Computation And Communication Efficient Chinese Remainder Theorem Based Multi-Party Key Generation Using Modified RSA

Arjun Singh Rawat and Maroti Deshmukh

1 Introduction

In day-to-day life people are dealing with many remote applications related to two participants or multi-participant group conversation. These applications encapsulated with security advancement makes people conversationally reliable with security, but still, these applications are not reliable in terms of communication and computation cost, thus, many researchers also working on these. Inefficient communication and computation cost create congestion on a network and delay on a sender and receiver side as well. There are many key agreement and key generation schemes, which provide not only security but also efficient communication and computation. The key agreement protocol is a cryptographic scheme [1] that follows the process of exchanging public keys with each other by hiding the private key over an unsecured channel, after exchanging the public information the participants agree on the common session key and perform encryption and decryption process [2–4]. The process of the key exchange protocol based on two participants is shown in Fig. 1, where User-A shares its public key to User-B, in the same way User-B shares its public key to User-A. After exchanging the information both agree on a common session key for further process of encryption and decryption.

One of the common public-key cryptography based key agreement protocols is Diffie–Hellman key exchange, developed in 1976 [5]. This approach is not cost-effective in terms of computation cost taking huge time, but gives better security with the advancement of discrete logarithm problems, but with large key size, for

A. S. Rawat (✉) · M. Deshmukh
National Institute of Technology, Srinagar, Uttarakhand, India
e-mail: arjunsinghrawat005@gmail.com

M. Deshmukh
e-mail: marotideshmukh@nituk.ac.in

P. Stănică et al. (eds.), *Security and Privacy*, Lecture Notes in Electrical Engineering 744,
https://doi.org/10.1007/978-981-33-6781-4_3

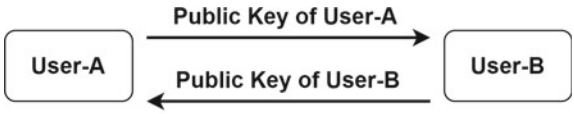
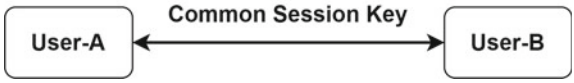


Fig. 1 Key exchange protocol



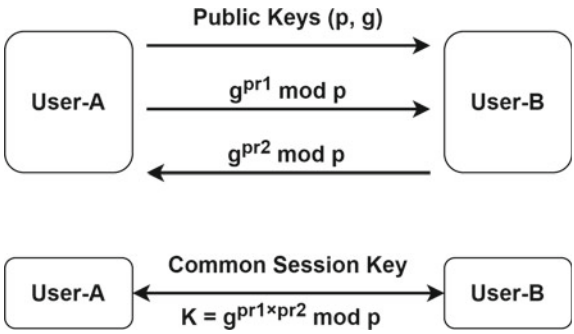


Fig. 2 Diffie–Hellman protocol

breaking that algorithms at least requiring of polynomial time. The Diffie–Hellman key exchange is shown in Fig. 1, the process has the following steps:

1. User-A and User-B agree upon public keys as a large prime number p and generator or the primitive root g less than p.
2. Both User-A and User-B, respectively, generate the random private keys as pr_1 and pr_2 less than large prime p.
3. User-A shares the public key $(g^{pr_1} \bmod p)$ to User-B.
4. User-B shares the public key $(g^{pr_2} \bmod p)$ to User-A.
5. After sharing the public keys both users agree upon a common session key $K = (g^{pr_1 \times pr_2} \bmod p)$ (Fig. 2).

According to this approach, to identify the value of the common session key there should be a requirement of polynomial time because of the discrete logarithm problem, by applying the brute force attack there would be a need for exponential time to break. The key generation protocol is the public-key cryptography, the popular key generation protocol is RSA [6], widely used as secure data transmission, where the encrypted key is public and decrypted key different from the encrypted key is private. The protocol is shown in Fig. 3, the process in the following steps:

The algorithm steps are as follows for 2 participants:

1. There are two participants Q and R.
2. Let participant Q generates the two large random prime numbers n_1 and n_1, where their product evaluation as, $n = n_1 \times n_2$.

