2009

Receiver-deniable Public-Key Encryption

Maged Ibrahim, *Helwan University*

Receiver-deniable Public-Key Encryption

Maged H. Ibrahim*
Department of Electronics, Communications and Computers, Faculty of Engineering, Helwan University
1, Sherif St., Helwan, Cairo, Egypt. (Email: mhii72@hotmail.com)
(Received July 25, 2007; revised Dec. 5, 2007; and accepted Mar. 3, 2008)

Abstract

Incoercible (or deniable) encryption is an important notion that allows a user (a sender and/or a receiver) to escape a coercion attempted by a coercive adversary. Such an adversary approaches the coerced user after transmission forcing him to reveal all his random inputs used during encryption or decryption. Since traditional encryption schemes commits the user to his random inputs, the user is forced to reveal the true values of all his random inputs (including the encrypted/decrypted messages and the encryption/decryption keys) which are verifiable by this coercer using the intercepted ciphertext. In this scenario, a coercer may force the user to perform actions against his wish. An appealing property in the mediated RSA PKI introduced in [2] is that, the user has no information, neither about his full private (decryption) key, nor the factorization of the RSA public modulus, which represents an excellent step toward achieving incoercibility in public key encryption, since, a coercer cannot ask the user to reveal such unknown information. In this paper we present a scheme for *receiver-deniable* public-key encryption, by which, the receiver is able to lie about the decrypted message to a coercer and hence, escape a coercion. On one hand, the receiver is able to decrypt for the correct message, on the other hand, all the information held by the receiver, when opened to a coercer, do not allow this coercer to verify the encrypted message and consequently, approaching this user becomes useless from the very beginning.

Keywords: Deniable encryption, mediated PKI, oblivious transfer, public-key encryption, RSA

1 Introduction

While traditional encryption schemes protect the privacy of the sender and the receiver against eavesdroppers (semantic security), they fail to provide protection against coercers. A coercive adversary has the power to approach the user (the sender and/or the receiver) after the ciphertext is transmitted and of course recorded by this adversary. She commands the user to reveal all his random inputs used during encryption or decryption. Since the ciphertext produced using standard encryption schemes (specially, public-key encryption) commits the user to his random inputs, he cannot lie about the true plaintext. Such commitments allow the coercive adversary to verify the validity of the opened message. Deniable encryption allows a user to escape a coercion. Namely, if this user opens all his inputs (including the claimed encrypted message) to a coercer, the coercer fails to prove the validity or invalidity of the opened message.

Deniable encryption maybe classified according to which party is coerced: sender-deniable encryption schemes are resilient against coercing the sender. The Definitions for receiver-deniable and sender-receiver-deniable follow analogously. When the sender and the receiver initially share a common secret key, this is spoken off as shared-key deniable encryption. In deniable public-key encryption, no pre-shared information and no communications are assumed prior to the encryption process. This follows from the assumptions of standard public-key encryption schemes. Yet, deniable public-key encryption is more challenging than deniable shared-key encryption since the public key of the receiver is already known to everyone including the coercer, consequently, neither the sender nor the receiver can lie about the receiver's public key.

The work in [6] showed that it is possible by simple tricks to transform any sender-deniable encryption scheme to a receiver-deniable encryption scheme and vice-versa. Also, they showed that, with the help of other parties with at least one of them remains un-attacked, it is possible to transform a sender-deniable encryption scheme to a sender-receiver-deniable encryption scheme.

In our recent work of [14], we devised a sender-deniable public-key encryption based on quadratic residuosity of a composite modulus and showed how to device a sender-deniable public-key encryption from any trapdoor permutation. However, when the schemes are transformed to be receiver-deniable using the tricks of [6], the schemes are no more one-move schemes. Considering only one-move schemes, receiver deniability is more challenging than sender-deniablility since in the later case, everyone knows the public-key of the receiver but the private key of the receiver is known only to the receiver who is beyond

*Part of the work in this paper was achieved while the author was visiting Dept. of Computer Science, ITE-UCONN, University of Connecticut, USA.

the reach of the coercer. In the former case, the receiver maybe coerced to reveal his private key which is verifiable using the public key and a dummy message.

