Efficient Coercion Resistant Public Key Encryption

Maged Hamada Ibrahim *mhii72@gmail.com*
Department of Electronics, Communications and Computers Engineering,
Faculty of Engineering, Helwan University
1, Sherif St., Helwan, Cairo; P.O. 11792, Egypt

Abstract

The notion of deniable encryption has been known in the literature since its introduction in [1] as coercion resistant encryption schemes that allow the user (sender and/or receiver) to escape a coercion attempted by a coercive adversary. The schemes allow the user to open fake message(s) to the coercer that when verified gives the same ciphertext as the true message, while the receiver is always able to decrypt for the true message. In this paper we focus on sender-incoercible encryption. The contribution of this paper is two-fold. First, we introduce a new classification of services that could be provided by coercion-resistant encryption showing that all previously proposed deniable PKE schemes fall in the category of unplanned incoercible PKE assuming the user is non-collaborative and do not satisfy the requirements for deniable encryption. Then we inspect, refine and improve the sender-incoercible PKE introduced in [2]. Our new scheme achieves constant transmission rate where the size of the plaintext may be calibrated to be sufficiently large i.e. the scheme encrypts arbitrary length messages without a blowup expansion in the ciphertext while the size of the ciphertext grows linearly with the number of fake messages.

Keywords: Network Security, Coercion-resistance, Deniable Encryption, Incoercible Encryption, Receipt-freeness, Hybrid Encryption, Electronic Voting.

1. INTRODUCTION

Consider the scenario of standard public key encryption where the receiver $\mathcal{R}$'s public key is known to everyone and could be used by arbitrary many senders (potentially unknown to the receiver at the time the public key is published). For an arbitrary sender $\mathcal{S}$ to send a message m to $\mathcal{R}$ without any further interaction, he computes and sends the ciphertext c which is essentially a function of $\mathcal{R}$'s public key pk and the message m. On the reception of c, $\mathcal{R}$ is able to decrypt for m using the corresponding secret key sk. Now, assume that there is an adversary that is able to eavesdrop on the channel between $\mathcal{S}$ and $\mathcal{R}$, of course, the one-way security property of the encryption scheme prevents a polynomial time adversary from reaching the plaintext given the ciphertext in hand without knowing the secret key. However, the one-way security of the encryption function is not enough since such a weak notion does not hide the statistical properties of the plaintext message (unless the message is truly random parameter with sufficiently large bit-length). Therefore, the semantic security notion (probabilistic encryption) or equivalently, indistinguishable chosen plaintext attack (IND-CPA notion) which emphasizes on the necessity of hiding the plaintext statistical properties using random coins has been introduced in [3]. Other active attacks that could be attempted by an adversary (such as tampering with the ciphertext in a way undetectable by the receiver) have been introduced (through several notions of security) in [4, 5, 6, 7] and many others. In all these notions, incorporating random coins are essential in performing the encryption securely.

1.1 Coercion

Now, consider the situation where the adversary $\mathcal{A}$ (in addition of being able to eavesdrop on the channel between $\mathcal{S}$ and $\mathcal{R}$) has some coercive power against the sender $\mathcal{S}$. Such a *coercive adversary* cannot corrupt $\mathcal{S}$ (i.e. cannot replace $\mathcal{S}$) and hence she is weaker than a corruptive

adversary, yet she only has some coercive power against $\mathcal{S}$ which allows her to force $\mathcal{S}$ to perform as she wishes. This adversary approaches $\mathcal{S}$ commanding him to perform some actions (e.g. sending a particular vote to $\mathcal{R}$) against $\mathcal{S}$'s desire. Notice that $\mathcal{A}$ is able to record all communications sent from $\mathcal{S}$ to $\mathcal{R}$. Now, the following question arises: Does a standard public key encryption scheme allows $\mathcal{S}$ to escape such coercion? In standard PKE, $\mathcal{S}$ computes the ciphertext on the form $c = \mathcal{E}_{pk}(r, m)$ where pk is $\mathcal{R}$'s public key, r is $\mathcal{S}$'s random coins and m is the message that $\mathcal{S}$ wishes to send. Now $\mathcal{A}$ approaches $\mathcal{S}$ after transmission and asks him to reveal r and m (in practice, $\mathcal{A}$ asks $\mathcal{S}$ to deliver his laptop, hard-drive or even the whole server to her). Let m' be the message satisfactory to $\mathcal{A}$. Notice that c commits $\mathcal{S}$ to both r and m i.e. $\mathcal{S}$ is not able to come up (in polynomial time) with a different random coins r' satisfying $\mathcal{E}_{pk}(r, m) = \mathcal{E}_{pk}(r', m')$. Also, $\mathcal{S}$ cannot claim that he erased his memory (specially the random coins) since in this case, $\mathcal{A}$ assumes that $\mathcal{S}$ didn't obey her desire and takes actions against him. Therefore, if $\mathcal{S}$ is to lie safely without being suspected as a liar, the encryption scheme must allow him to open the encryption without the need to claim the erasure of any of his local randomness that are known to exist.

1.2 An Overview of Existing Adversary Models

For seeking completeness, in this subsection we review the essential models of an adversary: There are several models of a corruptive adversary: *Stationary (non-mobile) adversary*, *Mobile adversary*, *Static adversary* and *Adaptive adversary*. In a stationary adversary, the adversary may attack a number of parties (minority), and this number is assumed not to exceed a certain value (the threshold) along the life time of the private inputs.

A mobile adversary [8, 9] is able to jump from one party to the other (mobile virus attacks), collecting as much information as she can, she has the whole life-time of a secret to do so. Hence, the assumption that the adversary will not exceed a certain threshold no more holds. To withstand such type of an adversary, the parties must pro-actively renew their private inputs (cooperatively) through proactive security techniques and erase any previously shared information.

In a static adversary [10], the parties that the adversary is to corrupt are defined prior to the multiparty protocol execution, and remains unchanged during execution, that is, the adversary does not adapt her behavior during execution of the protocol whenever (for example) she finds that some party did not erase previous information after pro-actively renew her private inputs.

