# A New Sender-Side Public-Key Deniable Encryption Scheme with Fast Decryption

**Tamer Mohamed Barakat**
Department of Electronics and Communications
Faculty of Engineering, Fayoum University, Fayoum, Egypt
[email: tmb00@fayoum.edu.eg]
*Corresponding author: Tamer M. Barakat

*Received March 24, 2014; revised July 13, 2014; accepted July 9, 2014; published September 30, 2014*

## *Abstract*

Deniable encryption, introduced in 1997 by Canetti, Dwork, Naor, and Ostrovsky, guarantees that the sender or the receiver of a secret message is able to "fake" the message encrypted in a specific ciphertext in the presence of a coercing adversary, without the adversary detecting that he was not given the real message. Sender - side deniable encryption scheme is considered to be one of the classification of deniable encryption technique which defined as resilient against coercing the sender. M. H. Ibrahim presented a sender – side deniable encryption scheme which based on public key and uncertainty of Jacobi Symbol [6]. This scheme has several problems; (1) it can't be able to derive the fake message $M_f$ that belongs to a valid message set, (2) it is not secure against Quadratic Residue Problem (QRP), and (3) the decryption process is very slow because it is based dramatically on square root computation until reach the message as a Quadratic Non Residue (QNR).
The first problem is solved by J. Howlader and S. Basu's scheme [7]; they presented a sender side encryption scheme that allows the sender to present a fake message $M_f$ from a valid message set, but it still suffers from the last two mentioned problems.
In this paper we present a new sender-side deniable public-key encryption scheme with fast decryption by which the sender is able to lie about the encrypted message to a coercer and hence escape coercion. While the receiver is able to decrypt for the true message, the sender has the ability to open a fake message of his choice to the coercer which, when verified, gives the same ciphertext as the true message. Compared with both Ibrahim's scheme and J. Howlader and S. Basu's scheme, our scheme enjoys nice two features which solved the mentioned problems: (1) It is semantically secure against Quadratic Residue Problem; (2) It is as fast, in the decryption process, as other schemes.
Finally, applying the proposed deniable encryption, we originally give a coercion resistant internet voting model without physical assumptions.

***Keywords:*** Deniable encryption, probabilistic encryption, quadratic residue problem, Composite Residuosity Classes, public key cryptosystem.

**http://dx.doi.org/10.3837/tiis.2014.09.016**

## 1. Introduction

One of the central goals of cryptography is protecting the secrecy of a transmitted message. The secrecy property of an encryption scheme is usually formalized as semantic security [1], which guarantees that an adversary cannot gain even partial information about an encrypted message.

The notion of semantic security has proven to be very useful in a large number of applications. However, there are some scenarios where semantic security is not sufficient. For example, semantic security does not ensure message secrecy if the adversary can coerce the sender or the receiver of a message to reveal the secret keys and/or the randomness that was used to form an encryption. Specifically, semantic security does not prevent an encryption scheme from being committing, in the sense that if an adversary sees a ciphertext and then tries to coerce the sender to reveal all of the input to the encryption (i.e., both the message and the randomness), any inputs that the sender can reveal that are consistent with the ciphertext must reveal the true message encrypted. In fact, many encryption schemes have only one set of possible inputs per ciphertext.

This committing property of encryption can be problematic in applications such as electronic voting [2] or keeping information secret when facing a coercer using physical force, or in the case of secure multi-party computation in the presence of an adaptive adversary [3].

Suppose that Eve has two children: Alice, who is away at college, and a young Bob, who still lives at home. The siblings are planning a surprise party for Eve, so to keep their plans secret; they communicate using public-key encryption. Eve, however, has taken note of their encrypted communications and grows suspicious.

Using her inherent parental authority, she demands that Alice and Bob reveal their secret decryption keys, as well as any of the encryption randomness they might have retained. Is there any way for Alice and Bob to comply, without spoiling the surprise? The answer seems to be obviously no: using the secret keys, Eve can simply decrypt their messages and learn about the party.

However, the above argument misses a subtle point: if Alice and Bob are able to produce alternative keys and randomness that are consistent with their ciphertexts so far, then they might be able to fool Eve into thinking that they are communicating about something else (or at least not alert her to the party). A scheme that makes this possible is said to be deniable, a notion formally introduced by R. Canetti et al. [4]. (Deniability is related to, but different from Benaloh and Tuinstra's notion of uncoercible communication [5], in which a sender is able to undetectably indicate to a receiver than he is being coerced to send a particular message.)

Deniable encryption allows a party to escape coercion. Namely, it allows the sender to produce a ciphertext $C$ that looks like an encryption of a true message $M_t$ and as an encryption of a fake message $M_f$ . Both messages are chosen by the sender. While the receiver is able to decrypt $C$ for $M_t$, the sender is able to open either $M_f$ or $M_t$ to a coercer when verified, produces the same ciphertext $C$. Deniable encryption maybe classified into categories based on which parity is coerced: a sender-side deniable scheme is resilient against coercion of the sender to produce his secret information, and receiver side deniable scheme is analogous to the previous, but in this case the coercion is on the receiver.

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 [7, 8]. It is also very useful to protect bidders in electronic auctions.
Generally, deniable encryption is very important when a party is forced to act with a gun pointing at his/her head.
We distinguish two types of deniability according to the time of coercion: plan-ahead-deniability and unplanned- deniability. In plan-ahead deniability, the sender chooses his fake message during encryption based on what the coercive adversary previously commanded him to do. In unplanned-deniability, the sender must be able to generate the fake message after transmission whenever a coercive adversary approaches him. Our proposed method is of the later type i.e. we assume that the coercer approaches the sender after transmission and the sender must be able to open any message satisfactory to the coercer.

M. H. Ibrahim in [6] presented a method for sender side deniable encryption based on public key and uncertainty of Jacobi Symbol. This scheme is suffered from several problems; it can't be able to derive the fake message $M_f$ that belongs to a valid message set, it is not secure against Quadratic Residue Problem (QRP) [10], and the decryption process is very slow because it is based dramatically on square root computation until reach the message as a Quadratic Non Residue (QNR). Some applications such as internet voting protocol, electronic bidding and auctions, where the number of users is very large, require fast decryption to complete the authintication process between sender and reciver before sending the required data. This process must take a minimal time since the final dicision for electronic bidding for example is depends mainly on the data sent from users electronically. Also, the authentication and digital signature processes in wireless network specially in wireless sensor network take along time as well as consume very high power for sensor devices if we use atraditional encryption scheme or the current deniable encryprion schems to secure the required data.

J. Howlader and S. Basu [9] presented a sender side encryption scheme that solve the first mentioned problem of Ibrahim's scheme which allows the sender to present a fake message $M_f$ from a valid message set.

Unfortunately, J. Howlader and S. Basu's scheme still suffers from the reset of Ibrahim's scheme problems; it is not secure against Quadratic Residue Problem (QRP) and the decryption process is very slow due to it is based dramatically on square root computation until reach the message as a Quadratic Non Residue (QNR).

In this paper we present a new sender-side deniable public-key encryption scheme which is semantically secure against QRP. Moreover, we will show that our scheme is as fast, in the decryption process, as both Ibrahim's scheme and J. Howlader and S. Basu's scheme .

