Commutative Re-encryption Techniques: Significance and Analysis

Article *in* Information Security Journal A Global Perspective · November 2015

DOI: 10.1080/19393555.2015.1107154

CITATIONS
8

READS
145

3 authors, including:

Nazmul Islam
Dalhousie University
12 PUBLICATIONS **50** CITATIONS
SEE PROFILE

Kazi Md. Rokibul Alam
Khulna University of Engineering and Technology
56 PUBLICATIONS **443** CITATIONS
SEE PROFILE

Information Security Journal: A Global Perspective, Vol. 24:4-6, pp.185-193

Commutative Re-encryption Techniques: Significance and Analysis

Nazmul Islam[1] , Kazi Md. Rokibul Alam[2], and Shaikh Shiam Rahman[2]
[1]Institute of Information and Communication Technology,
[2]Department of Computer Science and Engineering,
Khulna University of Engineering & Technology, Khulna,
Bangladesh.

Abstract—**Commutative re-encryption techniques are important tools for different applications where the encryption and decryption operations are performed in arbitrary order. These techniques also enable one to transfer information securely in a network without any knowledge of public keys of other parties. The process of message transmission vastly depends on the performance of underlying commutative technique used for cryptographic operations. The aim of this paper is to inquest some recently invented commutative re-encryption techniques at every step of their operation and expose the technique with better performance. The content of the paper is broadly divided into three parts-describing the importance of the commutative re-encryption techniques, presenting some latest techniques along with proposing a modification for making their operations faster and finally analyze every operational steps of both the naive and proposed techniques. The modification is proposed only for experimentally proved slow techniques to faster their operation and a comparison is made at the end to show the effectiveness of the modification in making the operations faster.**

Keywords: RSA, ElGamal, re-encryption, commutative cryptography.

I. INTRODUCTION

The commutative re-encryption techniques facilitate multiple parties with individual and repeated encryption of their messages while running a cryptographic application [14][15][16]. The techniques also assist the involved parties to decrypt particularly and repeatedly in an arbitrary order *i.e.* the order of encryptions and decryptions may be irrespective. To carry out these techniques, an existing underlying cryptosystem e.g. RSA or ElGamal is often used and parties involved in the operation keep their encryption and decryption keys secret. Otherwise anyone will be able to calculate the decryption keys of other parties [17].

In spite of much hassles and cumbersome procedures, it is easy to establish the fairness when applications like lottery system, voting [3][5], gambling [4] etc are executed in their conventional ways. Though the implementation of electronic devices and cryptographic techniques with proof mechanisms [7] provide convenience, it becomes vital to ensure the fairness [13]. In commutative re-encryption technique a message m_j is re-encrypted under a sequence of keys K_1, K_2, K_3, ..., K_n denoted as $C = E(K_3, ... E(K_3, E(K_2, E(K_1, m_j))) \ldots)$ and must be decrypted under the corresponding keys K_1^{-1}, K_2^{-1}, K_3^{-1}, ..., K_n^{-1} while the order may be arbitrary *i.e.* $m_j = D(K_j, \ldots D(K_2, D(K_3, D(K_1, C)))\ldots)$. Secret sharing based threshold cryptosystem [8][18] can be employed to execute an application where multiple key owners are not mutually independent. But it usually requires a trusted third party, hence not always preferred by many researchers. Electronic voting system and computerized lottery system are two cryptographic applications that are described hare to explain the demand of commutative re-encryption techniques.

The outline of the paper is as follows. Section II describes cryptographic applications which demand commutative re-encryption for their execution. Section III illustrates three recently invented commutative re-encryption techniques. Section IV proposes a modification to boost up the time efficiency of the techniques that are proved slower experimentally. Section V presents the experimental results and analyzes the reason behind the result followed by a comparison among the original and proposed techniques on the basis of the results. Finally Section VI concludes the paper with future research directions.

II. APPLICATIONS

A. An Electronic Voting System

The electronic voting system [7], considered in this section, consists of N voters V_j (j = 1,---,N), V_j's candidate choice in the election *i.e.* her vote v_j, P (at least 2) mutually independent election authorities (*EA*s), and public bulletin board (BB) as shown in Figure 1. The BB consists of 2 panels, *i.e.* the *VotingPanel* and the *TallyingPanel*, and both of them have memory sections that can be read by anyone at any time.

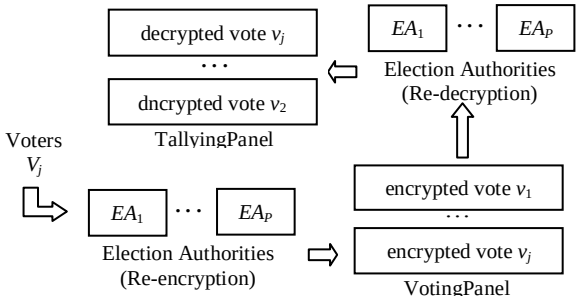


Figure 1: An Electronic Voting System.

Here, voter V_j puts her vote v_j which is re-encrypted by individual *EA*s at her designated location in the *VotingPanel*. Therefore no one except voter V_j herself can know the vote v_j. Later, *EA*s perform individual decryption and shuffling of these votes and move them to randomly selected sections in the *TallyingPanel*. For this system the order of decryptions should be irrespective to the order of encryptions to conceal the

linkages between voters and their votes. Figure 1 depicts the procedure of this system.