An adaptive adversary is a stronger adversary [10, 11]. This adversary is not only able to jump from one party to another, but she do that in a wise manner, according to her view of the communications among parties and her view of the computations of the already corrupted parties. Withstanding such type of adversaries is not an easy task especially in the existence of dishonest parties (non-erasing parties that are not trusted to erase their sensitive information).

The type of an adversary that we deal with in this paper cannot corrupt a party, yet, she has a coercive power that allows her to coerce a party to do as she wishes. This type of an adversary is known as a *coercive adversary* [1, 2, 12, 13]. The notion of *deniable* encryption deals with this type of an adversary in the sense that it allows a party to open any plaintext message that when verified gives the same ciphertext observed by this adversary.

1.3 Deniability and Adaptive Security

Adaptively secure encryption represents the tool (plug and play) to achieve adaptive security in multiparty protocols. This tool is a.k.a *non-committing* encryption. However, the term non-committing is misleading because such encryption is indeed committing as notified in [14]. This encryption is committing in the sense that an honest sender cannot later pretend that an alternate message was sent. That is, these cryptosystems are non-committing in the existence of the simulator (the ideal world, not the real world). A distinguish between deniable encryption [1, 2, 12] and non-committing (adaptively secure) encryption [14, 15, 16, 17, 18, 19] is as follows. Deniable

encryption is a true non-committing encryption which faces a type of an adversary known as a coercive adversary. This adversary is weaker than a corruptive adversary in the sense that, she cannot corrupt a party, yet, she has some power that allows her to coerce this party to do as she wishes. In deniable encryption, a sender can generate a ciphertext that appears as an encryption of many messages. Whereas, in non-committing encryption, the ciphertext that could be opened as an encryption of any message is generated by the simulator (in the ideal world) while the honest parties are still committed to the encryption. The simulator has many advantages over the sending and the receiving parties, for example, the simulator knows the public as well as the secret keys of all parties and all auxiliary information used to perform the encryption which makes non-committing encryption easier than coercion-resistant encryption. Yet, for a non-committing encryption scheme to be useful (i.e. achieves adaptive security in a multiparty environment), it must withstand corruption of the sender and the receiver simultaneously which is a strong assumption, whereas, coercion-resistant encryption could be designed under weaker assumptions (e.g. to achieve sender-only deniability or receiver-only deniability) where one of the two parties is beyond the reach of the adversary. However, when it comes to the design of a sender-and-receiver coercion-resistant encryption, it is much harder than non-committing encryption. In this paper we focus on the sender side and assume that the receiver is beyond coercion.

2. PREVIOUS WORK

The notion of deniable encryption has been introduced in [1] where a sender-incoercible public-key encryption scheme based on trapdoor permutations has been constructed. However, the scheme falls short of achieving an appropriate level of deniability, due to several reasons: i) To achieve a high level of incoercibility, the size of the ciphertext corresponding to a one bit encryption is super-polynomial and hence inefficient. ii) There is a significant probability of decryption errors. iii) A collaborative sender is able to setup the encryption in a way that allows him to later prove the decrypted message to the coercer.

The recent work in [2] introduced a scheme for a one-move sender-incoercible public-key encryption which is built using any trapdoor permutation. The scheme also allows the sender to lie safely but still the true message is provable by the sender. The communication complexity is in the order of $\mathcal{O}(k)$ bits for one bit plaintext, where k is a security parameter. To encrypt more than one bit, the scheme deviates from practicality as also notified in [20]. The same communication complexity applies to the public-key encryption scheme in [12].

3. MOTIVATIONS AND CONTRIBUTIONS

3.1 Motivations

We find that the notion of coercion-resistant encryption schemes (previously known as deniable encryption) needs to be refined from the point of view of classification of services provided, in the sense that – for example – the encryption scheme to be coercion-resistant, it is not necessarily be deniable, it is enough to allow the user to lie safely without claiming the erasure of his random inputs. The idea introduced in [2] is good as to provide an incoercible PKE scheme (not deniable PKE), however the construction was inefficient leading to high computations and communications complexity making the scheme deviates from practice as the plaintext bit-length increases [20].

3.2 Our Contributions

First, we introduce a new classification of services that could be provided by coercion-resistant encryption showing that all previously proposed schemes fall in the category of unplanned incoercible encryption and do not satisfy the requirements for deniable encryption. Then we inspect, refine and improve the sender-incoercible PKE introduced in [2]. Our new scheme achieves constant transmission rate where the size of the plaintext may be calibrated to be sufficiently large i.e. the scheme encrypts arbitrary length messages without blowup expansion in the size of the ciphertext. Also, the size of the ciphertext grows linearly with the number of fake messages.

4. COERCION-RESISTANT ENCRYPTION: DENIABLE vs. INCOERCIBLE

In this section we introduce the new classifications of services we want to escalate for coercion resistant encryption schemes. Consider – for example – the Yes/No e-voting scheme where the sender is supposed to submit his Yes or No vote against the coercer in an encrypted way to the authority. We have two cases here according to the time of coercion, either before or after transmission. In case the coercer approaches the sender before transmission, then the coercer has the chance to perform the encryption and generates the ciphertext by himself (since he knows the receiver's public-key), delivers the ciphertext to the sender forcing him to send this particular ciphertext (notice that the coercer is able to eavesdrop on the channel), consequently, the sender is trapped and has nothing to do but sending this ciphertext to the receiver. Approaching the sender before transmission traps the sender regardless of the type of encryption scheme used.

Now we are left with the case where the coercer approaches the sender after transmission that is the first contact between them is after the ciphertext has been placed on the channel. If the sender's claim that he erased all or part of his random inputs is accepted to a coercer then any standard PKE is indeed incoercible but not deniable since still the sender has the choice to open all his random inputs to the coercer (as a receipt) and prove the encrypted plaintext. However, the claim of the sender that he erased all or part of his random inputs will be taken against him. Here comes the difference we want to escalate between deniability and incoercibility: *Incoercible encryption is not receipt-free but allows the sender to lie safely without claiming the erasure of any of his random inputs while Deniable encryption is receipt-free i.e. disables the ability of the sender to prove to the coercer the plaintext that will be decrypted by the receiver.*

According to the instant of coercion, we have one of the following two situations:

1. *Unplanned coercion*: In this type we assume that there is absolutely no contact between the sender and the coercer before transmission of the ciphertext. The first contact between them is after the ciphertext has been transmitted and probably recorded by the coercer.