3. Now participant Q calculates the $\phi(n) = (n_1 - 1) \times (n_2 - 1)$.
4. Participant Q chooses the random public key e, where $1 < e < \phi(n)$ and gcd $(e, \phi(n)) = 1$.
5. Participant Q chooses the random private key d, where $1 < d < \phi(n)$ and $d \times e \equiv 1 \bmod \phi(n)$.
6. Participant R, sends the message m in encrypted form to the participant Q as $c = (m^e \bmod n)$, c is represented as cipher text.
7. Participant Q decrypts the cipher text as $(c^d \bmod n)$ and retrieves the original message m.

There are many algorithms to speed up the process of the RSA algorithm, one of them is the Chinese remainder theorem, giving the unique solution to simultaneous congruences related to coprime modulo. The process of the Chinese remainder theorem as follows:

1. Let $p_1, \ldots, p_n$, are large prime numbers, be pairwise coprime ($\gcd(p_i, p_j) = 1$, whenever $i \neq j$) to each other, and $1 \leq i, j \leq n$. The $pu_1, \ldots, pu_n$, are the random integers.
2. System of n equations of simultaneous congruences has a unique solution for $e\ modulo\ P$, where $P = p_1 \times p_2 \times \cdots \times p_n$.

$$e \equiv pu_1 \pmod{p_1}$$

$$\ldots$$

$$e \equiv pu_n \pmod{p_n}$$

3. Defining $a_j = P/p_j$ and $a_j^{'} = a_j^{-1} (\bmod\ p_j)$
4. The solution as follows: $e = \sum_{j=1}^{n} pu_j \times a_j \times a_j^{'} (\bmod\ P)$

The remaining paper as follows. Section 2, representing as a literature review of the group key agreement protocol. Section 3 represents the methodology of proposed work. Section 4 represents the experimental result and complexity analysis. Section 5 represents the conclusion of entire paper work.

2 Related Work

Rawat and Deshmukh [7] discussing the computation and communication efficient group key exchange protocol for low configuration system with minor degradation in security, the computation cost reduced by applying the modular multiplication instead of a modular exponential operation, for reducing the communication cost the approach used the divide and conquer approach, but still had a major issue in security.

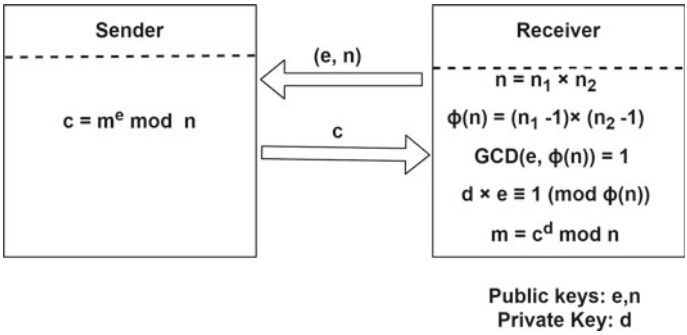


Fig. 3 Key generation as RSA

Mandal and Mohanty [8] discussed the multi-participant group key exchange protocol by achieving the perfect forward secrecy, the approach based on central authority as trusted third-party authority for managing the group operations. Michael et al. [9] discussed the improved Diffie–Hellman group key establishment protocol as well as comparative analysis on the basis of multiplication cost, exponentiation cost, and time complexity. Naresh and Murthy [10] discussed the optimized improved group Diffie–Hellman approach by comparing the number of rounds, number of messages, number of exponential operations. Murthy and Naresh [11] discussed the improved Diffie–Hellman key establishment protocol by any time sharing of multiple shared key with the help of case optimized complexity with polynomial time as well as the comparative analysis with Diffie–Hellman group key exchange protocol. Rawat and Deshmukh [12] discussing the common key generation for not only group but also small subgroups by applying the divide and conquer approach based on tree with the help of Elliptic curve Diffie–Hellman as well as proving the better security.