Also, we develope a secure internet voting model based on the proposed deniable scheme is originally developed.
The paper is organized as follows: Section 2 describes the related work in the field. Our motivations and contributions are given in Section 3. Section 4 describes the preliminaries and the notion of deniability. In Section 5 we present the proposed deniable encryption with its encryption and decryption techniques. We present the running time of the proposed scheme in the Section 6. Section 7 desribes internet voting protocol using the proposed scheme. Implementation data of the proposed scheme is presented in Section 8. Finally, the conclusions are given Section 9.

## 2. Related Work

More recently, O'Neill, Peikert, and Waters [11] announced a flexible bi-deniable encryption scheme with negligible deniability based on lattice assumptions. We view this latter work as orthogonal to our own: it is noninteractive and achieves deniability for both sender and receiver simultaneously, but the construction uses in

an essential way the fact that there are different honest and dishonest encryption algorithms. the work in [3] described a general multiparty computations allowing a set of players to compute a com- mon function of their inputs with the ability to escape a coercion.
Canetti et al. [4] also constructed a flexible (i.e., two-algorithm) sender-deniable encryption scheme with negligible deniability. The work in [5] also notified that in order to build one-round schemes, different approaches are required. Also it 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.

Ibrahim [6] devises 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. He supposes that s is generated and used on the fly to reach a QNR value in . He supposes that the program does not store s anywhere on the system since it is not part of the encryption pattern.

## 3. Motivations and Contributions

### 3.1 Motivations

Deniable encryption offers exactly the missing part. Given a ciphertext, public-key, all secret knowledge, and an alternative message, the sender and/or receiver is able to compute alternative secret knowledge (i.e., encryption algorithm randomness or secret key). The alternative secrets are required to be indistinguishable from honest secrets while delivering the alternative message.
The main motivation of deniable encryption is coercion resistance. A powerful adversary may demand secret key and encryption randomness for the intercepted communication. 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 is a very important attribute in some applications such as electronic voting, electronic bidding and auctions.
Deniable encryption has an impact on the design of adaptively secure multiparty computations [3] since, the notion of deniability is stronger than the notion of non-committing encryption.

### 3.2 Contributions

The contributions of this paper are to introduce an efficient sender-deniable public-key encryption scheme. We introduce two versions of our scheme. The first scheme for single bit encryption while the second scheme is for multi-bit message encryption. The main contributions of this paper are described as follows:

- An efficient sender-deniable encryption scheme is proposed. 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 deniability equivalent to the factorization of a large two-prime modulus
- semantically secure against QRP.
- The decryption process is very fast compared to other related scheme.
- The less overhead in term of the size of the ciphertext.

- A secure internet voting model based on the proposed deniable scheme is originally developed. The internet voting model have the following properties:
  - The model is coercion-resistance.
  - Coercion-resistance is implemented without physical assumptions.

## 4. Preliminaries

In this section we first describe the notion of deniability and then we introduce the quadratic residuosity of a composite in some details as it represents the basic primitive of the schemes presented in this paper.

### 4.1 Definition 1:

Let $n \in N$ be a security parameter. An efficiently computable protocol π between two parties $S$ and $R$ (sender and receiver, respectively) is called a sender-deniable public key bit encryption scheme if the following three conditions are satisfied:

• **Correctness:** The probability that the receiver output is different from the sender input is negligible.

• **Security:** For any two different messages $M_t$ and $M_f$, the communications for transmitting $M_t$ are computationally indistinguishable from the communications for transmitting $M_f$.

**Deniability:** The adversary's view of an honest encryption of $M_t$ according to protocol π is indistinguishable from the adversary's view when the ciphertext was generated while transmitting $M_t$ and the sender falsely claims that it is an encryption of $M_f$ .

### 4.2 Definition 2: Quadratic Residuosity

The proposed scheme is based on the quadratic residuosity problem [1, 9, 10], of a composite $n$, which is a product of two distinct primes

- **Basic definitions:**

For an integer $a \in Z_n^*$ is a quadratic residue modulo n, if there exists some $X \in Z_n^*$ such that $a \equiv X^2 \mod n$. We denote $a \in Q_n$. Otherwise a is quadratic nonresidue modulo n and denoted as $a \in \overline{Q_n}$

Define $J_n^+ \subset Z_n^*$ to be the subset of all integers such that for any $a \in J_n^+$ , the Jacobi symbol $\left(\frac{a}{n}\right) = +1$ and define $J_n^- \subset Z_n^*$ to be the subset of all integers such that $a \in J_n^-$ , the Jacobi symbol $\left(\frac{a}{n}\right) = -1$. We have $Q_n \subset J_n^+$.

### 4.3 Definition 3: Computing Composite Residuosity Classes

Let $g$ be some element of $\mathbb{Z}_{n^2}^*$. the computational problem Class [n] defined as follows: given $\omega \in \mathbb{Z}_{n^2}^*$ and $g \in \mathfrak{B}$, compute $[\![\omega]\!]_g$.

**Lemma 1**. For any $u \in S_n$ , where $S_n$ is the multiplcative supgroup of integer modulo $n^2$ and $S_n = \{u < n^2 \mid u = 1 \mod n\}$, the function L is defined as $\forall\, u \in S_n \;\; L(u) = \frac{u-1}{n}$

**Lemma 2**. For any $\omega \in \mathbb{Z}_{n^2}^*$ , $L(\omega^\lambda \mod n) = \lambda\, [\![\omega]\!]_{n+1} \mod n$

## 5. The Proposed Scheme

In this section we propose our scheme for both single bit and multiple-bit message. Firstly, we introduce the proposed scheme for 1-bit message, and then we extend our work for multiple-bit message. In this scheme, the receiver choose two large prime numbers $p$ and $q$. Then, he compute $n = pq$ as the reciever's public-key while $p$ and $q$ as the receiver's private-key. Our scheme is based on probabilistic encryption method [10].

### A. Single bit deniable encryption scheme.

Let $b_t$ be the true bit to be encrypted while $b_f$ be the fake bit. Then, the procedure of the proposed scheme is done as follows:

**Encryption:** the sender proceeds as follows:

- **Honest Encryption $(b_t = b_f)$**

  - Selects two prime $p, q$ and $n$ where $n = pq$.
  - Selects a bit stream $y$ of $k$ bits, where $y$ is QNR..
  - Selects $X \in Z_n^*$ at randome.
  - To negotiate y between the transmitter and receiver without any obscurity; the sender does the following:

  - **Method I** if the $i^{th}$ bit is 0 (i.e, $b_i^y = 0$), computes $a = X^2 \mod n$.

  - **Method II** if the $i^{th}$ bit is 1 (i.e, $b_i^y = 1$), the sender computes $a = yX^2 \mod n$, such that $a \in J_n^+$

  - To ensure that the receiver is able to distinguish whether $X \in Q_N$ or $X \in \overline{Q_N}$ as well as to allow the receiver to stop at the correct QNR which is y in our scheme, we should use a strong hash function H with an output bit-length L as follow:
    - The sender picks $e \in_R \{0,1\}$, sets $R_e = H(y)$ and $R_{1-e} \in_R \{0,1\}^\ell$.
    - Randomly selects $0 < r < n$, and then he computes $C = g^{y+nr} \mod n^2$, where $g$ is some element of $Z_n^*$.

- Scans the binary representation of y for an index i such that $b_i^{(y)} = b_t = b_f$.
- Sends $(i, C, R_0, R_1)$ to the receiver.