The problem associated with this system is that, voters cannot confirm whether the *TallyingPanel* includes all votes of the *VotingPanel* because each voter V_j knows only her vote v_j [19]. Namely, although V_j can find her v_j in the *TallyingPanel*, she cannot corroborate the correspondence between the content of v_j in the *TallyingPanel* with that in the *VotingPanel*. As a consequence, *EA*s can change, add or delete votes put in the *VotingPanel* without being detected. For an example, the last authority EA_p can decrypt votes re-encrypted and shuffled by multiple *EA*s into his favourite vote v_j regardless of the final decryption results as other entities do not know the decryption key of EA_p. Therefore a proof mechanism [20][21], which ensures that the votes disclosed in the *TallyingPanel* include only and all votes of the *VotingPanel*, is necessary. This kind of proof mechanism is usually developed based on ZKP, however, complicated behaviors of ZKP [12] make mechanisms inefficient or impractical. The proof mechanism proposed in [7] is an efficient and practical alteration of ZKP which exploits re-signing technique and attaches secret number to individual V_j.

In the proof mechanism proposed in [7] each voter V_j obtains signatures of authorities $\{EA_1,---,EA_p\}$ on her vote v_j as follows. Firstly, V_j asks multiple authorities $\{EA_1,---,EA_p\}$ in the encryption stage to re-encrypt v_j by using their encryption keys while attaching secret unique number C_{Cj} to it, namely, by using encryption key K_1, EA_1 encrypts $\{v_j, C_{Cj}\}$ into $E(K_1, \{v_j, C_{Cj}\})$, and EA_2 encrypts $E(K_1, \{v_j, C_{Cj}\})$ to $E(K_2, E(K_1, \{v_j, C_{Cj}\}))$ by EA_2's encryption key K_2; and by continuing this process, re-encrypted form of $\{v_j, C_{Cj}\}$ becomes $E(K_*, (\{v_j, C_{Cj}\})$, where, $E(K_i, x)$ represents EA_i's encryption of x by his encryption key K_i, and $E_*(\{v_j, C_{Cj}\})$ is used to represent $E(K_P, E(K_{P-1}, ---E(K_1, \{v_j, C_{Cj}\})))$. After $\{v_j, C_{Cj}\}$'s re-encryption is completed, authorities $\{EA_1,---,EA_P\}$ sign on $E_*(\{v_j, C_{Cj}\})$, and the final signed form becomes $S_*(E_*(\{v_j, C_{Cj}\}))$, where, d_j is the signing key of EA_i, $S(d_j, x)$ is the signature of EA_i on x and $S_*(E_*(\{v_j, C_{Cj}\}))$ represents re-signing result $S(d_P, S(d_{P-1},---S(d_1, E_*(\{v_j, C_{Cj}\})))$. Then, individual authorities in the decryption stage re-decrypt $\{S_*(E_*(\{v_1, C_{C1}\})), S_*(E_*(\{v_S, C_{C2}\})), ---, S_*(E_*(\{v_Z, C_N\}))\}$ into $\{S_*(\{v_1, C_{C1}\}), S_*(\{v_S, C_{C2}\}), ---, S_*(\{v_Z, C_{CN}\})\}$ while shuffling interim decryption results. Therefore, V_j can conceal her $\{v_j, C_{Cj}\}$ from others, unless all authorities conspire. Also signatures of multiple *EA*s on $\{v_j, C_{Cj}\}$ ensure that $\{v_j, C_{Cj}\}$ is moved from the *VotingPanel*, because no single entity can forge signatures of all authorities on $\{v_j, C_{Cj}\}$ and C_{Cj} is unique in the system.

To execute the above re-encryption procedure, the commutative property is mandatory. Re-signed form $S_*(E_*(\{v_j, C_{Cj}\}))$ of $\{v_j, C_{Cj}\}$ must be generated by calculating $E_*(\{v_j, C_{Cj}\})$ from $\{v_j, C_{Cj}\}$ by encryption keys $\{K_1, K_2,---,K_P\}$ and calculating $S_*(E_*(\{v_j, C_{Cj}\}))$ from $E_*(\{v_j, C_{Cj}\})$ by signing keys $\{d_1, d_2,---,d_P\}$ in this order, and conversely the validity of $S_*(\{v_j, C_{Cj}\})$ must be checked firstly by decrypting $S_*(E_*(\{v_j, C_{Cj}\}))$ into $S_*(\{v_j, C_{Cj}\})$ by decryption keys $\{K_1^{-1}, K_2^{-1},---,K_P^{-1}\}$ and then calculating $\{v_j, C_{Cj}\}$ from $S_*(\{v_j, C_{Cj}\})$ by verification keys $\{d_1^{-1}, d_2^{-1},---,d_P^{-1}\}$. When $\{EA_1,---,EA_P\}$ sign on votes before encrypting them, EA_P can know the signed forms of votes and can forge the signed forms of his favorite candidate's votes to put them in the *TallyingPanel* regardless of decryption results. Alternatively, when $S_*(E_*(\{v_j, C_{Cj}\}))$ are verified (*i.e.* decrypted by the verification keys) before calculating $S_*(\{v_j, C_{Cj}\})$, EA_P can decrypt re-encrypted vote $E_*(\{v_j, C_{Cj}\})$ in his favorite ways regardless of $E_*(\{v_j, C_{Cj}\})$ because no one know the re-encrypted form of v_j and no one except EA_P knows decryption key K_P^{-1}. Hence, the encryption and signing techniques must be commutative.