2. *Planned coercion*: In this type, the coercer may have one or more contacts with the sender before transmission which allows arrangements for the encryption and ciphertext transmission process.

In planned coercion, since the coercer may contact the sender before transmission, the coercer may deliver the sender the ciphertext he wants to see on the channel (i.e. the ciphertext is setup by the coercer himself), consequently, the sender is trapped and has nothing to do but obeying the coercer's order. For this reason, we have strong feeling that, theoretically, planned coercion is impossible to realize in practice, however, on the proof of this statement we make no claim, because some technical assumptions may open the door for realizing such a scheme.

We make another categorization according to the sender's will to cooperate with the coercer as follows:

1. *Collaborative-sender*: In this type, the sender is willing to cooperate with the coercer (e.g. knowing which message satisfactory to the coercer) so that, the sender is able to setup the encryption in a way that allows him later to prove the encrypted message to the coercer (i.e. the message that will be decrypted by the receiver) to benefit from such transmission (e.g. gaining some cash in return).

2. *Non-collaborative-sender*: The sender wants to perform the encryption and setup the ciphertext in a way that allows him later to *lie safely* to the coercer about the encrypted message without being proved as a liar.

In Collaborative-sender, notice that, the sender is willing to perform actions against his own beliefs, as long as such action satisfies the coercer. However, to do so, the sender needs to know exactly what message satisfies the coercer before transmission takes place so that he will be able to setup the encryption in a way that allows him later to prove to the coercer that this particular message will be decrypted by the receiver. For the sender to be able to do that (i.e. knowing what satisfies the coercer), there must be some sort of contact between him and the coercer prior to transmission. Hence, we conclude that *collaborative-sender is equivalent to planned coercion* which, again, is almost impossible to withstand in practice, since the sender can ask the coercer for the ciphertext that he wants to see on the channel.

In Non-collaborative sender, the sender is not willing to collaborate with the coercer, he just wants to perform actions satisfying his own beliefs and at the same time, is able to lie safely about his encrypted message (without claiming that he erased any of his local parameters such as random coins) to escape a coercive actions (e.g. violent actions) that could be attempted by the coercer. Again, we must emphasize that although the sender is non-collaborative, if the coercer approaches him before transmission, then the coercer will force him to send a particular ciphertext and the sender will not be able to lie in this case. Almost all previously proposed sender-Incoercible encryption and also the scheme introduced in this paper are unplanned Incoercible, that is, the coercer approaches the sender after transmission. *We want to emphasize that our sender-incoercible PKE (SI-PKE) scheme assumes absolutely no contact of any type between the coercer and the sender prior to the transmission of the ciphertext*. Under this assumption, after transmission, whenever the coercer approaches the sender, our SI-PKE allows the sender to open any message satisfactory to the coercer, and safely claims this fake message as the one that will be decrypted by $\mathcal{R}$ without the need to claim the erasure of any of his local randomness (e.g. random coins). In this case, the coercer is unable to prove that the sender's claim is false, and hence, Incoercibility holds.

New terminology distinction: In the literature, and up to this point, deniability and incoercibility were used alternatively for the same meaning. Here we make a new distinction between *Deniability* and *Incoercibility*. Our proposed scheme is in fact sender-incoercible but not sender-deniable. We introduce the following classification of coercion resistant encryption:

• *Incoercible encryption*: In this class of coercion resistant encryption, the sender still able to prove at least one of the encrypted messages to the coercer if he wants to, i.e., he can perform the encryption and setup the ciphertext in a way that allows him to prove to the coercer the message that will be decrypted by the receiver. Since the encryption in this case is still provable, we bring incoercibility as a new terminology for the encryption scheme that is still provable yet allows the sender to lie whenever he desires without claiming the erasure of any of his local randomness.

• *Deniable encryption*: In this class of coercion resistant encryption, the sender who sets up the ciphertext cannot prove (even to himself) the encrypted message. We notify that no PKE exists that satisfies this property in the non-erasure model. The only well-known fully deniable encryption scheme is the one-time-pad.

Figure 1 summarizes our new terminology distinction and categorization of services provided by coercion resistant schemes. It shows (by dark blocks) which services our SI-PKE provides. By inspecting previous schemes such as [1, 2] and those surveyed and reviewed in [20], we found that none of these encryption schemes are deniable in the sense mentioned above, since the sender is always able to prove to the coercer the plaintext that will be decrypted by the receiver. However, in the encryption schemes of [1, 2] the sender is able to lie safely to the coercer and hence, the schemes are incoercible.

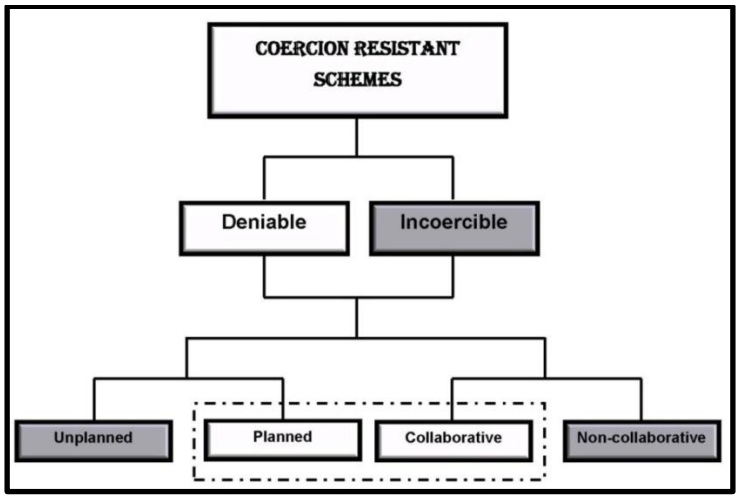

FIGURE 1: Our View of the Services Provided by Coercion Resistant Schemes.

