b_1^1|$, where κ is the security parameter. It then follows using telescoping and the triangle inequality that $\mathrm{Adv}_{\mathrm{A}^n}(\kappa) \leq n\varepsilon(\kappa) + \sum_{h=1}^n \mathrm{negl}_h(\kappa)$, where all negl_h are negligible in κ. The lemma then follows from the fact that the sum of polynomially many negligible functions is negligible. □

Notice that Lemma 2 means that a faked secret key sk_n, resulting from $\mathsf{F_R}^n$, must somehow *remember* the faking of each ciphertext involved in the process. In other words sk_n must not only fake a single ciphertext, it must ensure that every ciphertext c_i decrypts to the faked message m'_i with high probability. To see why consider the efficient adversary A of the receiver-deniable game against $(\mathsf{G}^n, \mathsf{E}^n, \mathsf{D}^n, \mathsf{F_R}^n)$ that simply outputs $b = 1$ if $m'_i = \mathsf{D}_{sk}(c_i)$ for all $i = 1, \dots, n$ and $b = 0$ otherwise. By correctness of the encryption scheme and by Lemma 2 the above property of sk_n becomes clear.

Let s be a bit string of length n. In the proof of the following theorem we use this property to show how to associate each bit of s with a faking of a ciphertext and thus how to store s in the *memory* of the faked secret key sk_n. The impossibility result arises from the fact that this can be done even for random s longer than sk_n.

3.2 Lower Bound

We here show a lower bound on ε in an ε-receiver-deniable encryption scheme. This bound immediately gives that one cannot obtain better than polynomial security. The bound is stated formally in the following theorem:

Theorem 1. *Let* $(\mathsf{G}, \mathsf{E}, \mathsf{D}, \mathsf{F_R})$ *be* ε*-receiver deniable, and let* σ *be an upper bound on the length of the secret keys of* $(\mathsf{G}, \mathsf{E}, \mathsf{D}, \mathsf{F_R})$*, including the faked ones. Then* $\varepsilon \geq \frac{1}{2}(\sigma + 1)^{-1} - \mathrm{negl}(\kappa)$.

Proof. We reach our bound via impossibility of compressing uniformly random data. Let $n = \sigma + 1$. We can assume that $(\mathsf{G}, \mathsf{E}, \mathsf{D}, \mathsf{F_R})$ can encrypt at least one bit, so $(\mathsf{G}^n, \mathsf{E}^n, \mathsf{D}^n, \mathsf{F_R}^n)$ can encrypt n-bit messages. Furthermore $(\mathsf{G}^n, \mathsf{E}^n, \mathsf{D}^n, \mathsf{F_R}^n)$ is $n\varepsilon$-receiver-deniable.

Consider the following communication protocol parametrized by κ. Here is how the sender works:

1. Sample $(pk, sk) \leftarrow \mathsf{G}^n(1^\kappa)$.
2. Sample uniformly random $m' \leftarrow \{0,1\}^n$ and let $m = 0^n$.
3. Sample $c \leftarrow \mathsf{E}^n_{pk}(m; r)$.
4. Let $sk' \leftarrow \mathsf{F_R}^n(sk, m, r, m')$.
5. Send (c, sk').

On receiving (c, sk') the receiver outputs $m'' = \mathsf{D}^n_{sk'}(c)$.

To bound the probability that this protocol fails i.e., that $m'' \neq m'$, consider the following adversary $\mathrm{A} = (\mathrm{A}_1, \mathrm{A}_2)$ for the receiver-deniable security games against $(\mathsf{G}^n, \mathsf{E}^n, \mathsf{D}^n, \mathsf{F_R}^n)$. On input pk A_1 outputs (m, m', st), where the messages m and m' are sampled as in step 2 of the sender algorithm above. The state st is set to be m'. On input (st, c, sk') A_2 computes $\mathsf{D}^n_{sk'}(c) = m''$ and outputs 1 if $m'' = m' = \mathsf{st}$ and 0 otherwise. Now notice that steps 1-4 of the sender algorithm above correspond to the first four steps of the receiver-deniable faking game against A. That is the probability that the communication protocol fails i.e., that $m'' \neq m'$, is exactly the same as A_2 outputting 0 in the faking game. In the honest game we have by correctness of $(\mathsf{G}^n, \mathsf{E}^n, \mathsf{D}^n, \mathsf{F_R}^n)$ that A_2 only outputs 0 with negligible probability. Thus by $n\varepsilon$-receiver deniability we have $\Pr[m'' \neq m'] \leq n\varepsilon(\kappa) + \mathrm{negl}(\kappa)$. We later use this bound on the correctness of the communication protocol to derive our bound, but first we transform the protocol a bit.