B. A Computerized Lottery System

A computerized lottery system [2] consists of N participants C_j (j = 1,---,N) associated with their names $\{N_1, N_2,---,N_N\}$, P (at least 2) mutually independent organizing authorities (*OA*s), and public BB, as shown in Figure 2. The BB consists of 2 panels, *i.e.* the Applicant and the Prize panels, and both are sets of memory sections that can be read by anyone at any time. Then the participant C_j puts her name N_j at the designated section in the *ApplicantPanel*, and authorities *OA*s shuffle these names and moves them to randomly selected sections in the *PrizePanel*. Here, prizes are assigned to selected sections in the *PrizePanel* in advance. Therefore prize winners are the participants whose names at the *ApplicantPanel* are placed at prize allocated sections in the *PrizePanel*.

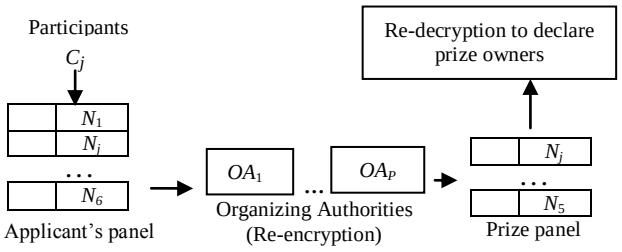


Figure 2: A Computerized Lottery System.

Participant C_j who could not get prize in the above computerized lottery may doubt that *OA*s did not move the names from the *ApplicantPanel* to the *PrizePanel* in a fair way. When authorities *OA*s are conspiring with some participant C_j, it is possible for *OA*s to put N_j, a name of participant C_j, to the section in the *PrizePanel* where the prize is allocated. This kind of dishonesties can be prevented by mechanisms used in mixnet to conceal participants' messages [6] while constructing *OA*s as a set of multiple independent authorities $\{OA_1,---,OA_P\}$. where each OA_i in the encryption stage encrypts names in the *ApplicantPanel* by his encryption key and shuffles the results to be encrypted by OA_{i+1}. Finally the last authority OA_P puts the re-encrypted results at randomly selected sections in the *PrizePanel*. The prize owners are revealed by decrypting re-encrypted names in the *PrizePanel* after all names in the *ApplicantPanel* are moved to the *PrizePanel*. As long as all *OA*s do not conspire, each OA_i cannot know decryption keys of other *OA*s before they are disclosed; therefore *OA*s cannot place names of participants that they are conspiring with at the prize allocated sections in the *PrizePanel*. Finally, because decryption keys of all authorities are disclosed, no authority can decrypt re-encrypted names in the *PrizePanel* dishonestly.

Here, the re-encryption technique used to conduct the computerized lottery system must be commutative. Moreover the probabilistic property is also desired. It must be noticed that in this system the name N_j of a participant C_j in the *ApplicantPanel* is unique and also all entities know N_j from the beginning. Hence, there is no inconvenience to disclose decryption keys of the authorities after all the re-encrypted

names are arranged in the *PrizePanel*, and no extra proof mechanism is necessary to prove the correct behaviors of individual *OA*s. When an authority OA_i encrypts or shuffles names of participants dishonestly, C_j can find the dishonesties about N_j, *i.e.* additions, deletions and modifications of N_j, by checking the number of appearances of N_j disclosed in the *PrizePanel*. C_j can also detect *OA*s' dishonest decryptions of re-encrypted names by decrypting re-encrypted forms in the *PrizePanel* by herself.

III. COMMUTATIVE RE-ENCRYPTION TECHNIQUES

To conduct cryptographic applications as presented in the previous section, a commutative re-encryption technique is essential. This Section illustrates three novice techniques presented in [1], [2] and [9] respectively, which are capable of satisfying the commutative requirement. In all techniques, there is a message owner known as participant C_j and there are P ($P \geq 2$) mutually independent authorities denoted as $\{A_1,---,A_P\}$ and also A_P is considered as a intended receiver of the message of C_j. The techniques presented in [1] and [2] are based on RSA cryptosystem where multiple authorities need to perform their encryption and decryption based on the same modulo arithmetic.