5. EXISTING COERCION RESISTANT SCHEMES

In this section we review several schemes and protocols to declare the distinction between incoercibility and deniability. First we review the SI-PKE scheme of [2] and show why this scheme is incoercible but not deniable. Next, we discuss the one-time-pad as a perfectly deniable encryption scheme and also discuss the perfectly secure message transmission (PSMT) techniques as a method to achieve deniability under assumptions that are too strong to be realized in practice.

5.1 Sender-Incoercible PKE

In this section we review the SI-PKE introduced in [2]. The scheme could be built given any trapdoor permutation (f, f^{-1}), where, f is the receiver's public function and f^{-1} is its trapdoor inverse known only to the receiver. By $(b_{k-1}^{(x)} \dots b_0^{(x)})$ we denote the binary representation of $x \in \{0,1\}^k$. Let $f^{(j)}(y_0) = f(f(\dots f(f(y_0))\dots))$ be the process of encrypting y_0 (i.e. applying f to y_0) j times where $y_0 \in \mathcal{D}_f$. Let d be the maximum number of decryptions that will be performed by the receiver (i.e. the receiver will apply f^{-1} no more than d times). Let $H: \{0,1\}^* \to \{0,1\}^\ell$ be a hash function with digest (output) bit-length ℓ. The pair (f, d) is the receiver's public key. Let b_t be the true bit to be encrypted while b_f be the fake bit. The scheme is described next.

Encryption: To encrypt one bit, the sender proceeds as follows:

- *Honest encryption* $(b_t = b_f)$.
  - Picks y_0 at random from the domain of f such that, $b_t = \bigoplus_{i=0}^{k-1} b_i^{(y_0)}$.
  - Picks a small integer $0 < r < d$.
  - Computes $C = f^{(r)}(y_0)$.
  - Picks $e \in_R \{0,1\}$, sets $R_e = H(y_0)$ and $R_{1-e} \in_R \{0,1\}^\ell$.
  - Sends (C, R_0, R_1) to the receiver.

- *Dishonest encryption* $(b_t = \bar{b}_f)$.
  - Picks y_0 at random from the domain of f such that, $b_t = \bigoplus_{i=0}^{k-1} b_i^{(y_0)}$.
  - Keeps on applying f to y_0, that is, to compute $y_j = f^{(j)}(y_0), (j = 1,2,\dots) < d$ until there exists some y_j with its binary representation satisfying, $b_f = \bigoplus_{i=0}^{k-1} b_i^{(y_j)}$.
  - Applies f at least one more time to compute $C = f^{(r)}(y_0), j < r \leq d$.
  - Picks $e \in_R \{0,1\}$, sets $R_e = H(y_0)$ and $R_{1-e} = H(y_j)$.
  - Sends (C, R_0, R_1) to the receiver.

Decryption: On the reception of (C, R_0, R_1), $\mathcal{R}$ starts decrypting by computing $y_{i-1} = f^{-1}(y_i)$ for $i = d \cdots 1$, arranges the resulting parameters as the tuple $\mathcal{Y} = \langle y_{j_1}, \cdots, y_{j_d} \rangle$, he inspects $\mathcal{Y}$ for the least index j_i such that $H(y_{j_i})$ matches either R_0 or R_1. In this case (when the match is found) then $y_{j_i} = y_0$. Finally, $\mathcal{R}$ uses y_0 to decrypt for $b_t = \bigoplus_{i=0}^{k-1} b_i^{(y_0)}$.

Opening the encryption: In order to open the encryption honestly, the sender reveals y_0. To open dishonestly, the sender reveals y_j , claims that y_j is picked at random from the domain of f and that R_e is random.

Incoercibility. In the dishonest encryption, after transmission, when the coercer approaches $\mathcal{S}$, $\mathcal{S}$ lies safely by opening y_j. It is infeasible for the coercer to perform decryptions without the knowledge of f^{-1} and hence cannot reach y_0. In this case, from the properties of the hash function, the claim of $\mathcal{R}$ that R_e is random cannot be proven false by the coercer.

Undeniability. It is obvious that if $\mathcal{S}$ opens honestly (i.e. y_0) then he proves to the coercer that y_0 and hence b_t is the bit decrypted by $\mathcal{R}$ since in this case the coercer is able to reach y_j and knows that both R_0 and R_1 are not random and that y_0 is indeed with the smallest index, and consequently the scheme satisfies sender-incoercibility but not deniability

5.2 Perfectly Deniable Schemes

In the following we describe the one-time-pad and the PSMT as perfectly deniable schemes, yet, their assumptions are impractical.

One-time-pad. In a one-time-pad, the sender $\mathcal{S}$ and the receiver $\mathcal{R}$ are sharing a secret parameter k which is used only once. $\mathcal{S}$ prepares his message m where $|k| = |m|$ and sends the ciphertext $c = k \oplus m$. The receiver $\mathcal{R}$ decrypts by computing $m = k \oplus c$. Although one-time-pad encryption is impractical, we describe it here as the only perfectly deniable encryption scheme known. Given any ciphertext c, $\mathcal{S}$ or $\mathcal{R}$ may open the encryption of a fake message $m_f \neq m_t$ and $|m_f| = |m_t|$ as the pair (k_f, m_f) where $k_f = c \oplus m_f$. For the one-time-pad to be sender-and-receiver deniable encryption, both $\mathcal{S}$ and $\mathcal{R}$ must coordinate their stories for m_f. It is obvious that, neither $\mathcal{S}$ nor $\mathcal{R}$ are able to prove the encryption even to themselves. Given any ciphertext c', then pick a pair (k', m') such that $|c'| = |k'| = |m'|$ and $c' = k' \oplus m'$.