Deniable encryption is very useful in the protocols where coercive adversaries come to play as a potential threat. For example, deniable encryption protects voters from being coerced during electronic elections [8, 11]. It is also very useful to protect bidders in electronic auctions. Generally, deniable encryption is very important when a party is forced to act against his/her wish.

Our construction assumes the existence of the simple and efficient mediated-RSA (mRSA) [2] as the PKI in place. mRSA was invented as a method to achieve fast revocation in RSA PKI. mRSA involves a special entity, called the SEM (SEcurity Mediator), an on-line partially trusted server, to help signing or decrypting messages. The CA generates the private key d corresponding to Bob's (the receiver's) public key e and splits this private key into two pieces. One piece (d_{SEM}) is delivered to the SEM and the other piece (d_{Bob}) is delivered to Bob. The pair (e, N) is the usual RSA public key. *An important property to notice here is that Bob himself has no information neither about his full private key, nor about the factorization of the public modulus N.* This property is an excellent step toward achieving deniability since, a coercer will not ask Bob to reveal such unknown information. To decrypt a received ciphertext, C, each party (Bob and SEM) performs his/her partial decryption on C, finally the partial decryptions are combined to recover the plaintext message M. To revoke Bob ability to sign or decrypt messages, the CA instructs the SEM to stop issuing partial decryptions or signatures (spoken of as tokens) for Bob public key. At this instant, Bob's signature and/or decryption capabilities are revoked. The functionality is equivalent to (and indistinguishable from) standard RSA due to the fact that the splitting of the private key is transparent to the outside, i.e., to those who use the corresponding public key. Also, knowledge of a half-key cannot be used to derive the entire private key. Therefore, neither Bob nor the SEM can decrypt or sign a message without mutual consent.

As our PKI is established, we turn to discuss our tools. To complete the deniability service, we need an efficient protocol for (1-out-of-n) oblivious transfer (OT_n^1). Rabin [20] proposed the concept of oblivious transfer (OT) in the cryptographic scenario. In this case the sender has only one secret bit b and would like to have the receiver to get it with probability 1/2, on the other hand, the receiver does not want the sender to know whether it gets b or not. For OT_2^1, the sender has two secrets b_1 and b_2, the receiver will get one of them at the receiver's choice. The receiver does not want the sender to know which bit he chooses and the receiver must not know any information other than what he has chosen. OT_n^1 is a natural extension of the OT_2^1 to the case of n secrets. However, constructing OT_n^1 from OT_2^1 is not a trivial problem. OT_n^1 is also known as "All or nothing disclosure of secrets (ANDOS)" [3, 9, 16, 21]. Oblivious transfer is a fundamental primitive in many cryptographic applications and secure distributed computations and has many applications such as private information retrieval (PIR), fair electronic contract signing, oblivious secure computation, etc. [7, 19]. Our proposed receiver-deniable public-key encryption scheme requires one invocation of OT_n^1 between Bob and the SEM. Also we need a SecureID mechanism [12] to make the OT protocol deniable.

The rest of the paper is organized as follows: Section 2 describes the related work. Section 3 gives our motivations and contributions. The underlying PKI and the oblivious transfer protocol are described in Section 4. Section 5 states our assumptions and model. Section 6 presents our weak RD-PKE scheme. The strong version is given in Section 7. Section 8 shows the techniques to transform deniability. The bandwidth is improved in Section 9. Finally, the conclusions are given in Section 10.