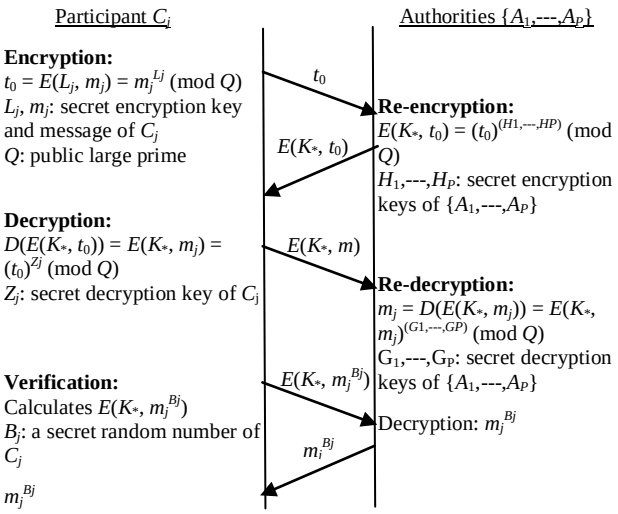


Figure 3: *Khayat's* technique

Figure 3 illustrates the interactions between the participant C_j and authorities in *Khayat*'s technique. Here the outcome of this re-encryption technique is non-probabilistic. Usually in some cryptographic applications, as discussed in Section II, different participants may have the same choice; therefore a probabilistic re-encryption technique is required. If not probabilistic, same choices are always encrypted into the same forms and a participant can know the choices of others who had chosen the same choice even they are encrypted. The technique proposed in [2] overcomes this limitation. The demand of commutative re-encryption of data along with probabilistic property can be implemented by *Tamura et al.'s* technique. Figure 4 illustrates the interactions between the participant C_j and the authorities in *Tamura et al.*'s technique.

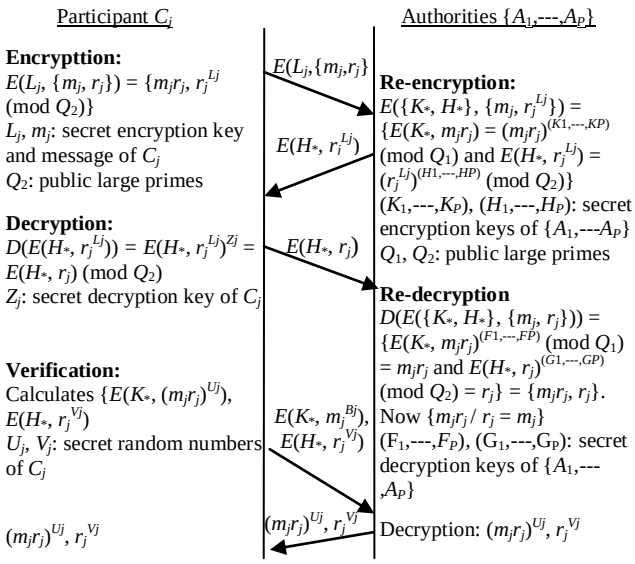


Figure 4: *Tamura et al.*'s technique.

Similar to *Tamura et al.'s* technique, *Wei*'s technique [9] also offers probabilistic feature but it adopts ElGamal cryptosystem rather than RSA. The interaction between the participant C_j and the authorities in *Wei's* technique is illustrated in figure 5.

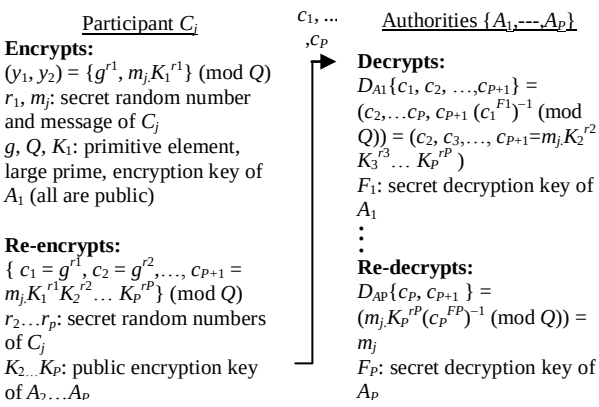


Figure 5 : *Wei's* technique.

IV. THE PROPOSED MODIFICATION

It has been experimntally proved that *wei*'s technique requires less time to complete its operation than the other two which is shown graphically and described analytically in next section. Hence, a modification is presented in order to make RSA based commutative re-encryption techniques efficient in terms of time.

A. Khayat's Technique

The modification of *Khayat's* technique consists of four phases: (1) Setup, (2) Encryption, (3) Decryption, and (4) Verification. They proceed as following way.

Setup; A message $m_j \varepsilon Z_Q$ is owned by the participant C_j and she, though not mandatory, has a secret encryption and decryption key pair (L_j, Z_j) such that $(L_j, Z_j) \varepsilon Z_Q$. Each authority A_i also has his own secret encryption and decryption key pair (H_i, G_i) such that $(H_i, G_i) \varepsilon Z_Q$. H_i is disclosed only in multiplied form(H^*) with encryption keys of other authorities *i.e.* $H^* = \prod_{i=1}^{P} H_i$ as, if key H_i is compromised, it is easy for A_j

to calculate A_i's decryption key G_i from the relation $H_iG_i \pmod{(Q-1)} = H_jG_j \pmod{(Q-1)}$. Here for any integer u, it satisfies the relation $u^{HiGi} \pmod{Q} = u \pmod{Q}$. Moreover, greatest common divisor (GCD) (L_j, $Q-1$) = 1 and GCD (H_i, $Q-1$) = 1; and $\{L_jZ_j = 1 \pmod{Q-1}$, and $H_iG_i = 1 \pmod{Q-1}\}$. Note that, it is computationally infeasible to break H^* down into individual keys H_i, hence they remain secrets while H^* is publicly disclosed.