Perfectly secure message transmission. In *perfectly secure message transmission* (PSMT) first introduced in [21], and improved in subsequent contributions (e.g. [22, 23, 24, 25]), a sender $\mathcal{S}$ and a receiver $\mathcal{R}$ are connected by $n = 3t + 1$ channels with at most t channels are corrupted by the adversary, while the remaining $n - t$ channels are beyond the reach of the adversary (physically secured). Under these assumptions (in one round of communication) $\mathcal{S}$ is able to transmit a message m to $\mathcal{R}$ in a perfectly secure way using polynomial sharing. Finally, $\mathcal{R}$ decodes for the message using the well-known BerleKamp-Welch decoder [26]. In three rounds of communication (assuming that $\mathcal{S}$ is the party that always start the communication) connectivity could be improved to $n = 2t + 1$. Such perfect secure transmission is indeed non-committing and hence, deniable and adaptively secure, since it could be easily shown that the parties will always be able to fake a conversation as long as at least $n - t$ channels are beyond the reach of the adversary. Under the assumption that physically secure channels exist between every pair of nodes in the network, coercion resistant encryption is useless. Yet, the assumption that some channels are physically secured is impractical in most network applications (e.g. the internet and wireless connectivity) and hence, standard encryption techniques are employed to protect from an adversary with eavesdropping capacities that extend to the whole network. In this case, privacy is preserved (in the cryptographic sense) and correctness is achieved by assuming that the adversary corrupts a fraction of the network paths. However, in addition to the fact that the system becomes cryptographically secure, employing standard encryption techniques gives rise to adaptive and coercive vulnerabilities.

6. OUR IDEA AND BASIC TOOLS

Since we focus in this paper on sender incoercibility, we mention what a sender-incoercible encryption scheme is required to satisfy as follows:

- The sender must be able to open any information the coercer asks for. On the other hand, the coercer will not ask for something that does not exist or that cannot be proven to exist.
- The information opened by the sender must be consistent (or appear to be consistent) with the transmitted ciphertext.
- The sender must not claim the erasure of any of his local randomness, since such claim will not be accepted by the coercer as long as such local randomness is proven to exist (e.g. random coins in conventional PKEs).

We remark that $\mathcal{S}$ may claim that some computed parameters are picked at random as long as the cryptographic assumption prevents the coercer from detecting such lie. For example, given a hash function H, $\mathcal{S}$ may pick x and compute $y = H(x)$ and claims to the coercer that y is picked at random. Here, from the one wayness of H the claim of $\mathcal{S}$ cannot be proven false by the coercer and hence the coercer finds no reason to ask for any x since in this case he cannot prove the existence of this x.

The idea to improve the complexity of our previously proposed scheme in [2] is to make use of hybrid encryption. In hybrid encryption the public key of the receiver is used to encrypt a symmetric key for a symmetric encryption scheme. The symmetric key is used to encrypt an arbitrary length plaintext. In our scheme, the public key encryption algorithm is used in its OW security (as a one way trapdoor permutation) to encrypt several symmetric keys (we call this process "folded encryption"). The symmetric keys are used to encrypt the true and the fake messages using any available symmetric key encryption algorithm. This allows the encryption of arbitrary length messages (unlike the schemes in [2]). On the other hand, with the help of a strong hash function, the receiver will be able to reach the symmetric key intended to encrypt the true message and bypass all other fake keys/messages.

6.1 Asymmetric Encryption

An asymmetric encryption scheme [e.g. 3, 27, 28, 29], $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D}, \mathrm{COINS}, \mathrm{MSPC})$ is a triple of algorithms, associated with finite sets, $\mathrm{COINS}(k)$ and $\mathrm{MSPC}(k) \subseteq \{0,1\}^*$, for $k \in \mathbb{N}$, where:

•The algorithm $\mathcal{K}$, called the key-generation algorithm, is a probabilistic algorithm which on input $1^k \in \mathbb{N}$ outputs a pair of strings $(pk, sk) \leftarrow \mathcal{K}(1^k)$.

•The algorithm $\mathcal{E}$, called the encryption algorithm, is a probabilistic algorithm that takes a pair of strings, pk and x, and a string $r \leftarrow \mathrm{COINS}(k)$, and produces a string $y = \mathcal{E}_{pk}(x; r)$.

•The algorithm $\mathcal{D}$, called the decryption algorithm, is a deterministic algorithm that takes a pair of strings, sk and y, and returns a string $x = D_{sk}(y)$.

It is required that, for any $k \in \mathbb{N}$, if $(pk, sk) \leftarrow \mathcal{K}(1^k)$, $x \in \mathrm{MSPC}$, and $y \leftarrow \mathcal{E}_{pk}(x)$, then $\mathcal{D}_{sk}(y) = x$. In our proposed scheme, we need an asymmetric encryption algorithm which is one-way secure, hence we need the weakest security notion of asymmetric encryption defined next.

One-way Encryption: Let $\mathcal{AE}$ be an asymmetric encryption. For $\mathcal{AE}$, we consider an algorithm, $\mathcal{A}$, called an adversary, that, taking a public-key, pk, outputted by $\mathcal{K}$, and an encryption, y, of a random plaintext in MSPC tries to decrypt y. The probability of $\mathcal{A}$'s success, denoted by the advantage of $\mathcal{A}$, depends on $\mathcal{A}$, $\mathcal{AE}$, and the random choice of a plaintext from MSPC. $\mathcal{A}$ does not have any decryption oracle (while an encryption oracle doesn't matter because chosen-

plaintext attacks are clearly unavoidable in an asymmetric encryption scheme). For $k \in \mathbb{N}$, define the advantage of $\mathcal{A}$ by

$$Adv^{OW}_{\mathcal{A},\mathcal{AE},MSPC}(k) = Pr[(pk, sk) \leftarrow \mathcal{K}(1^k); x \leftarrow MSPC(k); y \leftarrow \mathcal{E}_{pk}(x): \mathcal{A}(pk, y) = \mathcal{D}_{sk}(y)]$$

It is said that an adversary $\mathcal{A}$ (t, ε)-breaks $\mathcal{AE}$ in the sense of OW'ness if $\mathcal{A}$ runs in at most time t and achieves $Adv^{OW}_{\mathcal{A},\mathcal{AE},MSPC}(k) \geq \varepsilon$. It is said that $\mathcal{AE}$ is (t, ε)-secure in the sense of OW'ness if there is no adversary that (t, ε)-breaks $\mathcal{AE}$ in that sense. A OW-secure $\mathcal{AE}$ is spoken of as "one-way trapdoor permutation".