- **Dishonest Encryption $(\boldsymbol{b_t = \bar{b}_f})$.**

  - Selects two prime $p, q$ and $n$ where $n = pq$.
  - Selects a bit stream $y$ of $k$ bits, where $y$ is QNR..
  - Picks two small integers $0 < (r_1,\ r_2) < n$ and let $g$ is some element of $\mathbb{Z}_{n^2}^*$ .
  - Computes $y_1 = g^{y+nr_1} \ \ mod\ n^2$.
  - Scans the binary representation of both $y$ and $y_1$ such that $b_i^y = b_t$ and $b_i^{y_1} = b_f$.
  - Computes $C = g^{y_1+nr_2} \ mod\ n^2$.
  - Picks $e \ \in_R \{0, 1\}$, sets $R_e = H(y)$ and $R_{1-e} = H(y_1)$.
  - Sends $(i,\ C,\ R_0,\ R_1)$ to the receiver.

**Decryption:** the receiver decrypts the received message $(i,\ C,\ R_0,\ R_1)$ starting with $C$. Then, the receiver keeps on computing $y\ modulo\ n$ until he reches $y = \frac{L(C^\alpha \ mod\ n^2)}{L(g^\alpha \ mod\ n^2)} \ \ mod\ n$ as a QNR in $J_n^+$ satisfying either $R_0 = H(y)$ or $R_{1-e} = H(y_1)$, where $1 \le \alpha \le \lambda$ and $\lambda = lcm(p-1, q-1)$ . Hence, the receiver decrypt of $b_i^{(y)}$ as the encrypted bit $b$.

- **Proof of security for our scheme.**

**Opening an encryption.** To open an encryption honestly, the sender reveals $y$. To open dishonestly, the sender reveals $y_1$ and claims that $R_e$ is a random string.

**Security.** For any $b_t, b_f \in \{0,\ 1\}$, the communications between the sender and receiver for transmitting $b_t$ is indistinguishable from that for transmitting $b_f$.

**Correctness.** In the decryption process, on the reception of $(i,\ C,\ R_0,\ R_1)$, the receiver (starting from C) keeps on computing y. After each computation, he,(i) discards the two roots in $J_N^-$ (ii) hashes the QNR root in $J_N^+$ to see whether it matches either $R_0$ or $R_1$. If a match is found, he stops and takes this QNR as y. Otherwise, he continues computing y of the QR $J_N^+$ and repeats (i) and (ii). Hence, correctness follows immediately.

**Deniability proof in presence of coercer.** In case of sender-side coercion, the sender reveals $y_1$ dishonestly to the coercer. The sender is able to convince the coercer, that a bit $b_i^y = 0$, whereas the truth is $b_i^y = 1$ . To do this, the sender would say that $a_j$ for $0 \le j \le t-1$ are random selection from $J_n^+$ , that is, randomly selected using Method I, whereas $a_j$ is selected using Method II. However, sender cannot open a bit $b_i^y = 1$, whereas the truth is $b_i^y = 0$. So, in case of coercion the sender would flip a bit $b_i^y = 1$ to 0 by dishonestly opening $y_1$.

On the other hand, the sender falsely claims that $y_1$ is a QNR and $b_i^{(y_1)}$ is the encrypted bit. As $a_j$ is from $J_n^+$ and the coercer does not know the prime factors of $n$, the coercer automatically

accepts this claim since he cannot prove the contradiction, i.e., ha cannot prove that $y_1$ is a QR and that $R_e$ is not random.

## B. multiple bits deniable encryption

In this section, we extend the single bit deniable encryption scheme to multi-bit deniable encryption scheme. Let $M_t$ be the true message to be encrypted and let $M_f$ be the set of all possible fake binary messages of $m$ bits excluding $M_t$. We assume that $m$ is no more than several bits. The cheme is described as follows:

***Encryption:*** the sender proceeds as follows:

- **Honest Encryption $(b_t = b_f)$**
  - Selects two prime $p, q, p \neq q$.
  - He sets $n = pq$ as his public-key while keeping $p$ and $q$ secret.
  - Selects a pseudosquare $y \in Z_n$ $(i.e., y\ is\ QNR)$
  - Let message $m$ be a binary string $m = m_1, m_2, \ldots\ldots m_l$
  - For $i = 1 \ldots\ldots\ldots l$ do:
    - Select $x \in Z_n^*$ at random.
    - If $m_i = 0$, sender computes $a_j = X_j^2\ mod\ n$, where $X_j \in Z_n^*$, for $0 \leq j \leq m-1$
    - Otherwise, he computes $a_j = y\ X_j^2\ mod\ n$.
  - The sender scans the binary representation of $y$ for an index $i_j$ such that $b_{i_j}^{(y)} = b_j^{(M_t)}$.
  - To ensure that the receiver is able to distinguish whether $X \in Q_N$ or $X \in \overline{Q_N}$ as well as to allow the receiver to stop at the correct QNR which is y in our scheme, we should use a strong hash function H with an output bit-length L as follow:
    - Let $\varepsilon = 2^m - 1$. Defines strings $R_0, \ldots\ldots\ldots, R_\varepsilon$, selects a random $i \leq \varepsilon$, and sets $R_i = H(y)$. then, sets each other $R_{j \neq i} \in_R \{0,1\}^\ell$.
  - Randomly selects $0 < r < n$, and then he computes $C = g^{y+nr}\ \ mod\ n$, where $g$ is some element of $Z_n^*$.
  - Sends $(i_{m-1}, \ldots\ldots, i_0, C, R_0, \ldots\ldots, R_\epsilon)$ to the receiver.

- **Dishonest Encryption $(b_t = \overline{b}_f)$.**
  - Selects two prime $p, q$ and $n$ where $n = pq$.
  - Selects a bit stream $y$ of $k$ bits, where $y$ is QNR .
  - Picks two small integers $0 < (r_1,\ r_2) < n$ and let $g$ is some element of $Z_n^*$.
  - Computes $y_1 = g^{y+nr_1}\ \ mod\ n$.
  - Scans the binary representation of both $y$ and $y_1$ such that $b_{i_{m-1}}^{(y)} = b_{m-1}^{(M_t)} \ldots\ldots\ldots b_{i_0}^{(M_t)}$ and $b_{i_{m-1}}^{(y_1)} = b_{m-1}^{(M_f)} \ldots\ldots\ldots b_{i_0}^{(M_f)}$.

- Let $\varepsilon = 2^m - 1$ be the number of strings $y_j$ $(i.e, each\ y_j\ corrsponds\ to\ a\ one\ fake\ M_f)$. Then, Defines strings $R_0, \ldots\ldots, R_\varepsilon$, selects a random $i \leq \varepsilon$, and sets $R_i = H(y)$, and sets each other $R_{j\neq i} \in_R \{0,1\}^\ell$ as a value of $H(y_1)$ .
- Computes $C = g^{y_1 + nr_2} \ mod\ n$.
- Sends $(i_{m-1}, \ldots\ldots, i_0, C, R_0, \ldots\ldots, R_\epsilon)$ to the receiver.