Encryption; In this phase, the message m_j is encrypted by the participant using H^* as $E_j = (m_j)^{H^*} \pmod{Q}$.

Decryption; The ciphertext E_j is re-decrypted by the authorities in decryption phase. All $A_i \in \{A_1,---,A_P\}$ re-decrypt the ciphertext E_j *i.e.* calculate $(E_j)^{(G1,---,GP)} = m_j$ by their decryption keys $G_1,---,G_P$. It is to be noted that the specific decryption key can be applied in arbitrary order to retrieve the message m_j; therefore the technique is commutative.

Verification; To verify the correct operations of the authorities, C_j asks $\{A_1,---,A_P\}$ to decrypt $(m_j^{Bj})^{H^*}$ where B_j is a secret random number of C_j. If conspired, authorities cannot decrypt $(m_j^{Bj})^{H^*}$ into m_j^{Bj} as they do not know B_j. Therefore, although H_i of each A_i is secret, C_j can detect the dishonesty of authoritis. However, this phase is optional for C_j.

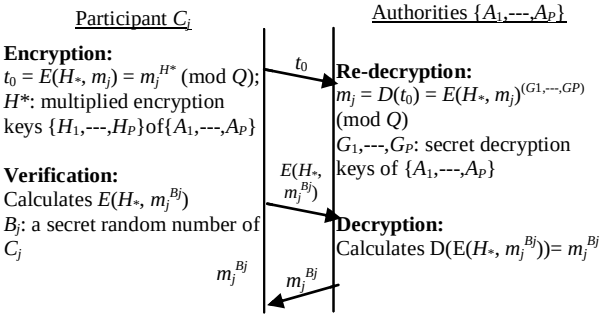


Figure 6: Modification of *Khayat*'s technique.

B. *Tamura et al.'s Technique*

Similar to modified *Khayat's* technique, the modification of *Tamura et al.*'s technique also uses modular exponentiation based on RSA cryptosystem. It considers two large primes Q_1 and Q_2 which are publicly known to all. The technique also consists of four phases: (1) Setup, (2) Encryption, (3) Decryption, and (4) Verification. Their roles are described below.

Setup: A participant C_j has a secret encryption and decryption key pair $\{L_j, Z_j\}$ though it is discretionary for this modified technique. Each authority A_i also has two sets of encryption and decryption key pairs: $\{K_i, F_i\}$ for encryption on modulo Q_1 arithmetic and $\{H_i, G_i\}$ for decryption on modulo Q_2 arithmetic. Similar to *Khayat's* technique, the key pairs are kept as A_i's secrets in order to enable each A_i to securely use his key pairs in an environment where multiple authorities share the same modulo arithmetic. H_i and K_i are disclosed only in multiplied form(H^* and K^*) with respected encryption keys of other authorities *i.e.* $H^* = \prod_{i=1}^{P} H_i$ and $K^* = \prod_{i=1}^{P} K_i$ where P is total number of authorities. When key K_i or H_i is disclosed, it is easy for anyone to calculate A_i's decryption key F_i from the relation $K_iF_i \pmod{(Q_1-1)} = K_jF_j \pmod{(Q_1-1)}$ or $H_iG_i \pmod{(Q_2-1)} = H_jG_j \pmod{(Q_2-1)}$ respectively. Here for any two integers u and w, it satisfies the relation $u^{KiFi} \pmod{Q_1} = u \pmod{Q_1}$, $w^{HiGi} \pmod{Q_2} = w \pmod{Q_2}$ and $w^{LjZj} \pmod{Q_2} = w \pmod{Q_2}$.

Encryption: In this phase, the participant C_j generates a secret random number r_j and encrypts her message m_j by calculating $E(\{K^*, H^*\}, \{m_j, r_j\}) = \{E(K^*, m_jr_j) = (m_jr_j)^{K^*} \pmod{Q_1}$ and $E(H^*, r_j) = (r_j)^{H^*} \pmod{Q_2}\}$ *i.e.* the encrypted form consists of a data part $E(K_i, m_jr_j)$ and a randomization part $E(H_i, r_j)$. Here authorities cannot calculate m_j from m_jr_j and r_j^{Lj}, because r_j is the secret of C_j and the calculation of r_j from r_j^{Lj} is a discrete logarithm problem.

Decryption: In this phase, the ciphertext $E(\{K^*, H^*\}, \{m_j, r_j\})$ is re-decrypted by the authorities $\{A_1,---,A_P\}$ into (m_jr_j, r_j) by calculating $E(K^*, m_jr_j)^{(F1,---,FP)} = (m_jr_j)^{K^*(F1,---,FP)} \pmod{Q_1} = m_jr_j$ and $E(H^*, r_j)^{(G1,---,GP)} = r_j^{H^*(G1,---,GP)} \pmod{Q_2} = r_j$ using their decryption keys $F_1,---,F_P$ and $G_1,---,G_P$; and finally r_j is used to retrive m from m_jr_j

Verification: For the confirmation of correct re-encryption of the authorities $\{A_1,---,A_P\}$, C_j asks them to decrypt $E(K^*, (m_jr_j)^{Uj})$ and $E(H^*, r_j^{Vj})$, where $\{U_j, V_j\}$ are secret random numbers of C_j. Dishonest authorities fail to complete the operation correctly as they do not know U_j, V_j, m_jr_j or r_j. Therefore, although K_i and H_i of each A_i are secret, C_j can confirm the correctness of the operations. However, this phase is optional for C_j. It is apparent that this re-encryption technique is probabilistic and commutative.