2 Related Work

The work in [6] constructed a sender-deniable public-key encryption scheme based on trapdoor permutations. However, the scheme (as stated in [6]) falls short of achieving an appropriate level of deniability, that is, to achieve a high deniability, the size of the ciphertext corresponding to a one bit encryption is super-polynomial and hence inefficient. In the deniable public-key encryption scheme of [6], a one bit plaintext requires tn bits of ciphertext where t is the bit-length of elements in a translucent set $\mathcal{S}_t$ and $t = s + k$ for security parameters n, s and k. The scheme provides deniability of $4/n$ and decryption error of $n2^{-k}$. Hence, to achieve a high level of deniability and a sufficiently low decryption error, the ciphertext is super-polynomial and almost impractical. [6] constructed two deniable public-key encryption schemes based on translucent sets, the first represents the building block for the second which they have called, the "Parity Scheme". The work in [6] also notified that in order to build a one-round scheme, different approaches are required. Also, [6] introduced techniques for the less challenging, deniable shared-key encryption and showed that the one-time-pad is a perfect deniable shared-key encryption.

Based on the sender-deniable public-key encryption, the work in [4] described a general multiparty computations allowing a set of players to compute a common function of their inputs with the ability to escape a coercion. In fact, deniable encryption has an impact on designing adaptively secure multiparty computations [5] since, the notion of deniability is stronger than the notion of non-committing encryption.

In our recent work of [14], we devised a sender-deniable public-key encryption based on quadratic residuosity of a composite modulus and showed how to device a sender-deniable public-key encryption from any trapdoor permutation. However, when the schemes are transformed to be receiver-deniable, the schemes are no more one-move

schemes.

3 Motivations and Contributions

In this Section we describe our motivations and contributions of the work in this paper.

3.1 Motivations

Deniable public-key encryption is a strong primitive, essential in all cryptographic protocols where a coercive adversary comes to play with high potential. Deniable public-key encryption realizes the "Receipt-freeness" attribute which is a very important attribute in electronic voting, electronic bidding and auctions. The schemes proposed in [6] fall short of achieving the desired level of deniability and correctness unless the size of the ciphertext corresponding to a one bit encryption is super-polynomial. An appealing property in the mRSA PKI [2] is that the user himself has no information neither about his full private key, nor about the factorization of the public modulus N, consequently, a coercer will not ask the user for such unknown information.

3.2 Contributions

The contributions of this paper is to introduce an efficient receiver-deniable public-key encryption (RD-PKE) scheme. Our proposed scheme enjoys the following properties:

- It is a one-move scheme without any pre-encryption information required to be sent between the sender and the receiver prior to encryption.
- No pre-shared secret information is required between the sender and the receiver.
- Achieves a high level of deniability equivalent to the factorization of a large two-prime modulus.
- No deciphering errors.
- The bandwidth (ciphertext bit-length) is significantly improved compared to previous constructions.

Efficiency. We reduce the required bandwidth (ciphertext bit-length) to, $2 \lg N$ bits for a single bit encryption, where N is a two-prime RSA modulus. Moreover, this bandwidth can be efficiently improved, that is, $2 \lg N$ bits of ciphertext allow about $\lg N - \delta$ bits of plaintext encryption where δ is a short randomizing string. At the same time, our scheme provides strong deniability (i.e. undetectable cheating) equivalent to the infeasibility to factor a sufficiently large two-prime modulus. Unlike the schemes of [6], our scheme produces no decryption errors and hence, more reliable. We introduce two versions of our RD-PKE scheme, a weak version to declare our idea and security proofs, then we show a simple modification to improve this weak version to be a strong RD-PKE scheme.

4 Preliminaries

In this Section we give detailed description of the mRSA scheme and the OT_n^1 protocol used in building our RD-PKE scheme.

4.1 Mediated RSA

Mediated RSA was invented as a simple method to achieve fast revocation in public-key cryptosystem. As usual, a trusted certificate authority (CA) sets up the RSA modulus N, the public exponent e and the private exponent d for the user. Next, instead of delivering d to the user, the CA splits d into two pieces d_{SEM} and d_{user} such that $d = d_{SEM} + d_{user} \bmod \varphi(N)$ where $\varphi(N)$ is the RSA Euler totient. Finally, the CA secretly delivers d_{user} to the user and d_{SEM} to the SEM.