For each κ, let r_κ be the value which minimizes the probability that $m'' \neq m'$ when $c_\kappa = \mathsf{E}_{pk}(0^n; r_\kappa)$. Consider then the following non-uniform communication protocol parametrized by κ. Here is how the sender works:

1. Sample $(pk, sk) \leftarrow \mathsf{G}(1^\kappa)$.
2. Sample $m' \leftarrow \{0,1\}^n$.
3. Let $sk' \leftarrow \mathsf{F}_\mathsf{R}(sk, 0^n, r_\kappa, m')$.
4. Send sk'.

The receiver outputs $m'' = \mathsf{D}_{sk'}(c_\kappa)$, where $c_\kappa = \mathsf{E}_{pk}(0; r_\kappa)$. Note that r_κ and c_κ are hardwired into the protocol and is therefore not communicated as part of the protocol. We still have that $\Pr[m'' \neq m'] \leq n\varepsilon(\kappa) + \text{negl}(\kappa)$. Using that $n = \sigma + 1$ we get that $(\sigma+1)\varepsilon(\kappa) \geq 1 - \Pr[m'' = m'] - \text{negl}(\kappa)$. From incompressibility of uniformly random data it follows that $\Pr[m'' = m'] \leq 2^{\sigma - n} = 2^{-1}$, as the protocol sends only sk', which is at most σ bits long and because m' is uniformly random and $n = \sigma + 1$ bits long. Combining these bounds we get that $\varepsilon(\kappa) \geq \frac{1}{2}(\sigma+1)^{-1} - \text{negl}(\kappa)$. □

In words, this bound says that any public-key cryptosystem with σ-bit keys can be be at most $\frac{1}{2}(\sigma+1)^{-1}$-receiver-deniable. Thus to get negligible receiver-deniability keys must be superpolynomial in size. This however would contradict the key generation algorithm being polynomial-time as required by our definition of a public-key cryptosystem.

4 From Multi-Distributional To Poly Deniability

We now give explicit constructions of poly-(sender/receiver/bi)-deniable public-key encryption schemes from any multi-distributional (sender/reciever/bi)-deniable public-key encryption scheme respectively. As in [CDNO97,OPW11], the basic idea in all these constructions is to encrypt a message bit b by first writing it as $b = \bigoplus_{i=1}^n b_i$ for random b_i's, and then encrypting each b_i independently using randomly either the honest or the fakeable encryption scheme. To fake we just have to identify an index j where the fakeable scheme was used and use the corresponding faking algorithm. This is no problem for sender and receiver deniablility since in those cases whoever is running the faking algorithm knows exactly on which indices the fakeable scheme was used. The bi-deniable case however is more challenging because sender and receiver must agree on an index j where they both used the fakeable scheme. As discussed in the introduction, a different solution for this problem was hinted in [OPW11]. All the constructions are for bit encryption: for longer plaintext space one can simply run the scheme in parallel.

In the following subsections we will need two technical lemmas which we state here. Let a *randomized encoding* E be a randomized function from $\{0,1\}$ to $\{0,1\}^n$. Consider the following game $\eth(A, E)$ between a randomized encoding E and an adversary A (an interactive Turing machine):

1. Run A to make it output a bit $b \in \{0,1\}$.

2. Sample $(b_1, \ldots, b_n) \leftarrow E(b)$.
3. Input $(b_1, \ldots, b_n)$ to A and run it to produce a guess $g \in \{0,1\}$.
4. Output g.

We define the *advantage* of A in distinguishing two randomized encodings E_0 and E_1 to be $\mathrm{Adv}_A(E_0, E_1) = |\Pr[\backepsilon(A, E_0) = 0] - \Pr[\backepsilon(A, E_1) = 0]|$. Notice that if we fix b, then $E_0(b)$ and $E_1(b)$ are random variables, making the statistical distance between them well-defined. Let σ_b denote the statistical distance between $E_0(b)$ and $E_1(b)$ and let $\sigma(E_0, E_1) = \max(\sigma_0, \sigma_1)$.