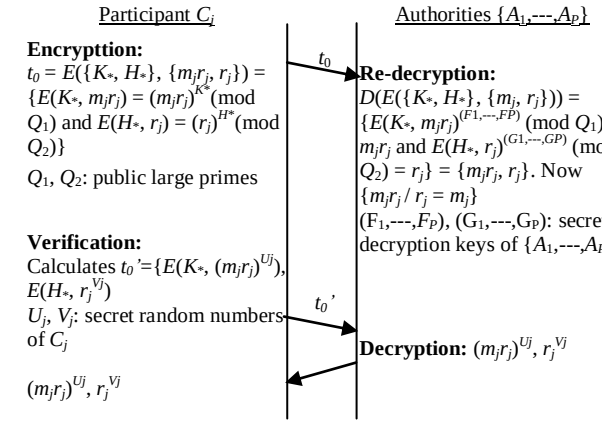


Figure 8: Modified *Tamura et al.*'s technique.

The existing techniques of *Khayat* and *Tamura et al.* facilitate participant C_j to perform information exchange secretly when receiver's or authority's public keys are not achievable. But the major limitation of these techniques is considerable time consumption. Though the proposed modification increases complexity of the naive techniques, it becomes successful in achieving the aim to make RSA based techniques faster *i.e.* making *Khayat* and *Tamura et al.* techniques time efficient.

V. Experimental Analysis

This section evaluates the time requirements of encryption, decryption and verification phases of the chosen re-encryption techniques based on the key length of the underlying cryptosystem.

A. *Experimental Setup*

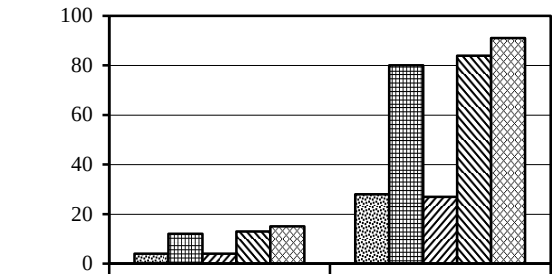
The experiments are executed in an environment that consists of 32 bit windows machine (windows 7), Core i3 3.60 GHz CPU with 2048MB of RAM and GMP [11] languages are used for the coding purpose. The techniques were

evaluated keeping the participant's, authority's number and message length fixed at 1, 3 and 40 digits and the key length of 1024 bit and 2048 bit. For RSA like encryption key generation, Euclidian algorithm [12] is an efficient method and is used to compute the GCD of 2 integers. Extended Euclidian algorithm is also used to find the modular multiplicative inverse of encryption key *i.e.* to generate the decryption key. During simulation, the travelling time of messages for both the sender and the receiver are assumed to be negligible *i.e.* all computation time do not include the communication time. Also the time requirement for encryption and decryption key or secret random number generation involved in the techniques and operations that are not related to cryptography are not considered.

B. Experimental Results

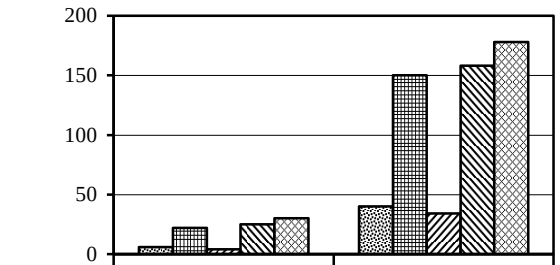
The time requirements (ms) for different steps of the chosen re-encryption techniques using GMP are shown in Figure 9, 9, 10. Whereas, time consumption of *Khayat's* and *Tamura et al.'s* techniques with proposed modification are shown in Figure 11 and 12. Tables of the figure contain short form of the step names– Participant's encryption (P_Enc), Participant's re-encryption (P_rEnc), Participant's decryption (P_Dec), Participant's verification (P_V), Authorities' re-encryption (A_rEnc) and Authorities' re-decryption (A_rDec).

In *Khayat's* and *Tamura et al.'s* techniques as authorities' encryption keys are kept secret, the participant may wish to verify the correctness of their encryptions. In contrast, in case of *Wei's* technique as the encryption keys are public; therefore this verification is not mandatory.



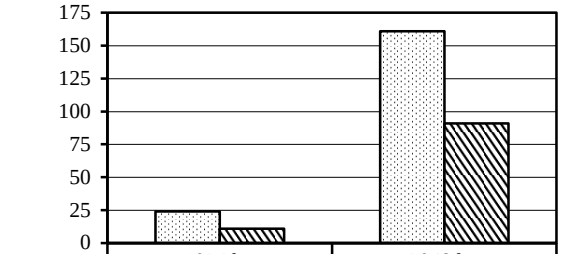
	1024 bit	2048 bit
P_Enc	4	28
A_rEnc	12	80
P_Dec	4	27
A_rDec	13	84
P_V	15	91

Figure 10: Khayat's technique.