**Decryption:** the receiver decrypts the received message $(i_{m-1}, \ldots\ldots, i_0, C, R_0, \ldots\ldots, R_\epsilon)$ starting with $C$. Then, the receiver keeps on computing $y\ modulo\ n$ until he reches $y = \frac{L(C^\alpha \bmod n^2)}{L(g^\alpha \bmod n^2)} \ mod\ n$ as a QNR in $J_n^+$ satisfying that $R_i = H(y)$ for any $i = 0, \ldots\ldots, \epsilon$. Hence, the receiver decrypt of $b_{i_{m-1}}^{(y)}, \ldots\ldots b_{i_0}^{y}$ as the cleartext bits.

- **Proof of security for our scheme.**

**Opening an encryption.** To open an encryption honestly, the sender reveals $y$. to open dishonestly, the sender reveals $y_1$ and claims that $R_\epsilon$ is a random string.

**Security.** Semantic security is immediate.

**Correctness.** Immediate.

**Deniability proof in presence of coercer.** In case of sender-side coercion, the sender reveals $y_1$ dishonestly to the coercer. The sender is able to convince the coercer, that a bit $b_i^y = 0$, whereas the truth is $b_i^y = 1$ . To do this, the sender would say that $a_j$ for $0 \leq j \leq m-1$ are random selection from $J_n^+$ , that is, randomly selected using Method I, whereas $a_j$ is selected using Method II. Howerver, sender cannot open a bit $b_i^y = 1$, whereas the truth is $b_i^y = 0$. So, in case of coercion the sender would flip a bit $b_i^y = 1$ to 0 by dishonestly opening $y_1$.
On the other hand, the sender falsely claims that $y_1$ is a QNR and $b_i^{(y_1)}$ is the encrypted bit. As $a_j$ is from $J_n^+$ and the coercer does not know the prime factors of $n$, the coercer automatically accepts this claim since he cannot prove the contradiction, i.e., he cannot prove that $y_1$ is a QR.

## 6. Running time of our scheme

In this section, we briefly analyze the main practical aspects of computations required by our scheme.

**Key generation**. The prime factors $p$ and $q$ must be generating according to the usual recommendations in order to make n as hard to factor as possible. The most computationally expensive operation involved in decryption is the modular exponentiation $C \rightarrow C^\alpha \bmod n^2$ $O(|n|^2|\alpha|)$ . If g is chosen in such a way that $|\alpha| = (|n|^\epsilon)$ for some $\epsilon > 0$, then decryption will only take $O(|n|^{2+\epsilon})$ bit operations. On the other hand, the base g can be chosen randomly among elements of order divisible by $n$. the whole generation may be made easier by carrying

out computations separately $mod\ p^2$ and $mod\ q^2$ and Chinese-remaindering $g\ mod\ p^2$ and $g\ mod\ q^2$ at the very end.

**Encryption.** Encryption requires a modular exponentiation of base $g$. The computation may be significally accelerated by a judicious choice of $g$. taking $g = 2$ allow an immediate speed-up of whole encryption process. Optionally, $g$ could even be fixed to a constant value if the key generation process includes a specific adjustment. At the same time, pre-processing techniques for exponentiation a constant base can dramatically reduce the computational cost.

**Decryption.** Computing $L(x)$ for $x \epsilon Z_n$ may be achieved at a very low cost (only one multiplication modulo $2^{|n|}$) by precomputing $n^{-1}$ mod $2^{|n|}$. The constant parameter $L(g^{\alpha}\ mod\ n^2)^{-1}\ mod\ n$ can also be precomputed once for all. On the other hand, decryption process uses Chinese- Remainder Theorem (CRT) [12] which used to efficiently reduce the decryption workload of our scheme. Therefore, the decryption process can be made faster by separately computing the message mod p and mod q and recombining modular residues afterwards:

$$m_p = L_p(C^{p-1}\ mod\ p^2)\ h_p\ mod\ p \quad (1)$$

$$m_q = L_q(C^{q-1}\ mod\ q^2)\ h_q\ mod\ q \quad (2)$$

$$m = CRT\ (m_p, m_q)\ \ mod\ pq \quad (3)$$

***Where:***

- $L_p(x) = \frac{x-1}{p}$ and $L_q(x) = \frac{x-1}{q}$
- $h_p = L_p(g^{p-1}\ mod\ p^2)\ mod\ p$ and $h_q = L_q(g^{q-1}\ mod\ q^2)\ mod\ q$
- $p-1$ and $p-1$ have to be replaced by $\alpha$ in the fast decryption.

## 7. Internet Voting Protocol using the proposed sender deniable encryption scheme

Deniable encryption scheme uses in many applications such as electronic voting protocol, protection against vote buying, auction protocol, secure mutliparty computation and deniable authentication process. This type of deniability is very common in internet voting protocol.
This section will describe how to express the idea of the internet voting protocol model using the proposed sender side deniable encryption scheme.
The proposed internet voting model includes three phases: preparation phase, registration phase and voting phase.

### 7. 1 Preliminaries

In this section, we review some notations and assumptions that will be used in the proposed voting protocol.

#### Notation 1

- $\boldsymbol{ID_j}$ : the identification of voter $V_j$ .
- A : authority which it is responsible for elections.

- $B^t$ : ballot that will be used during the voting process.
- BB : bulletin board.

**Assumption 1**

- In order to express the idea clearly and simplify the model, we suppose there is only one authority.
- We use the proposed single bit deniable encryption scheme where the message set is $\{TRUE, FALSE\}$ or eqivalent to $\{1, 0\}$.

- **Preparation phase:**

  - Authority A and voter $V_j$ generate the public and private keys according to the proposed sender deniable encryption scheme. The private keys of voter and authority are secret
  - Authority generates the ballot $B^t$and send $B^t$and its digital signature to bulletin board denoted by BB .

- **Registration phase:**

  Voter $\mathbf{V_j}$ firstly registered to authority **A** as followes:
  - voter $V_j$ computes $C = g^{y+nr} \mod n^2$ using the proposed sender deniable encryption scheme, where $y = ID_j$. Then, voter sends $(i, C, R_0, R_1)$ as his/her public key to authority A.
  - authority A then decrypt the recived message using the proposed sender deniable encryption scheme to obtain y by satisfying that either $R_0 = H(y)$ or $R_{1-e} = H(y_1)$.
  - Once Authority A obtains the value of y, Authority A begins to register the voter $V_j$ using his/her identification ($ID_j$). **Fig. 1** describes the registration phase.

- **Voting phase:**

  - voter $V_j$ chooses his/her favorite ballot $B^t$.
  - voter $V_j$ then encrypts his/her credential using the proposed sender deniable encryption scheme and sends it to BB at the reciver.
  - The reciever then decrypts the received message using the proposed sender deniable encryption scheme and verify the identification of the voter $V_j$.
  - If the credential is not valid, the protocol is terminated, otherwise BB sends the verification message to voter $V_j$ and ask him/her about his/her valid ballot.
  - After voter $V_j$ recievs the verification message from BB, he/she sends the encrypted ballot, which contains his/her voting dicision, to BB which it accept the reciving voting and put it in its database. **Fig. 2** describes the voting phase.