6.2 Symmetric Encryption

A symmetric encryption scheme, $\mathcal{SK} = (E, D, \mathrm{KSPC}, \mathrm{MSPC})$ is a pair of algorithms associated with finite sets, $\mathrm{KSPC}(k)$ and $\mathrm{MSPC}(k)$, $\subseteq \{0,1\}^*$, for $k \in \mathbb{N}$, where:

• E, called the encryption algorithm, is a deterministic algorithm that takes a pair of strings, K and x, and produces $y = E_K(x)$.

• D, called the decryption algorithm, is a deterministic algorithm that takes a pair of strings, K and y, and outputs a string $x = D_K(y)$.

It is required that, for any $k \in \mathbb{N}$, if $K \in \mathrm{KSPC}(k)$, $x \in \mathrm{MSPC}$ and $y = E_K(x)$, then $D_K(y) = x$.

6.3 Important Notations for Our Scheme

The following notations are important to clearly understand our SI-PKE scheme:

- We denote by $y_j = \mathcal{E}^{(j)}_{pk}(y_0)$ the process of encrypting $y_0 \in \mathrm{MSPC}$ j-times $(j \geq 1)$ that is, $y_j = \mathcal{E}^{(j)}_{pk}(y_0) = \mathcal{E}_{pk}(\mathcal{E}_{pk}(\cdots \mathcal{E}_{pk}(y_0)))$
- We denote by $y_{j-i} = \mathcal{D}^{(i)}_{sk}(y_j)$ the process of decrypting y_j i-times $(1 \leq i \leq j)$ that is $y_{j-i} = \mathcal{D}^{(i)}_{sk}(y_j) = \mathcal{D}_{sk}(\mathcal{D}_{sk}(\cdots \mathcal{D}_{sk}(y_j)))$. Notice that $y_0 = \mathcal{D}^{(j)}_{sk}(y_j)$.

7. OUR SI-PKE SCHEME

We assume that the receiver $\mathcal{R}$ who is beyond coercion has a public/private key pair (pk, sk) for a OW-secure asymmetric encryption scheme $\mathcal{AE}$ (eg. RSA one-way trapdoor permutation). Let $H: \{0,1\}^* \rightarrow \{0,1\}^\ell$ be a strong hash function [30] with digest (output) bit-length ℓ. Let $(d > 2) \in \mathbb{N}$ be a small integer picked by $\mathcal{R}$. The pair (pk, d) is known to everyone including the sender $\mathcal{S}$ as $\mathcal{R}$'s public-key for the SI-PKE while sk is kept private to $\mathcal{R}$.

7.1 SI-PKE for One Fake Message

The sender $\mathcal{S}$ has two arbitrary length messages $m_t \in \mathrm{MSPC}^{\mathrm{sym}}$ and $m_f \in \mathrm{MSPC}^{\mathrm{sym}}$ where m_t is his true message aimed to be decrypted by $\mathcal{R}$ while m_f is the fake message that he may wish to open to a coercer later after transmission. Of course, we must have $|m_t| = |m_f|$. The SI-PKE scheme operates as follows:

Encryption: $\mathcal{S}$ proceeds as follows:

- *Honest Encryption* $(m_t = m_f)$:
  - Picks r at random such that $1 \leq r \leq d$.
  - Picks $K_0 \in_R \mathrm{MSPC}^{\mathrm{asym}}$.
  - Computes $K_r = \mathcal{E}^{(r)}_{pk}(K_0)$.
  - Picks $e \in_R \{0,1\}$.
  - Sets $(R_e, C_e) = (H(K_0), E_{K_0}(m_t))$ while $R_{1-e} \in_R \{0,1\}^\ell$ and $C_{1-e} \in_R \mathrm{MSPC}^{\mathrm{sym}}$.
  - Sends the tuple $\mathcal{C} = \langle K_r, (R_0, C_0), (R_1, C_1) \rangle$.

- *Dishonest Encryption* ($m_t \neq m_f$):
  - Picks r at random such that $1 \leq r \leq d$.
  - Picks $K_0 \in_R \text{MSPC}^{\text{asym}}$.
  - Computes $K_r = \mathcal{E}_{pk}^{(r)}(K_0)$.
  - Picks any K_j, $(0 < j < r)$.
  - Picks $e \in_R \{0,1\}$.
  - Sets $(R_e, C_e) = (H(K_0), E_{K_0}(m_t))$ and $(R_{1-e}, C_{1-e}) = (H(K_j), E_{K_j}(m_f))$.
  - Sends the tuple $\mathcal{C} = \langle K_r, (R_0, C_0), (R_1, C_1)\rangle$.

Decryption: On the reception of $\mathcal{C}$, $\mathcal{R}$ starts decrypting by computing $K_{i-1} = D_{sk}(K_i)$ for $i = d \cdots 1$, arranges the resulting keys as the tuple $\mathcal{K} = \langle K_{j_1}, \cdots, K_{j_d}\rangle$, he inspects $\mathcal{K}$ for the least index j_i such that $H(K_{j_i})$ matches either R_0 or R_1. In this case, when the match is found, then $K_{j_i} = K_0$. Finally, $\mathcal{R}$ uses K_0 to decrypt for $m_t = D_{K_0}(C_e)$.

Opening the encryption: In order to open the encryption honestly, the sender reveals K_0. To open dishonestly, the sender reveals K_j , claims that K_j is picked at random from $\text{MSPC}^{\text{asym}}$ and that C_e and R_e are random.

Incoercibility. In the dishonest encryption, after transmission, when the coercer approaches $\mathcal{S}$, $\mathcal{S}$ lies safely by opening K_j. Since from the OW security of the asymmetric encryption scheme and assuming $\mathcal{R}$ is beyond coercion/corruption, it is infeasible for the coercer to perform decryptions without the knowledge of sk and hence cannot reach K_0. In this case, from the properties of the hash function, the claim of $\mathcal{R}$ that R_e is random cannot be proven false by the coercer. From the chosen ciphertext/plaintext security of the symmetric encryption scheme, even if the coercer knows m_t and given the corresponding C_e, he cannot prove that C_e is not random without the knowledge of the encryption key K_0. Hence, by using our SI-PKE, $\mathcal{S}$ can safely lie to the coercer without being caught.