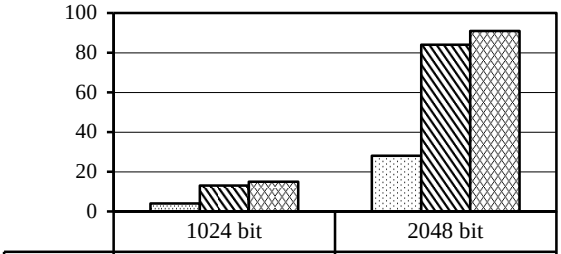
	1024 bit	2048 bit
P_Enc	6	40
A_rEnc	22	150
P_Dec	4	34
A_rDec	25	158
P_V	30	178

Figure 11: *Tamura et al.*'s technique.



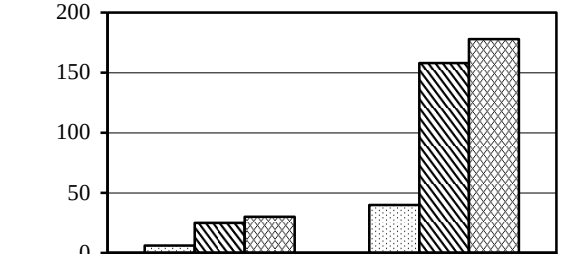
	1024 bit	2048 bit
P_rEnc	24	161
A_rDec	11	91

Figure 12: *Wei's* technique.



	1024 bit	2048 bit
P_rEnc	4	28
A_rDec	13	84
P_V	15	91

Figure 13: Modified *Khayat's* technique.



	1024 bit	2048 bit
P_rEnc	6	40
A_rDec	25	158
P_V	30	178

Figure 14: Modified *Tamura et al's* technique.

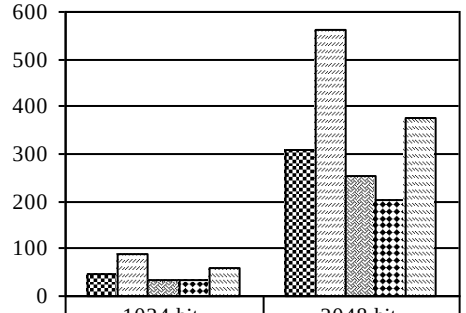
C. Comparison and Discussion

For *Khayat's* technique, all operations have modular exponentiation. Therefore, time requirements of the participant's encryption and decryption are equal. Similarly, it is also same in case of authorities' re-encryption and re-decryption and which are greater than that of the participant's. Here, the time requirements are proportional to the number of authorities involved in the operation. Finally, the verification requires more time than re-encryption or re-decryption because it consists of both participant's encryption and authorities' re-decryption operations.

For *Tamura et al.'s* technique, the participant's encryption includes multiplication and modular exponentiation whereas the decryption needs only modular exponentiation and hence the time requirements are not equal. Again, authorities' re-encryption and re-decryption contain equal number of modular exponentiations based on two different modulo, therefore time requirements are also same. The time requirement of verification step is higher than re-encryption or re-decryption because it consists of both participant's encryption and authorities' re-decryption based on two different modulo.

Unlike the previous two, modified *Khayat*'s and *Tamura et al.*'s technique consumes far less time because of reduction of steps. For *Wei's* technique, the time requirement of participant's re-encryption is higher as it performs more operations than authorities' re-decryption.

Among the techniques, while the total time requirements are compared (Figure 13), *Tamura et al.'s* requires the utmost time whereas *Wei's* technique requires the least. The reason is that almost all operations of *Tamura et al.'s* technique are based on two different modulo while the operations of other two techniques are based on a single modulo. For *Wei's* technique, it requires only participant's re-encryption and authorities' re-decryption which gives it time efficiency. However, modified *Khayat*'s technique demands less time than *Wei*'s because of reduction of operations which makes this method preferable in applications of commutative re-encryption. For all the techniques, time requirement is proportional to the key length of the underlying cryptosystem.



	1024 bit	2048 bit
Khayat	48	310
Tamura et al.	87	560
Wei	35	252
Khayat (Modified)	32	203
Tamura et al.	61	376

Figure 15; Comparison among the techniques.

VI. Conclusion and Future Works

This paper emphasizes the significance of commutative re-encryption techniques on many cryptographic applications and presents an empirical analysis of both RSA and ElGamal based commutative re-encryption techniques. Unlike the usual cryptography where the order of encryption and decryption is symmetric, commutative re-encryption flex the order of encryptions and decryption. Therefore enables multiple mutually independent parties to decrypt the re-encrypted ciphertext in an arbitrary order which is essential to conduct many cryptographic applications.

A future plan of research is to improve the proposed modification to increase the overall efficiency. Though the techniques shown here are time efficient, they amplify the complexity. Hence, to find out a way of reducing complexity and maintaining time efficiency at the same time is a good research issue. Moreover, naïve RSA based techniques flex participants to perform operation without any knowledge of other parties. Research initiatives to retain the property for modified techniques can also be taken.

References

[1] S. H. Khayat, "Using Commutative Encryption to Share a Secret," IACR Cryptology ePrint Archive 2008: 356, 2008.

[2] S. Tamura, A. K. Md. Rokibul, and H. A. Haddad, "A Probabilistic and Commutative Re-encryption Scheme," in Proc. Asis Simulation Conference 2009, ID 032, Ritsumeikan University, Shiga, Japan, October 7 –9, 2009.