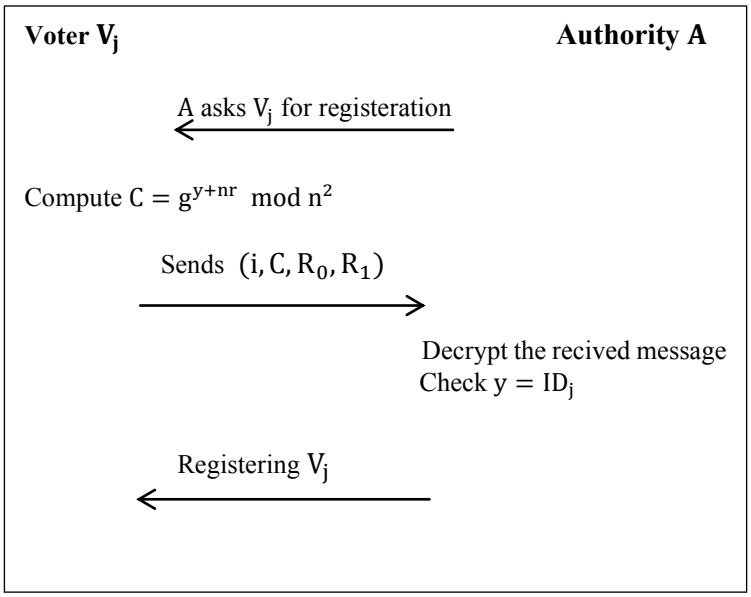



**Fig. 1.** Registering Phase

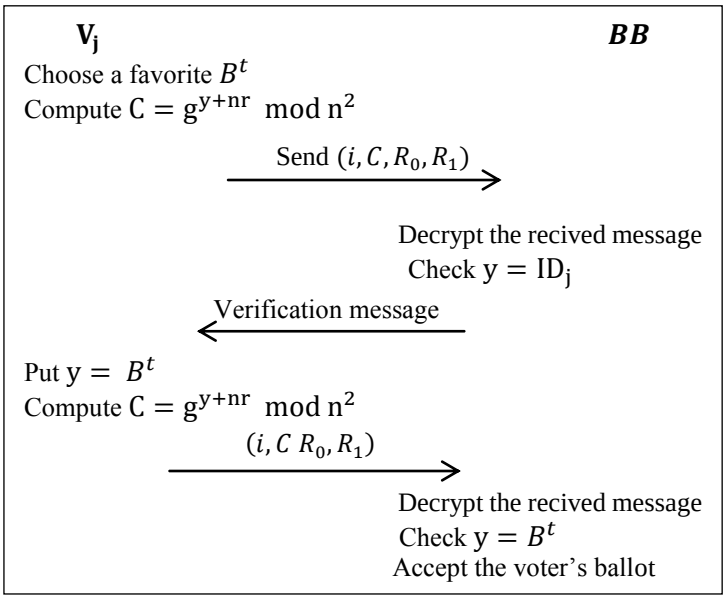



**Fig. 2.** Voting Phase

## 8. Implementation data of the proposed scheme

In this section we analyzing, evaluating, and compring among Ibrahim's scheme, J. Howlader and S. Basu's scheme and the proposed scheme in terms of the following evaluation parameters:

1. Running time of both encryption and decryption processes based on different values of modulus $n$.
2. Computation time.
3. Memory usage.

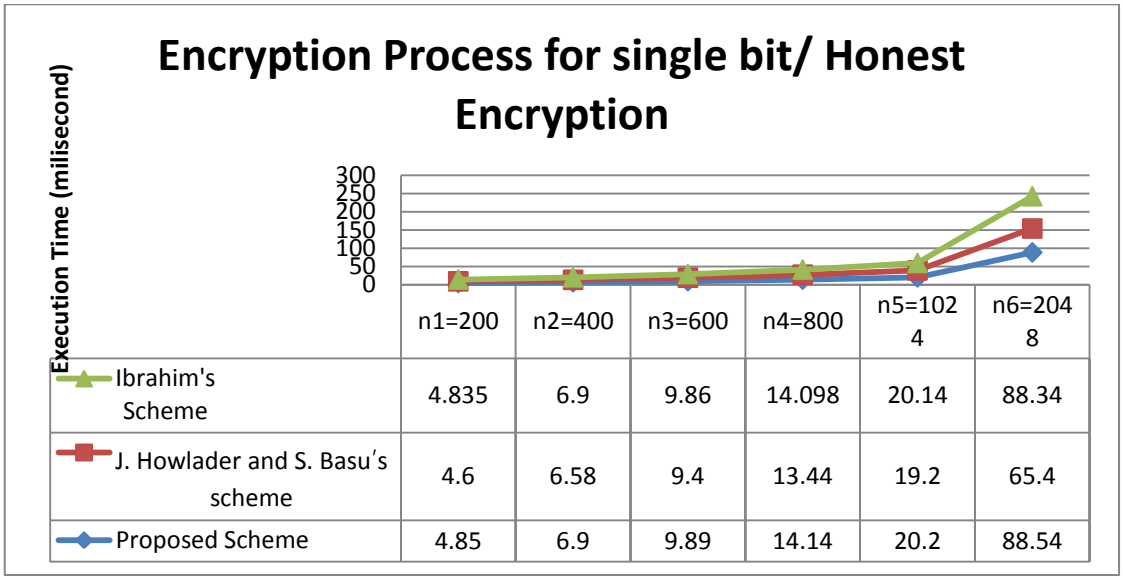
In order to demonstrate the improved efficiency of our scheme, we implemented this scheme on Intel(R) Core(TM) i3 CPU, M370 @ 2.40 GHz with 4 GB RAM using C# programming language.

We implementing six different sizes of the modulus $(n)$, namely at $n = 200$, $n = 400$, $n = 600$, $n = 800$, $n = 1024$, and $n = 2048$ bits For each value of the modulus $(n)$, the modular multiplication of bit size |n| is taken as the unitary operation. We assume that the execution time of a modular multiplication is quadratic in the operand size and that modular squares are computed by the same routine. The public exponent is taken equal to $e = 2^{16} + 1$. The parameter g is set to 2 in our main scheme. Other parameters,
secret exponents or messages are assumed to contain about the same number of ones and zeroes in their binary representation.
The five text files of different sizes are used to conduct five experiments, where a comparison of three algorithms is performed.