Encryption. For Alice to encrypt a message $M \in Z_N$ to Bob, she uses Bob's public pair (N, e) to compute the usual RSA ciphertext $C = M^e \bmod N$ and sends C to Bob.

Decryption. On the reception of C by Bob, the decryption process is as follows:

- Bob delivers C to the SEM.
- If Bob's key is revoked, the SEM returns ERROR and aborts, else,
- The SEM computes her partial decryption $PD_{SEM} = C^{d_{SEM}} \bmod N$ and returns PD_{SEM} to Bob.
- Bob computes his partial decryption $PD_{Bob} = C^{d_{Bob}} \bmod N$ and extracts $M = PD_{SEM} PD_{Bob} \bmod N$.

It is important to notice that the SEM gains no information about the decrypted message M [2].

4.2 Oblivious Transfer

Our proposed RD-PKE requires that Bob involves with the SEM in an OT_n^1 invocation to get his encrypted bit. The main objective of the oblivious transfer protocols by Naor and Pinkas in [18] was to improve the efficiency and security of the protocols in [17]. The protocols of [18] have several appealing properties. First, they prove efficiency over previous protocols, second, there are no number theoretic constrains on the strings to be obliviously transferred, third, the protocols have bandwidth-computation tradeoffs which make them suitable for variety of applications. The protocols of [18] operate over a group Z_q of prime order, more precisely, G_q is a subgroup of order q of Z_p^* where p is prime and $q|p-1$. Let g be a generator group and assume that the Diffie-Hellman assumption holds. In their OT_2^1: The sender owns two strings r_0 and r_1. He chooses a random element $U \in Z_q$ and publishes it. The chooser picks a random $1 \leq k \leq q$ and sets $pk_\sigma = g^k$ where $\sigma \in \{0, 1\}$ is the chooser's choice. The chooser also

computes $pk_{1-\sigma} = U/pk_\sigma$ and sends pk_0 to the sender. The sender picks a random R and computes g^R and U^R, he also computes pk_0^R and $pk_1^R = U^R/pk_0^R$. The sender sends g^R as well as the two encryptions, $H(pk_0^R, 0) \oplus r_0$ and $H(pk_1^R, 1) \oplus r_1$ to the chooser, where H is a random oracle modeled by a suitable hash function. The chooser is able to decrypt his choice using pk_σ.

In their OT_n^1: The sender owns n strings, $r_0, ..., r_{n-1}$. He picks $n-1$ random values $U_1, ..., U_{n-1}$ and publishes them, he also picks a random R and sends g^R to the chooser. The chooser selects a random k and sets $pk_\sigma = g^k$ where $\sigma \in \{0, ..., n-1\}$ is his choice, it holds that $pk_i = U_i/pk_0 \ \forall i = (1, ..., n-1)$. The chooser sends pk_0 to the sender. the sender computes pk_0^R as well as $pk_i^R = U_i^R/pk_0^R \ \forall i = (1, ..., n-1)$. The sender sends g^R to the chooser as well as the encryption of each r_i, $H(pk_i^R, w, i) \oplus r_i$ where w is a random string known to both parties. Finally, the chooser is able to decrypt his choice using pk_σ.

In our proposed RD-PKE scheme, during the OT_n^1 invocation, the SEM plays the role of the sender while Bob plays the role of the chooser. More precisely, Bob will hold an index (pointer) to the encrypted bit included in a random string held by the SEM. By one invocation of OT_n^1, Bob is able to get only the pointed bit. Bob knows nothing other than what he gets while the SEM knows nothing about Bob's choice.

5 Assumptions and Model

We define a *receiver-deniable public-key encryption (RD-PKE) scheme* as a scheme by which, the receiver is able to lie about the decrypted message to a coercer and hence, escape a coercion. On one hand, the receiver is able to decrypt for the correct message, on the other hand, all the information held (or extractable) by the receiver when opened to a coercer, do not allow this coercer to verify the encrypted message and consequently, approaching the receiver becomes useless from the very beginning.