[3] Y.A.R. Peter, S. Steve, T. Vanessa, "End-to-End Verifiability in Voting Systems, from Theory to Practice", IEEE Security & Privacy, vol.13, no. 3, pp. 59-62, 2015.

[4] S. J. Aboud, M. A. AL-Fayoumi, "An Efficient Internet Bingo Scheme" International Journal of Advanced Research in Computer Science and Software Engineering, Vol 4, Issue 1, pp. 456-460, , 2014.

[5] B. N. Jonathan, F. Niko, L. Morgan, R. Ben, R. Alon, T. S. Amnon and W. Douglas, "A New Implementation of a Dual (Paper and Cryptographic) Voting System," In 5th International Conference on Electronic Voting (EVOTE 2012). Lochau/Bregenz, Austria, 2012.

[6] N. Islam, A. K. M. Rokibul, and A. Rahman,"The effectiveness of mixnets – an empirical study," Elsevier Computer Fraud & Security, Vol. 2013, Issue 12, pp. 9–14, December, 2013.

[7] A. K. M. Rokibul, S. Tamura, S. Taniguchi, and T. Yanase, "An Anonymous Voting Scheme based on Confirmation Numbers," IEEJ Transactions EIS, Vol. 130, No. 11, pp. 2065–2073, 2010.

[8] H. Chunqiang, L. Xiaofeng and C. Xiuzhen,"Verifiable multi-secret sharing based on LFSR sequences," Elsevier Theoretical Computer Science, Vol. 445, No. 2012, pp. 52–62, 2012.

[9] D. Wei, "Commutative-like Encryption: A New Characterization of ElGamal," The Computing Research Repository, Vol. 1011, 2010.

[10] G. Rossum. Python Software Foundation. Software available at http://python.org/ on April, 2014.

[11] T. Granlund. GNU Multiple Precision Arithmetic Library (GMP). Software available at http://gmplib.org/ on April, 2014.

[12] F. Kerschbaum, "Public-Key Encrypted Bloom Filters with Applications to Supply Chain Integrity", in Proc. of the 25th IFIP WG 11.3 Conference on Data and Applications Security, 2011.

[13] A Sodiya, S Onashoga and D I Adelani, "Secure EVoting Architecture", in Proc. of Eighth International Conference on Information Technology: New Generations, IEEE Computer Society, pp. 342–347, 2011.

[14] N. Saputro, K. Akkaya, "Performance evaluation of smart grid data aggregation via homomorphic encryption" in IEEE Wireless Communications and Networking Conference (WCNC), pp. 2945–2950, Shanghai, 2012.

[15] G. Sujitha., T. Rajeswaran, R. Thiagarajan, K. Vidya, S. S. Mercy, "Preserving Privacy of Cloud Data Using Homomorphic Encryption in MapReduce", Intl. J. Hybrid Information Technology, Vol.7 No.3, pp. 363–376, 2014.

[16] D.J. Guan, C.Y. Tsai, E.S. Zhuang, "Detect Zero by Using Symmetric Homomorphic Encryption", in IEEE 8th Asia Joint Conference on Information Security (Asia JCIS), pp. 1–7, Seoul, 2013.

[17] X Wang, D Zhao, "A special attack on the asymmetric cryptosystem based on phase-truncated Fourier transforms", in Elsevier Optic Communication, Volume 285, No. 6, pp. 1078–1081, 2012.
[18] H. Lein, L. Changlu, "Strong (n, t, n) verifiable secret sharing scheme", in Elsevier Information Sciences, Vol. 180, No. 16, pp. 3059–3064, 2010.
[19] M. Bishop, D. Wagner, "Risks of e-voting", Communications of the ACM, Vol: 50, No. 11, pp. 120, 2008.
[20] P. L. Cayrel, P. Véron, A. Smey, "A Zero-Knowledge Identification Scheme Based on the q-ary Syndrome Decoding Problem", in Springe Selected Areas in Cryptography, Vol. 6544, pp. 171–186, 2011.
[21] C. Melissa, K. Markulf, L. Anna and M. Sarah ,"Malleable Proof Systems and Applications", Advances in Cryptology – EUROCRYPT 2012, Vol 7237, pp. 281-300, 2012.

Mr. Nazmul Islam is currently a lecturer at Institute of Information and Communication Technology of Khulna University of Engineering & Technology. He has completed his B.Sc. degree in Computer Science and Engineering from the same University in 2013. His research interests include pervasive computing and information security.

Dr. Kazi Md. Rokibul Alam is currently a professor at Dept. of Computer Science and Engineering of Khulna University of Engineering & Technology. He received Dr. (Eng.) degree in Intelligent Information Systems from University of Fukui, Japan and M.Sc. degree in Computer Science and Engineering from Bangladesh University of Engineering & Technology in 2010 and 2004 respectively. His research interests include applied cryptography, information security and machine learning.

Shaikh Shiam Rahman has completed his B.Sc. degree in Computer Science and Engineering from Khulna University of Engineering & Technology. Now he is serving as Solution Delivery Engineer in Systems Solutions and Development Technologies Limited (SSD-TECH).