### 8.1 Experimental Results and analysis for running time parameter

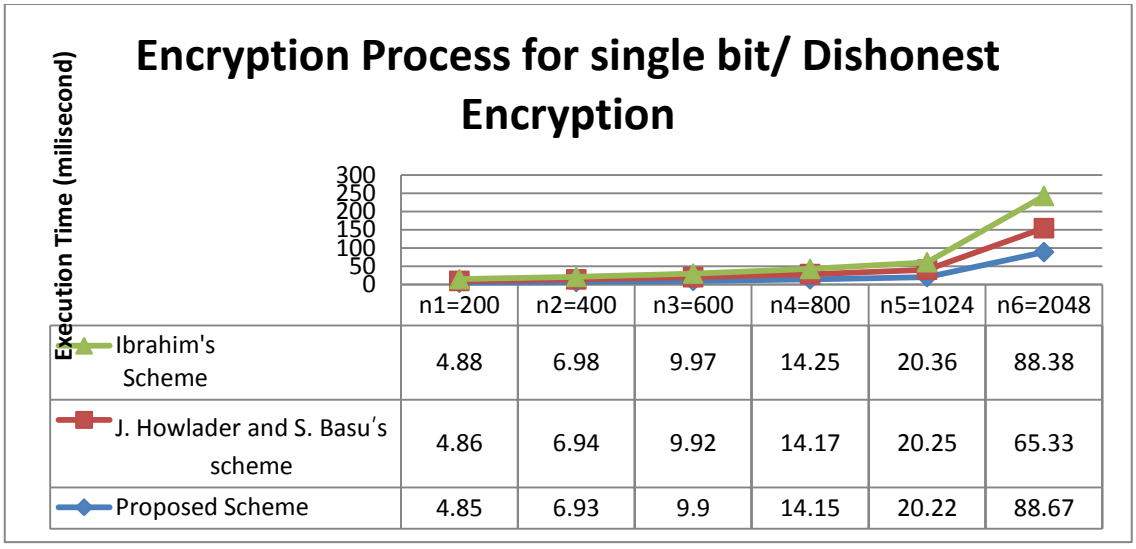
Experimental results of the running time for encryption and decryption algorithms for three schems are shown in **Fig. 3** to **Fig. 8** which show the comparison of three schemes using different values of modulus $n$.
By analyzing **Fig.3** and **Fig. 4** which show the time Taken for encryption process for single bit on various size of modulus $n$ by three algorithms i:e Ibrahim's scheme, J. Howlader and S. Basu's scheme and the proposed scheme. It is noticed that, J. Howlader and S. Basu's scheme consumes least time for encryption. Whereas Ibrahim's scheme and the proposed scheme show very minor difference in time taken for encryption.
**Fig. 5** shows the time taken for decryption process for single bit on various size of modulus $n$. It is noticed that the decryption time for Ibrahim's scheme is the highest for all sizes of modulus $n$, while the proposed scheme takes the lowest decryption time for all sizes of modulus $n$.
Similary, we take the same results when run our expremintal for both encryption and decryption processes for multiple bits. The simulation results are shown in **Fig. 6**, **Fig. 7**, and **Fig. 8**.

### 8.1.1. Simulation Results




| | n1=200 | n2=400 | n3=600 | n4=800 | n5=1024 | n6=2048 |
|---|---|---|---|---|---|---|
| Ibrahim's Scheme | 4.835 | 6.9 | 9.86 | 14.098 | 20.14 | 88.34 |
| J. Howlader and S. Basu's scheme | 4.6 | 6.58 | 9.4 | 13.44 | 19.2 | 65.4 |
| Proposed Scheme | 4.85 | 6.9 | 9.89 | 14.14 | 20.2 | 88.54 |

**Fig. 3.** comparison of encryption process for single bit/ honest encryption among Ibrahim's scheme, J. Howlader and S. Basu's scheme and the proposed scheme




| | n1=200 | n2=400 | n3=600 | n4=800 | n5=1024 | n6=2048 |
|---|---|---|---|---|---|---|
| Ibrahim's Scheme | 4.88 | 6.98 | 9.97 | 14.25 | 20.36 | 88.38 |
| J. Howlader and S. Basu's scheme | 4.86 | 6.94 | 9.92 | 14.17 | 20.25 | 65.33 |
| Proposed Scheme | 4.85 | 6.93 | 9.9 | 14.15 | 20.22 | 88.67 |

**Fig. 4.** comparison of encryption process for single bit/ dishonest encryption among Ibrahim's scheme, J. Howlader and S. Basu's scheme and the proposed scheme

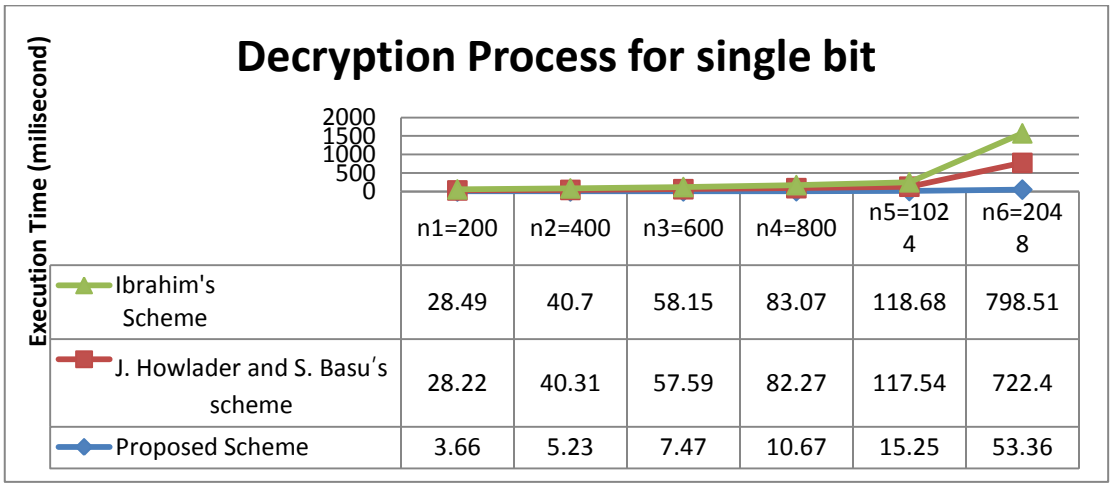



**Fig. 5.** comparison of decryption process for single bit among Ibrahim's scheme, J. Howlader and S. Basu's scheme and the proposed scheme

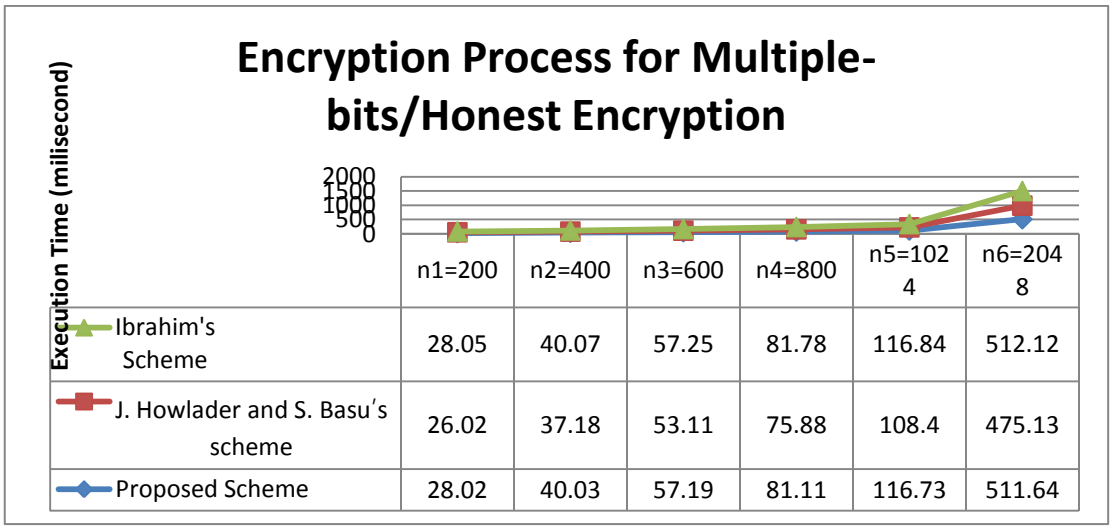



**Fig. 6.** comparison of encryption process for multiple bits/ honest encryption among Ibrahim's scheme, J. Howlader and S. Basu's scheme and the proposed scheme