Lemma 3. *It holds for all adversaries A and all randomized encodings E_0 and E_1 that $\mathrm{Adv}_A(E_0, E_1) \leq \sigma(E_0, E_1)$.*

Lemma 4. *Let $s = 1, 2, \ldots$ be a parameter. Let $N : \mathbb{N} \to \mathbb{N}$, where $N_s = N(s)$ is the number of samples at setting s. For each s, let*

$$D_s = \begin{cases} -p & \textit{with probability } q \\ q & \textit{with probability } p \\ 0 & \textit{with probability } 1 - p - q \end{cases} ,$$

where p and q might be functions of s. Let $X_{s,1}, \ldots, X_{s,N_s}$ be N_s i.i.d. variables, distributed according to D_s. Let $X_s = \sum_{i=1}^{N_s} X_{s,i}$ and let $S_s = \Pr[X_s \in [0, \frac{1}{2})]$. Then

$$S_s \leq \frac{1}{\sqrt{pq(p+q)N_s}} \left(\frac{p^2 + q^2}{p + q} + \frac{1}{2\sqrt{2\pi}} \right) .$$

The first lemma is trivial to prove, and the second follows directly from the Berry-Esseen inequality [KS10]. Full proofs can be found in the full version.

4.1 Poly-Sender-Deniability

As a warm up we show that a multi-distributional sender-deniable scheme implies a poly-sender-deniable scheme. From a scheme $(\mathsf{G}, \mathsf{E}, \mathsf{E_F}, \mathsf{D}, \mathsf{F_S})$ we produce a scheme $(\mathsf{G}', \mathsf{E}', \mathsf{D}', \mathsf{F_S}')$ which encrypts a single bit b. The produced scheme is basically the *Parity Scheme* of [CDNO97] only whereas our scheme is based on a multi-distributional sender-deniable scheme, the scheme in [CDNO97] is based on a so-called *translucent set*.

Key Generation $\mathsf{G}'(1^\kappa)$: Output $(pk, sk) \leftarrow \mathsf{G}(1^\kappa)$.

Encryption $\mathsf{E}'_{pk}(b)$: Sample a uniformly random index $j \in \{0, \ldots, n\}$ so that j is even for $b = 0$ and odd for $b = 1$. For $i = 1, \ldots n$ do the following.
   1. For $i \leq j$ sample $c_i \leftarrow \mathsf{E}_{\mathsf{F}pk}(1; r_i)$.
   2. For $i > j$ sample $c_i \leftarrow \mathsf{E}_{pk}(0; r_i)$.

   Output $C = (c_i)_{i=1}^n$.

Decryption $\mathsf{D}'_{sk}(C)$: Parse C as $(c_i)_{i=1}^n$. Compute $b_i = \mathsf{D}_{sk}(c_i)$ for $i = 1, \ldots, n$ and output $b = \bigoplus_{i=1}^n b_i$.

Fake $\mathsf{F_S}'(pk, b, (j, (r_i)_{i=1}^n), b')$: If $b = b'$ output $(j, (r_i)_{i=1}^n)$. Otherwise let $r'_j = \mathsf{F_S}(pk, 1, r_j, 0)$ and $j' = j - 1$. Let all $r'_i = r_i$ for $i \neq j$ and output $(j', (r'_i)_{i=1}^n)$.

Theorem 2. *If* $(\mathsf{G}, \mathsf{E}, \mathsf{E_F}, \mathsf{D}, \mathsf{F_S})$ *is multi-distributional sender-deniable, then* $(\mathsf{G}', \mathsf{E}', \mathsf{D}', \mathsf{F_S}')$ *is* $4/n$*-sender-deniable.*

Proof. Correctness and semantic security is obvious. To prove poly-sender-deniability we first consider the following hybrid game H_1.