The participants in our scheme are the certificate authority (CA), the security mediator (SEM), the sender (Alice), the receiver (Bob) and the coercive adversary (coercer). As usual, the CA is assumed to be fully trusted by all participants. The SEM is a semi-trusted party in the sense that it follows the execution steps word for word but it is willing to learn any information that could be leaked during execution. Alice is assumed to be beyond the reach of any coercer while Bob is possibly coerced.

The coercer has the power to approach Bob coercing him to reveal the decrypted message, the decryption partial key and all the parameters he used during decryption. In our weak version of the scheme, the coercer has the ability to eavesdrop the channel between Alice and Bob while the channel between Bob and the SEM is beyond his reach and assumed untappable. In our full version, we assume that the coercer can eavesdrop both channels.

We introduce our RD-PKE in the weakest notion of semantic security, namely, probabilistic encryption [10] or equivalently, indistinguishable chosen plaintext attack (IND-CPA) model. Since we concentrate on the deniability notion, we do not consider CCA security model in this paper although one may realize security against such attacks by applying the generic constructions of [1] or by employing hybrid encryption techniques [15].

6 Our Weak RD-PKE Scheme

In this Section we introduce the weak version of our RD-PKE scheme. The scheme is weak in the sense that the channel between Bob (the receiver) and the SEM is assumed to be beyond the reach of the coercer. The coercer can only eavesdrop the link between Alice and Bob. This version of our RD-PKE – although weak – yet practical if one assumes that the coercer is outside the domain of Bob. We describe the scheme allowing one bit encryption at a time. The reader will notice that the scheme can be easily adapted to allow multiple bits encryption at a time. We assume that an mRSA PKI is already in place. Hence, the pair (e, N) represents Bob's public key while d_{Bob} (respectively d_{SEM}) are the pieces of Bob's private key d held by Bob (respectively the SEM). Let b_t be the true bit to be encrypted by Alice to Bob. The scheme is described next.

Encryption. To encrypt the bit b_t to Bob, Alice proceeds as follows:

- Picks a $\lg N$ bits string $R \in_{\mathcal{R}} Z_N$. Let $r_0 ... r_{n-1}$ be the binary representation of R.
- Scans the binary representation of R for an index (pointer) i such that $r_i = b_t$.
- Computes and sends the two encryptions, $C_i = i^e \bmod N$ and $C_R = R^e \bmod N$ to Bob.

  We emphasize that the encryption of i must be probabilistic (e.g. random padding).

Decryption. On the reception of C_i and C_R by Bob, the decryption process must end such that the SEM only knows R while Bob only knows the pointer i. The decryption process is as follows:

- Bob performs his partial decryptions on C_R and C_i to compute $PD_R^{(Bob)} = C_R^{d_{Bob}} \bmod N$ and $PD_i^{(Bob)} = C_i^{d_{Bob}} \bmod N$.
- Bob sends $PD_R^{(Bob)}$ and C_i to the SEM.
- The SEM performs her partial decryption on C_i to compute $PD_i^{(SEM)} = C_i^{d_{SEM}} \bmod N$. She returns $PD_i^{(SEM)}$ to Bob.
- The SEM is able to compute $R = PD_R^{(Bob)} C_R^{d_{SEM}} \bmod N$, while Bob is able to compute $i = PD_i^{(SEM)} PD_i^{(Bob)} \bmod N$.

- At this point the SEM knows R while Bob knows the index i pointing to $r_i = b_t$ in R. Bob and the SEM run one invocation of OT_n^1 oblivious transfer. At the end, Bob gets r_i as the encrypted bit b_t.