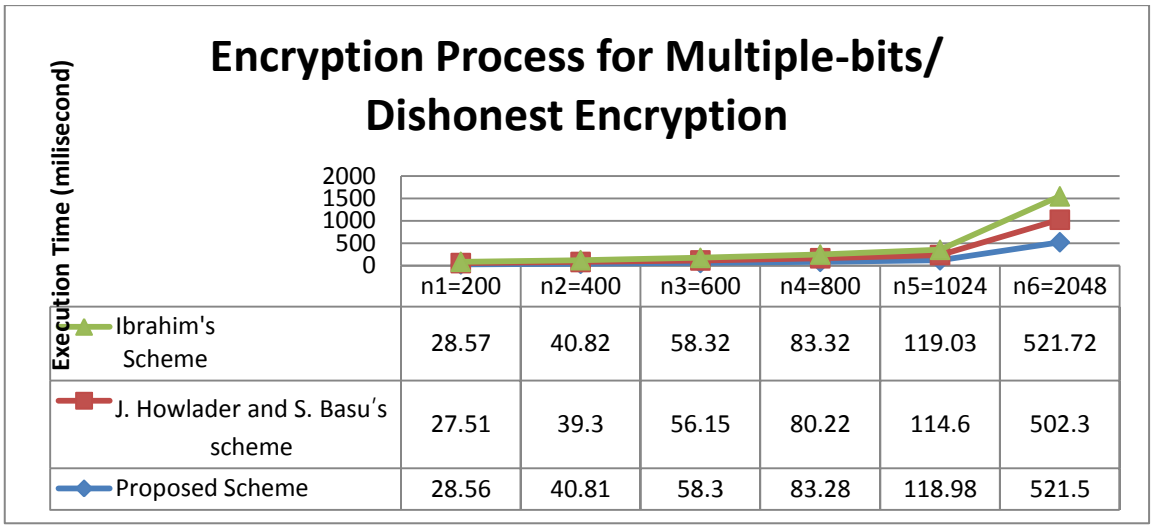



**Fig. 7.** comparison of encryption process for multiple bits/ dishonest encryption among Ibrahim's scheme, J. Howlader and S. Basu's scheme and the proposed scheme

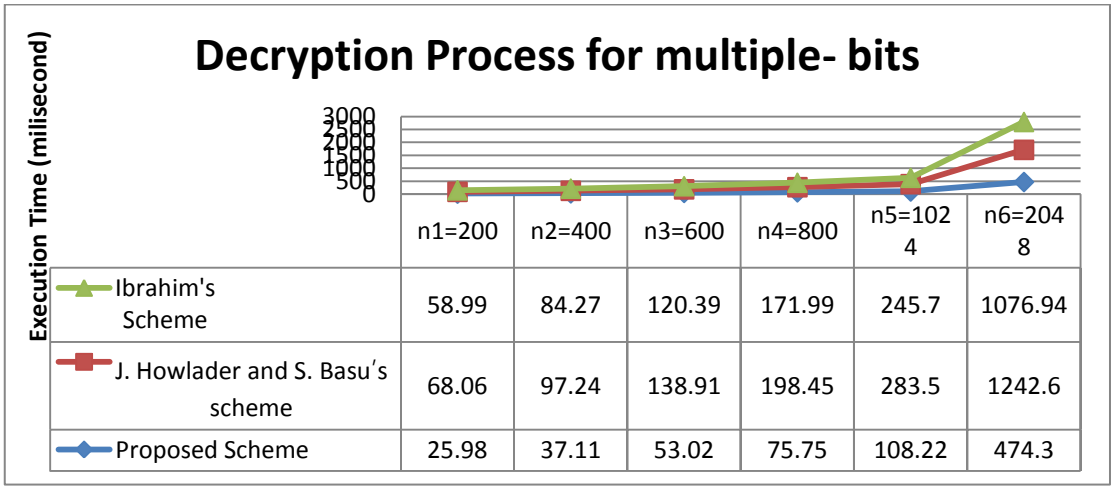



**Fig. 8.** comparison of decryption process for multiple bits among Ibrahim's scheme, J. Howlader and S. Basu's scheme and the proposed scheme

### 8.2 Experimental Results and analysis for computation time and memory usage parameters

In this section, we show the comparison among Ibrahim's scheme, J. Howlader and S. Basu's scheme and the proposed scheme in terms of computation time and memory usage.
The computation time means that the total time of encryption and decryption algorithms which be taken by those schemes at different sizes of files.
Experimental results are shown in **Table 1**, which shows the required comparison using five text files of different sizes.

By analyzing of **Table 1** we noticed that, the computational time taken by the proposed scheme is much lower compare to the time taken by Ibrahim's scheme and J. Howlader and S. Basu's scheme.
Also, we opserved that the memory usage for the proposed scheme is much lower than the memory usage for both Ibrahim's scheme and J. Howlader and S. Basu's scheme. Whereas Ibrahim's scheme and J. Howlader and S. Basu's scheme show very minor difference memory usage. The simulation results are shown in **Fig. 9** and **Fig. 10.**

**Table 1.** Expremintal Results

| Data | Scheme | Computational Time (ms) | Memory Usage (KB) |
|---|---|---|---|
| FILE 1 (68KB) | proposed scheme | 224.95 | 81912 |
| | J. Howlader and S. Basu | 391.9 | 91814 |
| | Ibrahim | 362.54 | 85261 |
| FILE 2 (105 KB) | proposed scheme | 271.35 | 90103 |
| | J. Howlader and S. Basu | 559.85 | 100995 |
| | Ibrahim | 517.91 | 93787 |
| FILE 3 (124 KB) | proposed scheme | 459.07 | 99114 |
| | J. Howlader and S. Basu | 799.79 | 111095 |
| | Ibrahim | 739.86 | 103166 |
| FILE 4 (235 KB) | proposed scheme | 655.82 | 109025 |
| | J. Howlader and S. Basu | 1142.56 | 122204 |
| | Ibrahim | 1056.96 | 113482 |
| FILE 5 (435 KB) | proposed scheme | 936.88 | 119927 |
| | J. Howlader and S. Basu | 1632.22 | 134425 |
| | Ibrahim | 1509.95 | 124830 |