H_1 proceeds exactly as the faking game for sender-deniability only it modifies the faking algorithm $\mathsf{F_S}'$ by simply sampling r'_j as randomness for the honest encryption algorithm E, and replaces the ciphertext $C = (c_i)_{i=1}^n$ with $C' = (c'_i)_{i=1}^n$ where $c'_j = \mathsf{E}_{pk}(0; r'_j)$ and $c'_i = c_i$ for all $i \neq j$. Notice that the H_1 only changes the distribution of r'_j and c'_j, the distribution of all other inputs to the adversary remains the same. In other words distinguishing the two games comes down to distinguishing an honest encryption of 0 from an encryption faked to an honest encryption of 0. Thus by the multi-distributional sender-deniability of $(\mathsf{G}, \mathsf{E}, \mathsf{E_F}, \mathsf{D}, \mathsf{F_S})$ the advantage of any adversary in distinguishing the two games will be negligible in κ.

Now consider another hybrid game H_2. H_2 proceeds exactly as the honest game for sender-deniability except that it modifies the encryption algorithm E' by picking j in the following way: first it picks a uniformly random index $i \in \{0, \ldots, n\}$ such that i is odd for $b = 0$ and even for $b = 1$ (i.e., the opposite of how E' picks j) and then sets $j = i-1$. Notice now that H_2 outputs exactly the same as H_1 to the adversary only the output is generated in a slightly different order. I.e., H_1 and H_2 are perfectly indistinguishable. However since H_2 proceeds exactly as the honest game, except that it picks j from a different distribution, distinguishing H_2 from the honest game comes down to distinguishing the two different distributions of j.

In order to utilize Lemma 3 we can view these distributions as randomized encodings. Let us denote by E_0 and E_1 the encodings that encodes a bit b as j 1's followed by $n - j$ 0's. For E_0 j is sampled as in the honest game where the adversary outputs b and for E_1 j is sampled as in the hybrid game H_2 where the adversary outputs b. If $j = -1$ in the hybrid game E_1 will encode this as a special string, say a 0 followed by $n - 1$ 1's. First notice that for $b = 0$ both games sample j uniformly random in $\{0, 2, 4, \ldots, n-1\}$, i.e., $\sigma_0 = 0$. However for $b = 1$ the honest game samples j uniformly random in $\{1, 3, 5, \ldots, n\}$ whereas H_2 samples uniformly random in $\{-1, 1, 3, \ldots, n - 2\}$. Thus clearly $\sigma_1 = 4/n$.

Now by Lemma 3 we have that any adversary has advantage at most $4/n$ in distinguishing the honest game from H_2. By the above hybrid argument it follows that any adversary has advantage at most $4/n + \text{negl}(\kappa)$ in distinguishing the honest game from the faking game. I.e., $(\mathsf{G}, \mathsf{E}, \mathsf{E_F}, \mathsf{D}, \mathsf{F_S})$ is $4/n$-deniable. □

4.2 Poly-Receiver-Deniability

We show that a multi-distributional receiver-deniable scheme implies a poly-receiver-deniable scheme. From a scheme $(\mathsf{G}, \mathsf{G_F}, \mathsf{E}, \mathsf{D}, \mathsf{F_R})$ we produce a scheme $(\mathsf{G}', \mathsf{E}', \mathsf{D}', \mathsf{F_R}')$ which encrypts a single bit b.

Key generation $\mathsf{G}'(1^\kappa)$**:** For $i = 1, \ldots, n$ sample uniformly random bits $a_i \in \{0,1\}$ and then sample $(pk_i, sk_i) \leftarrow \mathsf{G}^{a_i}$, where $\mathsf{G}^0 = \mathsf{G}$ and $\mathsf{G}^1 = \mathsf{G}_\mathsf{F}$. Output $(PK, SK) = ((pk_i)_{i=1}^n, (sk_i, a_i)_{i=1}^n)$.

Encryption $\mathsf{E}'_{PK}(b)$**:** Parse PK as $(pk_i)_{i=1}^n$. For $i = 1, \ldots, n-1$, sample b_i uniformly at random and let $b_n = b \oplus \bigoplus_i^{n-1} b_i$, compute $c_i \leftarrow \mathsf{E}_{pk_i}(b_i)$ and output $C = (c_i)_{i=1}^n$.

Decryption $\mathsf{D}'_{SK}(C)$**:** Parse SK as $(sk_i, a_i)_{i=1}^n$ and C as $(c_i)_{i=1}^n$. Compute $b_i = \mathsf{D}_{sk_i}(c_i)$ for $i = 1, \ldots, n$ and output $b = \bigoplus_{i=1}^n b_i$.