Lemma 1. (Privacy). *Under the assumption that the OT protocol is secure (i.e. preserves the privacy of Bob and the SEM), no information is revealed to the SEM about the encrypted bit, b_t.*

Proof. The decryption process is performed such that the SEM only knows the random string R while Bob only knows the index i pointing to the encrypted bit in R. Bob and the SEM perform one invocation of OT_n^1. From the properties of the OT protocol, the SEM gains no information about the index i and consequently knows nothing about which of the $\lg N$ bits of R represents the plaintext bit b_t. □

Lemma 2. (Bob's protection). *Under the assumption that the OT protocol is secure, no information is revealed to Bob about any bit of the random string R other than the bit indexed by i, r_i.*

Proof. Follows directly from the security properties of the OT protocol. The reader must notice that the OT protects Bob from harmful knowledge, since, if Bob receives (or is able to extract) R, the coercer will ask Bob to reveal R, a value that Bob cannot lie about to a coercer. Notice that R is verifiable through the encryption $R^e \bmod N$. □

Lemma 3. (Deniability). *Assuming that the channel between Bob and the SEM is beyond the reach of the coercer, the above RD-PKE allows Bob to safely open r_i as either 0 or 1 to a coercer without Bob being detected as a cheater. The deniability of the scheme is equivalent to factoring N.*

Proof. Our assumption that the channel between Bob and the SEM must be physically untappable to a coercer is due to the fact that, the OT protocol is not deniable, the OT protocol commits Bob to the bit he receives from the SEM and all his random choices. When a coercer approaches Bob after recording the transmission from Alice, Bob opens every thing he has. That is, he opens d_{Bob}, i and all the local randomness used in performing the OT_n^1 invocation. Since the SEM-Bob channel is untappable, the opened parameters used in the OT invocation is useless to the coercer. Bob must open i correctly since the coercer can verify with the recorded C_i, yet, the index i alone without knowing the random string R is useless. The coercer cannot ask Bob to reveal R which is known only to the SEM (Lemma 2). Hence, when Bob opens r_i either honestly or dishonestly, this opened bit cannot be verified by the coercer. One may notice that Bob himself cannot verify what he receives from the SEM, he actually relies on the semi-honesty assumption of the SEM. Hence, we deduce that the deniability of our scheme is equivalent to factoring the RSA modulus N. □

Lemma 4. (Correctness). *Given that the SEM is honest-but-curious (semi-trusted) Bob will always get the correct plaintext bit b_t.*

Multi-bit message encryption. It is possible to encrypt $m > 1$ bits to Bob at a time. Alice simply encrypts the concatenations, $I = i_{m-1}||...||i_1||i_0$, of the indices pointing to the m plaintext bits in R. She sends $C_I = I^e \bmod N$ and $C_R = R^e \bmod N$ to Bob. In this case, it requires one invocation of OT_n^1 between Bob and the SEM for each index. Since each index is of $\lg\lg N$ bits, as long as $m \leq \lg N / \lg\lg N$, the bandwidth is still $2 \lg N$. Numerically, for $\lg N = 1024$ bits, $\lg\lg N = 10$ bits and hence, about 102 bits of plaintext requires 2048 bits of ciphertext.

7 Full Deniability

In this Section we show how to achieve full deniability in our RD-PKE scheme, i.e., the scheme will be deniable even if the coercer is capable of eavesdropping the Alice-Bob channel and the SEM-Bob channel as well. The problem is that the OT protocol is not deniable and hence commits Bob to what he receives from the SEM. We benefit from the fact that the SEM and all its users are in the same domain (or system). This fact facilitates the sharing of a time-synchronous pseudo-random string between the SEM server and each user in its domain. Typical example is the OTPs (one time passwords) achieved via secure ID tokens (e.g. the well known and widely used tamper-resistance RSA-SecureID tokens [12]). The SEM and the user in the SEM's domain share a pseudo-random string which is updated every 30 (or 60) seconds at both parties. It is important to notice that this pseudo-random string is synchronously shared based on internal clocks implemented at both parties, consequently, the update is performed offline without any communication, hence this pseudo-random string cannot be reached via eavesdropping. Let $X(\tau)$ be the pseudo-random string shared between Bob and the SEM at any given time interval, τ. The n bits, $x_0, ..., x_{n-1}$ are trivially generated from $X(\tau)$ at both parties (e.g. the least significant n bits of the binary representation of $X(\tau)$). At the end of the OT_n^1 invocation, when the SEM sends the encrypted bits to Bob; instead of encrypting the n bits, $r_0, ..., r_{n-1}$, the SEM encrypts $r_1 \oplus x_1, ..., r_{n-1} \oplus x_{n-1}$. This allows Bob to open a fake r_i since he can easily lie about x_i. The cheating is undetectable even if Bob is coerced to show the token to the coercer at a later time.