We remark that opening K_0 allows the coercer to reach K_j by performing $K_j = \mathcal{E}_{pk}^{(j)}(K_0)$, therefore the coercer will easily detect that C_{1-e} and R_{1-e} are not random. This does not threaten incoercibility since the claim of $\mathcal{S}$ that K_0 is the key with the smallest index is true. We emphasize that when $\mathcal{S}$ opens K_j and claims that K_j is the key with the smallest index (i.e. K_0), the coercer given pk and the ciphertext cannot reach any K_i with $i < j$.

7.2 SI-PKE for Multiple Fake Messages

We simply extend our SI-PKE for one fake message to the case where the sender needs to prepare for more than one fake message. Here, the complexity will grow linearly with the number of fake messages. Let $m_t \in \text{MSPC}^{\text{sym}}$ be the true message while $m_f^{(1)}, \cdots, m_f^{(n)} \in \text{MSPC}^{\text{sym}}$ be the $n \geq 1$ possible fake messages that the sender may want to open any of them later to the coercer. We have $|m_t| = |m_f^{(i)}|, \forall i$ and the number of fake messages (n) must be fixed and does not change from one encryption to another. It is required that $d > n$ and this could be easily satisfied by allowing $\mathcal{R}$ to choose d alittle bit larger than the case for one fake message encryption. Here, we may call n and d, *the faking capacity* of the SI-PKE scheme. The SI-PKE scheme is as described next.

Encryption: $\mathcal{S}$ proceeds as follows:

- *Honest encryption*:
  - Picks $K_0 \in_R \text{MSPC}^{\text{asym}}$.
  - Picks a random r such that $n \leq r \leq d$.
  - Computes $K_r = \mathcal{E}_{pk}^{(r)}(K_0)$.
  - Picks $e \in_R \{0, \cdots, n\}$.

- Sets $(R_e, C_e) = (H(K_0), E_{K_0}(m_t))$ while $R_{v\neq e} \in_R \{0,1\}^{\ell}$ and $C_{v\neq e} \in_R \text{MSPC}^{\text{sym}}$ for $v = 0 \cdots n$.
- Sends the tuple, $\mathcal{C} = \langle K_r, (R_0, C_0), \cdots, (R_n, C_n)\rangle$ to $\mathcal{R}$.

• *Dishonest Encryption* ($m_t \neq m_f^{(1)} \neq \cdots \neq m_f^{(n)}$):
- Picks $K_0 \in_R MSPC^{\text{asym}}$.
- Picks a random r such that $n \leq r \leq d$.
- Computes $K_r = \mathcal{E}_{pk}^{(r)}(K_0)$.
- Picks any K_j's, $(0 < j < r)$ and arranges them as the tuple $\langle K_{j_1}, \cdots, K_{j_n}\rangle$.
- Picks $e \in_R \{0, \cdots, n\}$.
-Sets $(R_e, C_e) = (H(K_0), E_{K_0}(m_t))$ and assigns each other $(R_{v\neq e}, C_{v\neq e})$ a value $\left(H\left(K_{j_{v\neq 0}}\right), E_{K_{j_{v\neq 0}}}\left(m_f^{(v\neq 0)}\right)\right), v = 0, \cdots, n$.
- Sends the tuple, $\mathcal{C} = \langle K_r, (R_0, C_0), \cdots, (R_n, C_n)\rangle$ to $\mathcal{R}$.

Decryption: On the reception of $\mathcal{C}$, $\mathcal{R}$ starts decrypting by computing $K_{i-1} = D_{sk}(K_i)$ for $i = d \cdots 1$, arranges the resulting keys as the tuple $\mathcal{K} = \langle K_{j_1}, \cdots, K_{j_d}\rangle$, he inspects $\mathcal{K}$ for the least index j_i such that $H(K_{j_i})$ matches any of the R_v's for $v = 0 \cdots n$. In this case, when the match is found, then $K_{j_i} = K_0$ and $R_v = R_e$. Finally, $\mathcal{R}$ uses K_0 to decrypt for $m_t = D_{K_0}(C_e)$.

Opening the encryption: In order to open the encryption honestly, the sender reveals K_0. To open dishonestly, the sender opens the fake message he wants (say $m_f^{(v)}$) and reveals the corresponding key K_{j_v} claiming that K_{j_v} is picked at random from $\text{MSPC}^{\text{asym}}$ (i.e. $K_{j_v} = K_0$).

Incoercibility. The sender $\mathcal{S}$ picks any fake message $m_f^{(v)} \in \{m_f^{(1)}, \cdots, m_f^{(n)}\}$ and lies safely by opening the corresponding key K_{j_v} used to encrypt this message. Since from the OW security of the asymmetric encryption scheme and assuming $\mathcal{R}$ is beyond coercion/corruption, it is infeasible for the coercer to perform decryptions without the knowledge of sk and hence cannot reach K_0 or any key $K_{j_1}, \cdots, K_{j_{v-1}}$. In this case, from the properties of the hash function, $\mathcal{S}$'s claim that $R_0, \cdots, R_{v-1}$ are random cannot be proven false by the coercer. From the chosen ciphertext/plaintext security of the symmetric encryption scheme, even if the coercer knows m_t and all fake messages and given $C_0, \cdots, C_n$, he cannot prove that $C_0, \cdots, C_{v-1}$ are not random without the knowledge of the encryption keys $K_0, K_{j_1} \cdots, K_{j_{v-1}}$. Hence, by using our SI-PKE, $\mathcal{S}$ is able to safely lie to the coercer without being caught. Notice that the coercer given K_{j_v} is able to find all keys $K_{j_{v+1}}, \cdots, K_{j_r}$ and hence he knows that $R_{v+1}, \ldots, R_n$ and $C_{v+1}, \ldots, C_n$ are not random. We emphasize that the main claim of $\mathcal{S}$ is that the opened key K_{j_v} is the key with the smallest index (i.e. $j_v = 0$) which still cannot be proven false by the coercer and therefore incoercibility still holds.

Finally, from previous discussions in this paper, it is clear by inspection that our SI-PKE scheme is unplanned incoercible assuming the sender is non-collaborative and the receiver is beyond coercion, yet, the scheme is undeniable.