Fake $\mathsf{F_R}'(SK, C, b')$**:** If $b' = \mathsf{D}'_{SK}(C)$ output SK. Otherwise parse SK as $(sk_i, a_i)_{i=1}^n$ and C as $(c_i)_{i=1}^n$. Pick a uniformly random index i for which $a_i = 1$, compute $b_i = \mathsf{D}_{sk_i}(c_i)$ and let $sk_i' = \mathsf{F_R}(sk_i, c_i, 1 - b_i)$ and $a_i' = 0$. For all $j \neq i$, let $sk_j' = sk_j$ and $a_j' = a_j$. Output $SK' = (sk_j', a_j')_{j=1}^n$.

If κ is they key length of the underlying scheme then the above scheme has keys of length $n\kappa$. The following result then implies that one can build a $1/n$-receiver deniable scheme with keys of size $\sigma = O(n^2\kappa)$.

Theorem 3. *If* $(\mathsf{G}, \mathsf{G_F}, \mathsf{E}, \mathsf{D}, \mathsf{F_R})$ *is multi-distributional receiver-deniable, then* $(\mathsf{G}', \mathsf{E}', \mathsf{D}', \mathsf{F_R}')$ *is* $(n-1)^{-1/2}$*-receiver-deniable.*

Proof. In the following we assume for simplicity that n is odd, a similar analysis can be made in the case of n even. Correctness and semantic security is obvious. Using a hybrid argument, the distinguishing probability of any poly-time adversary against the above scheme is negligible close to the best distinguishing advantage between the two randomized encoding E_0 and E_1 defined as follows:

1. $E_0(b) = (b_1, \ldots, b_n)$, where the $b_i \in \{0,1\}$ are uniformly random and independent except that $b = \bigoplus_{i=1}^n b_i$.
2. $E_1(b) = (b_1, \ldots, b_n)$ is sampled as follows. First sample $b_i' \in \{0,1\}$ as in $E_0(b \oplus 1)$. Then, if $\sum_i b_i' = 0$, let $(b_1, \ldots, b_n) = (b_1', \ldots, b_n')$. Otherwise, pick a uniformly random $j \in \{1, \ldots, n\}$ for which $b_j' = 1$ and then let $b_j = 0$ and let $b_i = b_i'$ for $i \neq j$.

The event $\sum_i b_i' = 0$ happens with negligible probability, so we can analyze under the assumption that this does not happen. In that case the bits b_n and b_n' can be computed as $b_n = b \oplus \bigoplus_{i=1}^{n-1} b_i$ respectively $b_n' = b \oplus \bigoplus_{i=1}^{n-1} b_i'$. So, one can distinguish $D_0(b) = (b_1, \ldots, b_{n-1})$ and $D_1(b) = (b_1', \ldots, b_{n-1}')$ with the same advantage as one can distinguish $E_0(b)$ and $E_1(b)$. The distribution $D_0(b)$ consists of $n-1$ uniformly random bits. The distribution $D_1(b)$ consists of $n-1$ uniformly random bits, where we flipped a random occurence of 1 to 0. For $\boldsymbol{b} \in \{0,1\}^{n-1}$, let $\#_1(\boldsymbol{b}) = \sum_{i=1}^{n-1} \boldsymbol{b}_i$ be the number of 1's in the vector and let $\#_0(\boldsymbol{b}) = n - 1 - \#_1(\boldsymbol{b})$ be the number of 0's. By the symmetry of the distributions, it is easy to see that one can distinguish $\#_1(D_0(b))$ and $\#_1(D_1(b))$ with the same advantage as one can distinguish $D_0(b)$ and $D_1(b)$. Since $\#_1(D_0(b))$ is binomially distributed with expectation $\frac{n-1}{2}$ and $\#_1(D_1(b)) = \#_1(D_0(b)) - 1$, it follows that an optimal distinguisher for $\#_1(D_0(b))$ and $\#_1(D_1(b))$ is to guess

0 if $\#_1(D) \geq \frac{n-2}{2}$ and guess 1 otherwise, as this is a maximum likelyhood distinguisher. The advantage of this distinguisher is

$$\begin{aligned}\mathrm{Adv} &= \frac{1}{2}\left|\Pr\left[\#_1(D_0(b)) \geq \frac{n-2}{2}\right] - \Pr\left[\#_1(D_1(b)) \geq \frac{n-2}{2}\right]\right| \\ &= \frac{1}{2}\left|\Pr\left[\#_1(D_0(b)) \geq \frac{n-2}{2}\right] - \Pr\left[\#_1(D_0(b)) \geq \frac{n-2}{2}+1\right]\right| \\ &= \frac{1}{2}\Pr\left[\#_1(D_0(b)) \in \left[\frac{n-2}{2}, \frac{n-2}{2}+1\right)\right] \ .\end{aligned}$$