8 Deniability Transformation

Our proposed scheme cannot withstand coercion of the sender, since a coerced sender is forced to reveal R and the index i which are verifiable by the coercer using the receiver's public key. A sender-deniable encryption is easily transformed to a receiver-deniable encryption and

vice-versa as follows [6]: Let $\mathcal{A}$ be our receiver-deniable public-key scheme. Let b_t be the bit to be encrypted and transmitted from the sender to the receiver. The receiver chooses a random bit b and invokes scheme $\mathcal{A}$ to encrypt and send b to the sender (as if the sender and the receiver have exchanged places). The sender replies by $b \oplus b_t$ in the clear.

A sender-receiver deniable scheme requires n intermediaries, $I_1, ..., I_n$, with at least one of them remains honest (unattacked). The sender chooses n bits $b_1, ..., b_n$ such that $\bigoplus_i b_i = b_t$ and sends b_i to each I_i using the sender-deniable public-key encryption. Each I_i transmits b_i to the receiver using the receiver-deniable public-key encryption. Finally, the receiver computes $b_t = \bigoplus_i b_i$.

9 Bandwidth Improvement

It is possible to further improve the bandwidth of our receiver-deniable public-key encryption scheme as follows: Let $\mathcal{M}= \{M_0, ..., M_{m-1}\}$ be the set of all possible strings of ℓ bits, then $m = 2^\ell$. For any $\lg N$ bits string $R \in_{\mathcal{R}} Z_N$, the condition, $\ell.2^\ell < \lg N$, allows the 2^ℓ strings to be contained in R. Alice sets R as $r||M_0||M_1...||M_{m-1}$ where r is a randomizing string for the purpose of probabilistic encryption (e.g. in the order of 128 bits). According to the plaintext message, Alice sets the indices $\mathcal{I}= i_{v-1}, ..., i_0$ where i_j points to M_{i_j} in R. In this case, each index i_j is of ℓ bits where $\ell < \lg\lg N - \lg \ell$. The maximum number of indices per encryption (i.e. contained in C_I) is $v_{max} \simeq \lg N/\ell$. Since each index points to a string of ℓ bits, then, the encryption pair ($C_\mathcal{I}$, C_R) encrypts about $\ell \lg N/\ell = \lg N$ bits of plaintext. Hence, for a 1024 bits RSA modulus, a 2048 bits of ciphertext encrypts $1024 - \delta$ bits of plaintext where δ is the bit-length of r. Finally, for each index, i_j, Bob involves with the SEM in one invocation of OT^1_m oblivious transfer of strings to get M_{i_j}. We must emphasize that, $\ell.2^\ell$ must be less than $\lg N$ to allow enough space for a random padding r.

10 Conclusions

We proposed a scheme for receiver-deniable public-key encryption. Our scheme is based on mediated RSA PKI. Our scheme proves efficiency over that proposed in [6] in the sense of bandwidth, deniability and decipherability. The scheme can be transformed to a sender-deniable or a sender-receiver-deniable using the tricks of [6]. The complexity of the oblivious transfer protocol used in our RD-PKE was studied and improved in [13]. the reader may have noticed that, our proposed scheme is not restricted to RSA. Our scheme could be applied to any PKI with the mediated property. A final thing worth noting is that when our receiver-deniable scheme is transformed to a sender-deniable one, it is no more a one-move scheme. To construct a one-move sender-deniable scheme, other approaches must be invented. For example, one may consider the sender-deniable scheme in [14].

Acknowledgements

The author would like to thank the anonymous reviewers of the IJNS for their valuable comments.

References

[1] M. Bellare, and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," *1st Conference on Computer and Communications Security*, pp. 62-73, 1993.