### 8.2.1. Simulation Results

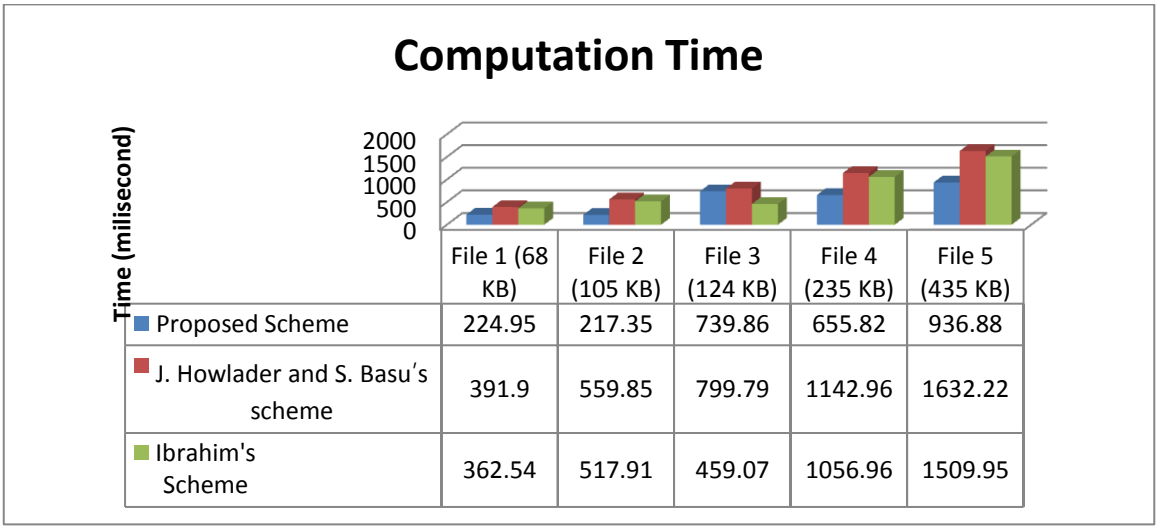
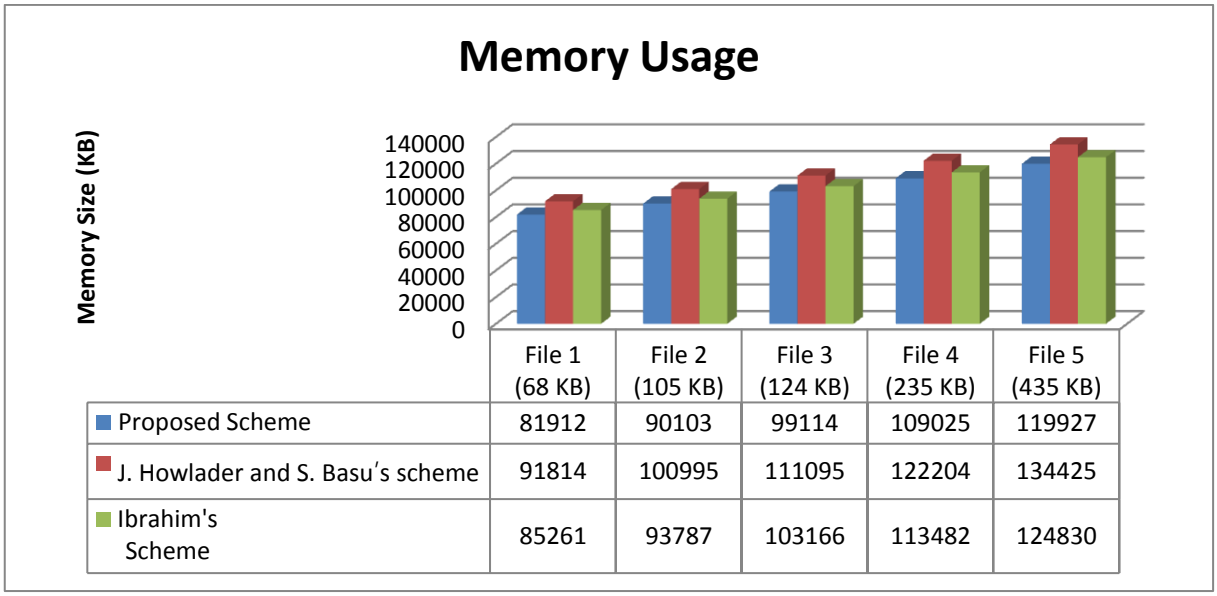



**Fig. 9.** comparison of computation time among Ibrahim's scheme, J. Howlader and S. Basu's scheme and the proposed scheme




| | File 1 (68 KB) | File 2 (105 KB) | File 3 (124 KB) | File 4 (235 KB) | File 5 (435 KB) |
|---|---|---|---|---|---|
| Proposed Scheme | 81912 | 90103 | 99114 | 109025 | 119927 |
| J. Howlader and S. Basu's scheme | 91814 | 100995 | 111095 | 122204 | 134425 |
| Ibrahim's Scheme | 85261 | 93787 | 103166 | 113482 | 124830 |

**Fig. 10.** comparison of memory usage by Ibrahim's scheme, J. Howlader and S. Basu's scheme and the proposed scheme

## 9. Conclusions

We proposed an efficient scheme for sender deniable encryption and it providing to both single-bit and multiple-bit message encryptions. Based on this scheme we prove that our scheme is more secure over that proposed in [6, 9] in the sense of deniability and decipherability. Moreover, our scheme is based on probabilistic encryption model and it enjoys the following properties:

- No pre-shared secret information is required between sender and receiver.
- Achieves a deniability equivalent to the factorization of a large two-prime modulus
- Semantically secure against QRP.
- The decryption process is very fast compared to other related scheme in [6, 9]
- The less overhead in term of the size of the ciphertext.
- No extra computation is required for dishonest opening of y in presence of coercion.
- The proposed scheme has very low power cost compared with other related schems since it consume very low computation time and memory usage.
- A secure internet voting model based on the proposed deniable scheme is originally developed. The internet voting model have the following properties:
  - The model is coercion-resistance.
  - Coercion-resistance is implemented without physical assumptions.

## Acknowledgments

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

## References


[1] S. Goldwasser and S. Micali. "Probabilistic encryption," *Journal of Computer and System Sciences*. vol. 28, issue 2, pp. 270-299, Apr. 1984. Preliminary version in 14th Annual ACM Symposium on Theory of Computing (STOC). Article (CrossRef Link)

[2] K. Sako and J. Kilian. "Receipt-free mix-type voting scheme: A practical solution to the implementation of a voting booth," *Advances in Cryptology — EUROCRYPT '95*, Springer LNCS 921 (1995), 393–403. Article (CrossRef Link)

[3] R. Canetti, U. Feige, O. Goldreich, and M. Naor. "Adaptively secure multi-party computation," *STOC (1996),* pp. 639–648. Article (CrossRef Link)

[4] R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky. " Deniable encryption*," CRYPTO*, pp 90–104. 1997. Article (CrossRef Link)

[5] J. Benaloh and D. Tuinstra. "Uncoercible communication," *Technical Report TR-MCS-94-1*, Clarkson University, March 1994. Article (CrossRef Link)

[6] M. H. Ibrahim: "A method for obtaining deniable Public-Key Encryption," *Trans. On International Journal of Network Security (IJNS),* vol. 8, no. 1. pp 1-9, Jan 2009. Article (CrossRef Link)

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

[8] M. Hirt, and K. Sako, "Efficient receipt-free voting based on homomorphic encryption," *Eurocrypt '00*, pp. 539-556, 2000. Article (CrossRef Link)

[9] J. Howlader and S. Basu: "Sender-Side Public key Deniable Encryption Scheme," in *Proc. of International Conference on Advances in Recent Technologies in Communication and Computing*, pp 9-13, 2009. Article (CrossRef Link)

[10] G. J. Fuchsbauer, "An Introduction to Probabilistic Encryption," *Osjecki Matematicki List 6*, pp 37-44, 2006. Article (CrossRef Link)

[11] A. O'Neill, C. Peikert, and B. Waters. "Bi-deniable public-key encryption," *Manuscript (2010).* Article (CrossRef Link)

[12] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity," *Classes Advances in Computer Science – EUROCRYPT'99,* pp 223-238, Springer-Verlag, 1999. Article (CrossRef Link)


**Tamer Barakat** received his BSc in communications and computers engineering from Helwan University, Cairo; Egypt in 2000. Received his MSc in Cryptography and Network security systems from Helwan University in 2004 and received his PhD in Cryptography and Network security systems from Cairo University in 2008. 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 specially, he is working on the design of efficient and secure cryptographic algorithms, in particular, security in the wireless sensor networks. Other things that interest him are number theory and the investigation of mathematics for designing secure and efficient cryptographic schemes.