From $\#_1(D_0(b)) = (n-1) - \#_0(D_0(b))$, we get that $2\#_1(D_0(b)) = \#_1(D_0(b)) + (n-1) - \#_0(D_0(b))$, so $\#_1(D_0(b)) = \frac{n-1}{2} + \frac{1}{2}(\#_1(D_0(b)) - \#_0(D_0(b)))$, and it follows that

$$\begin{aligned}\mathrm{Adv} &= \frac{1}{2}\Pr\left[\frac{1}{2}(\#_1(D_0(b)) - \#_0(D_0(b))) \in \left[-\frac{1}{2}, \frac{1}{2}\right)\right] \\ &= \frac{1}{2}\Pr\left[\frac{1}{2}\#_1(D_0(b)) - \frac{1}{2}\#_0(D_0(b)) \in \left[0, \frac{1}{2}\right)\right] \ .\end{aligned}$$

The last equality follows from n being odd. Consider then Lemma 4, with $p = q = \frac{1}{2}$ and $N_s = s-1$. The variable X_s in the premise then has exactly the same distribution as $\frac{1}{2}\#_1(D_0(b)) - \frac{1}{2}\#_0(D_0(b))$ when $s = n$. Plugging $p = q = \frac{1}{2}$ and $N_s = n-1$ into Lemma 4 we get that $\Pr[\frac{1}{2}\#_1(D_0(b)) - \frac{1}{2}\#_0(D_0(b)) \in [0, \frac{1}{2})] \leq \frac{2}{\sqrt{s-1}}$. □

4.3 Poly-Bi-Deniability

We show that a multi-distributional bi-deniable scheme implies a poly-bi-deniable scheme. From a scheme $(\mathsf{G}, \mathsf{G_F}, \mathsf{E}, \mathsf{E_F}, \mathsf{D}, \mathsf{F_S}, \mathsf{F_R})$ we produce a scheme $(\mathsf{G}', \mathsf{E}', \mathsf{D}', \mathsf{F_S}', \mathsf{F_R}')$ which encrypts a single bit.

Key generation $\mathsf{G}'(1^\kappa)$: For $i = 1, \ldots, n^2$ sample random bits $a_i \in \{0,1\}$ and then sample $(pk_i, sk_i) \leftarrow \mathsf{G}^{a_i}(1^\kappa)$, where $\mathsf{G}^0 = \mathsf{G}$ and $\mathsf{G}^1 = \mathsf{G_F}$. Sample the a_i's independently with $\Pr[a_i = 0] = 1/n$. Output $(PK, SK) = ((pk_i)_{i=1}^{n^2}, (sk_i, a_i)_{i=1}^{n^2})$.

Encryption $\mathsf{E}'_{PK}(b)$: Parse PK as $(pk_i)_{i=1}^{n^2}$. For $i = 1, \ldots, n^2$

1. Sample uniformly random $b_i \in_{\mathrm{R}} \{0,1\}$ and $m_i \in_{\mathrm{R}} \{0,1\}^\kappa$ such that $b = \bigoplus_{i=1}^{n^2} b_i$.
2. Compute $c_i \leftarrow \mathsf{E}_{pk_i}^{b_i}(m_i', r_i)$, where $m_i' = b_i m_i$ ($0m_i = 0^\kappa$ and $1m_i = m_i$), $\mathsf{E}^0 = \mathsf{E}$ and $\mathsf{E}^1 = \mathsf{E_F}$.

Output $C = (c_i)_{i=1}^{n^2}$.

Decryption $\mathsf{D}'_{SK}(C)$: Parse SK as $(sk_i, a_i)_{i=1}^{n^2}$ and C as $(c_i)_{i=1}^{n^2}$. For $i = 1, \ldots, n^2$, compute $m_i' = \mathsf{D}_{sk}(c_i)$ and let $b_i' = 1$ if $m_i' \neq 0$ and $b_i' = 0$ if $m_i' = 0$. Output $b = \bigoplus_{i=1}^{n^2} b_i'$.