[2] D. Boneh, X. Ding, G. Tsudik, and M. Wong, "A method for fast revocation of public key certificates and security capabilities," *Proceedings of the 10th USENIX Security Symposium*, pp. 297-308.

[3] C. Cachin, S. Micali, and M. Stadler, "Computationally private information retrieval with polylogarithmic communication," *Advances in Cryptography - Eurocrypt '99*, pp. 402-414, 1999.

[4] R. Canetti, and R. Gennaro, "Incoercible multiparty computation," *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, pp. 504-513, 1996.

[5] R. Canetti, U. Feige, O. Goldreich, and M. Naor, "Adaptively secure multi-party computation," *Proccedings 28th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 639-648, 1996.

[6] R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky, "Deniable encryption," *Proceedings of the 17th Annual international Cryptology Conference on Advances in Cryptology*, pp. 90-104, Springer-Verlag, London, 1997.

[7] B. Chor, O. Goldreich, E. Kushilevitz, and M. Susdan, "Private information retrieval," *Journal of the ACM*, vol. 45, no. 6, pp. 965-982, 1998.

[8] R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure and optimally efficient multi-authority election scheme," *Eurocrypt '97*, pp. 103-118, 1997.

[9] Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin, "Protecting data privacy in information retrieval schemes," *Proceedings of 30th Annual ACM Symposium on Theory of Computing*, pp. 151-160, 1998.

[10] S. Goldwasser, and S. Micali, "Probabilistic encryption," *Journal of Computer and System Sciences*, vol. 28, no. 2, pp. 270-299, 1984.

[11] M. Hirt, and K. Sako, "Efficient receipt-free voting based on homomorphic encryption," *Eurocrypt '00*, pp. 539-556, 2000.

[12] (http://www.rsa.com/rsalabs)

[13] M. H. Ibrahim, "Eliminating quadratic slowdown in two-prime RSA function sharing," *International Journal of Network Security (IJNS)*, vol. 7, no. 1, pp. 107-114, 2008.

[14] M. H. Ibrahim, "A method for obtaining deniable public-key encryption," *International Journal of Network Security (IJNS)*, to appear.

[15] K. Kurosawa, and Y. Desmedt, "A new paradigm of hybrid encryption scheme," *Advances in Cryptology - Crypto '04*, LNCS 3152, pp. 426-442, Matthew Franklin, Editor, 2004.

[16] E. Kushilevitz, and R. Ostrovsky, "Single-database computationally private information retrieval," *Proceedings of 38th Annual Symposium on Foundations of Computer Science*, pp. 364-373, 1997.

[17] M. Naor, and B. Pinkas, "Oblivious transfer and polynomial evaluation," *Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing*, pp. 245-254, 1999.

[18] M. Naor and B. Pinkas, "Efficient oblivious transfer protocols," *Proceedings of SIAM Symposium on Discrete Algorithms*, pp. 7-9, 2001.

[19] M. B. Or, S. Goldwasser, and A. Wigderson, " Completeness theorems for non-cryptographic fault-tolerant distributed computation," *Proceedings of the 20th ACM Symposium on the Theory of Computing*, pp. 1-10, 1988.

[20] M. Rabin, *How to Exchange Secrets by Oblivious Transfer*, Technical Report TR-81, Harvard Aiken Computation Laboratory, 1981.

[21] J. Stern, "A new and efficient all-or-nothing disclosure of secrets protocol," *Asiacrypt '98*, pp. 357-371, Springer-Verlag, 1998.

Maged Hamada Ibrahim Received his BSc in communications and computers engineering from Helwan University, Cairo; Egypt. Received his MSc and PhD in Cryptography and Network security systems from Helwan University in 2001 and 2005 respectively. Currently, working as a lecturer, post doctor researcher and also joining several network security projects in Egypt. His main interest is Cryptography and network security. More specifically, working on the design of efficient and secure cryptographic algorithms, in particular, secure distributed multiparty computations. Other things that interest him are number theory and the investigation of mathematics for designing secure and efficient cryptographic schemes.