8. CONCLUSIONS

In this paper we introduced a new classification of services in the area of coercion resistant encryption showing that coercion resistant encryption schemes are classified as either incoercible encryption or deniable encryption where: incoercible encryption allows the sender to lie without claiming the erasure of any of his random inputs but still committed to the encryption and is able (under his own choice) to prove to the coercer the message that will be decrypted by the receiver, while in deniable encryption, the sender cannot prove even to himself the decrypted message. We also showed (heuristically) that planned coercion-resistant encryption, where the coercer is assumed to approach the sender before transmission is impossible to realize in practice. Then we proposed an improvement to our previously proposed sender incoercible encryption to achieve constant transmission rate where the size of the plaintext may be calibrated to be sufficiently

large. Unlike previous schemes where the size of the ciphertext is super polynomial, our scheme encrypts arbitrary length messages without a blowup expansion in the ciphertext while the size of the ciphertext grows linearly with the number of fake messages.

9. REFERENCES

[1] R. Canetti, C. Dwork, M. Naor, R. Ostrovsky, “Deniable Encryption,” in CRYPTO’97, 1997, pp.90-104.

[2] M. H. Ibrahim, “A Method for Obtaining Deniable Public-Key Encryption,” international journal of network security (ijns), Vol. 8, No. 1, 2009, pp. 1–9.

[3] S. Goldwasser, S. Micali, “Probabilistic Encryption,” Journal Computer System Science, 28(2), 1984, pp. 270–299.

[4] M. Naor, M. Yung, “Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks,” in proceedings of STOC 1990, pp. 427–437.

[5] C. Rackoff, D. R. Simon, “Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack,” CRYPTO 1991, pp. 433–444.

[6] D. Dolev, C. Dwork, M. Naor, “Non-Malleable Cryptography,” STOC 1991, pp. 542–552.

[7] A De Santis, G. D. Crescenzo, R. Ostrovsky, G. Persiano, A. Sahai, “Robust Non-interactive Zero Knowledge,” CRYPTO 2001, pp. 566–598.

[8] A. Herzberg, S. Jarecki, H. Krawczyk, M. Yung, "Proactive secret sharing or: How to cope with perpetual leakage," LNCS 963, Proc. Crypto’95, Springer Verlag, 1995, pp. 339–352.

[9] R. Ostrovsky, M. Yung: How to Withstand Mobile Virus Attacks, (Extended Abstract), PODC 1991, pp. 51–59.

[10] R. Canetti, I. Damgard, S. Dziembowski, Y. Ishai, T. Malkin: On Adaptive vs. Non-adaptive Security of Multiparty Protocols. EUROCRYPT 2001: 262–279.

[11] R. Canetti, U. Feige, O. Goldreich, M. Naor, “Adaptively Secure Multi-Party Computation,” STOC 1996, pp. 639-648

[12] M. H. Ibrahim, “Receiver-Deniable Public-Key Encryption,” international journal of network security (ijns), Vol. 8, No. 2, 2009, pp. 159–165.

[13] R. Canetti, R. Gennaro, “Incoercible multiparty computation,” in Proceedings of the 37th Annual Symposium on Foundations of Computer Science, 1996, pp. 504–513.

[14] D. Beaver, “Plug and Play Encryption,” in proceedings of CRYPTO 1997, pp. 75–89.

[15] R. Canetti, S. Halevi, J. Katz, “Adaptively-Secure, Non-interactive Public-Key Encryption,” in proceedings of TCC 2005, pp. 150–168.

[16] R. Canetti, U. Feige, O. Goldreich, M. Naor, “Adaptively Secure Multi-Party Computation,” STOC 1996, pp. 639–648.

[17] J. B. Nielsen, “Separating Random Oracle Proofs from Complexity Theoretic Proofs: The Non-committing Encryption Case,” CRYPTO 2002, pp. 111–126.

[18] I. Damgard, J. B. Nielsen, “Improved Non-committing Encryption Schemes Based on a General Complexity Assumption,” CRYPTO 2000, pp. 432–450.

[19] S. Jarecki, A. Lysyanskaya, “Adaptively Secure Threshold Cryptography: Introducing Concurrency, Removing Erasures,” EUROCRYPT 2000, pp. 221–242.

[20] B. Meng, “A Critical Review of Receipt-Freeness and Coercion-Resistance,” Information Technology Journal, Volume 8 Issue 7, 2009.

[21] D. Dolev, C. Dwork, O. Waarts, and M. Yung, “Perfectly secure message transmission,” Journal of the ACM, 40(1), Jan. 1993, pp. 17-47.

[22] K. Srinathan, A. Patra, A. Choudhary, C. P. Rangan, “Probabilistic Perfectly Reliable and Secure Message Transmission - Possibility, Feasibility and Optimality,” INDOCRYPT 2007, pp. 101-122.

[23] A. Patra, A. Choudhary, K. Srinathan, C. P. Rangan, “Constant Phase Bit Optimal Protocols for Perfectly Reliable and Secure Message Transmission,” INDOCRYPT 2006: pp. 221-235.

[24] M. Fitzi, M. K. Franklin, J. A. Garay, S. H. Vardhan, “Towards Optimal and Efficient Perfectly Secure Message Transmission,” TCC 2007, pp. 311-322

[25] K. Kurosawa, K. Suzuki, “Truly Efficient 2-Round Perfectly Secure Message Transmission Scheme,” EUROCRYPT 2008, pp. 324-340.

[26] E. Berlekamp and L. Welch, “Error correction of algebraic block codes.” US Patent, 4,633,470, USA.

[27] T. El-Gamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” IEEE Transactions on Information Theory, 31(4), 1985, pp. 469–472.

[28] R. Cramer, V. Shoup, “A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack,” CRYPTO 1998, pp. 13–25.

[29] P. Paillier, “Public-Key Cryptosystems Based on Composite Degree Residuosity Classes,” EUROCRYPT 1999, pp. 223–238.

[30] M. Bellare, P. Rogaway, “Random Oracles are Practical: A Paradigm for Designing Efficient Protocols,” ACM Conference on Computer and Communications Security, 1993, pp. 62–73.