Fake (sender) $\mathsf{F_S}'(PK, b, (r_i, m_i, b_i)_{i=1}^{n^2}, b')$**:** If $b = b'$ output $(r_i, m_i, b_i)_{i=1}^{n^2}$. Otherwise parse PK as $(pk_i)_{i=1}^{n^2}$. Let $m' = \min\{m'_i = b_i m_i | i \in \{1, \ldots, n^2\} \wedge m'_i \neq 0^\kappa\}$ and pick the unique (ewnp.) index k for which $m'_k = b_k m_k = m'$ (notice this implies $b_k = 1$). I.e., k is the index of the c_i containing the smallest non-zero plaintext. The minimum is taken according to lexicographic order. Then let $r'_k = \mathsf{F_S}(pk_k, m'_k, r_k, 0^\kappa)$, $m'_k = m_k$ and $b'_k = 0$. For all $j \neq k$, let $r'_j = r_j$, $m'_j = m_j$ and $b'_j = b_j$. Output $(r_j, m_j, b_j)_{j=1}^{n^2}$.

Fake (receiver) $\mathsf{F_R}'(SK, C, b')$**:** If $\mathsf{D}'_{SK}(C) = b'$ output SK. Otherwise parse SK as $(sk_i, a_i)_{i=1}^{n^2}$ and C as $(c_i)_{i=1}^{n^2}$ and compute $m'_i = \mathsf{D}_{sk}(c_i)$. Let $m' = \min\{m'_i | i \in \{1, \ldots, n^2\} \wedge m'_i \neq 0^\kappa\}$ and pick the unique (ewnp.) index k for which $m'_k = m'$. I.e., k is the index of the c_i containing the smallest non-zero plaintext. The minimum is taken according to lexicographic order. If $a_k = 0$, then give up. If $a_k = 1$, then let $sk'_k = \mathsf{F_R}(sk_k, c_k, 0^\kappa)$ and $a'_k = 0$. For all $j \neq k$, let $sk'_j = sk_j$ and $a'_j = a_j$. Output $SK' = (sk'_j, a'_j)_{j=1}^{n^2}$.

Theorem 4. *If* $(\mathsf{G}, \mathsf{G_F}, \mathsf{E}, \mathsf{E_F}, \mathsf{D}, \mathsf{F_S}, \mathsf{F_R})$ *is multi-distributional bi-deniable, then* $(\mathsf{G}', \mathsf{E}', \mathsf{D}', \mathsf{F_S}', \mathsf{F_R}')$ *is* $O(n^{-1/2})$*-bi-deniable.*

Proof. Correctness follows by observing that $b'_i = b_i$ unless one of the uniformly random κ-bit messages m_i happens to be 0^κ, which is a negligible event. Semantic security is obvious. As for bi-deniability, by a hybrid argument similar to that in the proofs of Thm. 2 and Thm. 3, distinguishing the honest and faking game comes down to distinguishing the following two random encodings of a bit b.

1. $E_0(b) = (b_1, \ldots, b_{n^2}, a_1, \ldots, a_{n^2})$, where the $b_i \in \{0, 1\}$ are sampled uniformly at random except that $\bigoplus_{i=1}^{n^2} b_i = b$ and the $a_i \in \{0, 1\}$ are sampled such that $\Pr[a_i = 0] = 1/n$.
2. $E_1(b) = (b_1, \ldots, b_{n^2}, a_1, \ldots, a_{n^2})$ is sampled as follows. First sample $b'_i, a'_i \in \{0, 1\}$ as in $E_0(b \oplus 1)$. Then, if $\sum_i b'_i = 0$, let $(b_1, \ldots, b_{n^2}) = (b'_1, \ldots, b'_{n^2})$. Otherwise, pick a uniformly random $k \in \{1, \ldots, n^2\}$ for which $b'_k = 1$ and then let $b_k = 0$ and let $b_k = b'_k$ for $i \neq k$. If $a'_k = 1$ let $a_k = 0$ and let $a_i = a'_i$ for $i \neq k$.

It happens that $a'_k = 0$ with probability $1/n$, so by adding $1/n$ to the bound in the end, we can analyse under the assumption that $a'_k = 1$. In that case we can describe $E_1(b)$ as above, except that we pick k uniformly at random among the i's for which $b'_i = 1$ and $a'_i = 1$. Then we set $b_k = 0$ and $a_k = 0$ and set $b_{i \neq k} = b'_i$ and $a_{i \neq k} = a'_i$.

Given a vector $\boldsymbol{v} = (b_1, \ldots, b_{n^2}, a_1, \ldots, a_{n^2})$, we let $\#_{00}(\boldsymbol{v})$ be the number of i's for which $b_i = a_i = 0$ and we let $\#_{11}(\boldsymbol{v})$ be the number of i's for which $b_i = a_i = 1$. For simplicity we assume that b is uniformly random, such that $b_1, \ldots, b_{n^2}$ is uniform in $\{0, 1\}^{n^2}$. Deriving the same bound for fixed $b = 0$ and $b = 1$ is straight-forward. Let $p = \frac{1}{2n}$ be the probability that $a_i = 0$ and $b_i = 0$. Let $q = \frac{n-1}{2n}$ be the probability that $a_i = 1$ and $b_i = 1$. The expected value of $\#_{00}(E_0(b))$ is pn^2. The expected value of $\#_{11}(E_0(b))$ is qn^2, and $\#_{00}(E_1(b)) = \#_{00}(E_0(b)) + 1$ and $\#_{11}(E_1(b)) = \#_{11}(E_0(b)) - 1$. From this it can be derived as in the proof of Thm. 3 that the maximum likelihood distinguisher for $E_0(b)$ and $E_1(b)$ guesses 0 if $q\#_{00} - p\#_{11} > 0$ and that its advantage is $\frac{1}{2}\Pr[q\#_{00}(E_0(b)) - p\#_{11}(E_1(b)) \in [0, \frac{1}{2})]$. Using Lemma 4 as in the proof of Thm. 3, with $s = n$, $N_s = s^2$ and the p and q defined above, it follows that

$$\Pr[q\#_{00}(E_0(b)) - p\#_{11}(E_1(b)) \in [0, \frac{1}{2})] \leq \frac{1}{\sqrt{s}}\left(\sqrt{2} + \frac{1}{\sqrt{\pi}}\right) .$$

The theorem then follows from $\sqrt{2} + \frac{1}{\sqrt{\pi}} \leq 2$. □

Acknowledgements: the authors would like to thank Adam O'Neill, Chris Peikert and Brent Waters for sharing with us an early copy of [OPW11].

References

[Bea97] Donald Beaver. Plug and play encryption. In *CRYPTO*, pages 75–89, 1997.

[CDNO97] Ran Canetti, Cynthia Dwork, Moni Naor, and Rafail Ostrovsky. Deniable encryption. In *CRYPTO*, pages 90–104, 1997.

[CDSMW09] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Improved non-committing encryption with applications to adaptively secure protocols. In *ASIACRYPT*, pages 287–302, 2009.

[CFGN96] Ran Canetti, Uriel Feige, Oded Goldreich, and Moni Naor. Adaptively secure multi-party computation. In *STOC*, pages 639–648, 1996.

[DF11] Markus Dürmuth and David Mandell Freeman. Deniable encryption with negligible detection probability: An interactive construction. In *EUROCRYPT*, pages 610–626, 2011.

[DN00] Ivan Damgård and Jesper Buus Nielsen. Improved non-committing encryption schemes based on a general complexity assumption. In *CRYPTO*, pages 432–450, 2000.

[GWZ09] Juan A. Garay, Daniel Wichs, and Hong-Sheng Zhou. Somewhat non-committing encryption and efficient adaptively secure oblivious transfer. In *CRYPTO*, pages 505–523, 2009.

[KO04] Jonathan Katz and Rafail Ostrovsky. Round-optimal secure two-party computation. In *CRYPTO*, pages 335–354, 2004.

[KS10] V. Yu. Korolev and I. G. Shevtsova. On the upper bound for the absolute constant in the Berry–Esseen inequality. *Theory of Probability and its Applications*, 54(4):638–658, 2010.

[Nie02] Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In *CRYPTO*, pages 111–126, 2002.

[OPW11] Adam O'Neill, Chris Peikert, and Brent Waters. Bi-deniable public-key encryption. In *CRYPTO*, pages 525–542, 2011.