3 Proposed Model

The major problem with the existing approaches was they are less secure and inefficient in terms of computation and communication cost for multiple participants (group). The proposed approach is based on the client and server mechanism, providing secure communication, by applying modified RSA using the Chinese remainder theorem for group communication. The Chinese remainder theorem provides a common key for a group without violating the security in less computation cost and modified RSA providing secure communication for the group. The Chinese remainder theorem and modified RSA based multiple participants are described in two different phases as a common key generation for the group, group communication.

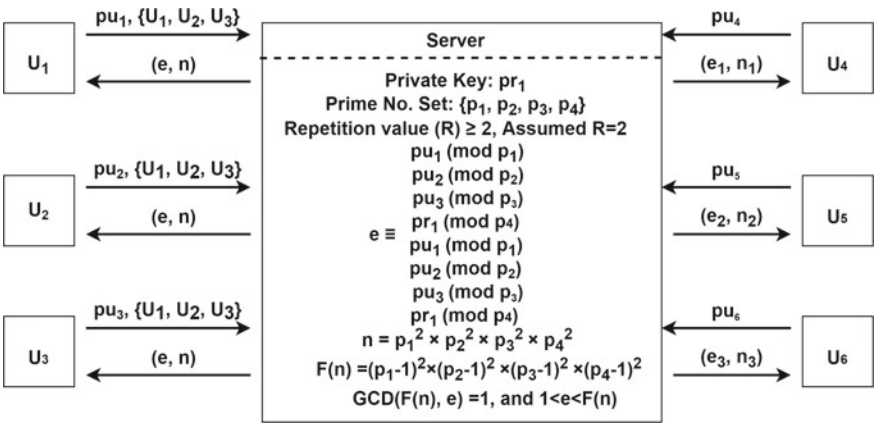


Fig. 4 Generating common key using multi-party modified RSA

3.1 Common Key Generation for the Group

In this phase, the algorithm is illustrated for the three participants as shown in Fig. 4. The common key generation for the group in the following steps:

1. There are six participants $U_1, U_2, U_3, U_4, U_5, U_6$, and a server S, out of which three participants U_1, U_2, U_3 sending the group request to the server S, by performing the multi-party RSA algorithm, U_4, U_5, U_6 are the individual users, performing the simple RSA algorithms for communication.
2. Participants U_1, U_2, U_3 send their public keys pu_1, pu_2, pu_3, respectively, along with group request $\{U_1, U_2, U_3\}$ to the Server S. The pu_4, pu_5, pu_6 are the public keys sent by the individual users U_4, U_5, U_6 to the Server S.
3. Server S separately generates the random private key pr_1 and prime numbers p_1, p_2, p_3, p_4. The server generates as many prime numbers as the number of participants in the group plus server. In the given illustration, the group size is 3, hence 4 prime numbers generated.
4. Server S performs the Chinese remainder theorem as $\mathrm{e} \equiv pu_1 \bmod p_1$, $\mathrm{e} \equiv pu_2 \bmod p_2$, $\mathrm{e} \equiv pu_3 \bmod p_3$, $\mathrm{e} \equiv pr_1 \bmod p_4$.
5. Repetition R, where $R > 1$, is for applying the Chinese remainder theorem repeatedly, assuming $R = 2$ (increased the value of R, reducing the threat of factorizing attack), then $\mathrm{e} \equiv pu_1 \bmod p_1$, $\mathrm{e} \equiv pu_2 \bmod p_2$, $\mathrm{e} \equiv pu_3 \bmod p_3$, $\mathrm{e} \equiv pr_1 \bmod p_4$, $\mathrm{e} \equiv pu_1 \bmod p_1$, $\mathrm{e} \equiv pu_2 \bmod p_2$, $\mathrm{e} \equiv pu_3 \bmod p_3$, $\mathrm{e} \equiv pr_1 \bmod p_4$.
6. Server S, multiplying all random prime numbers, including repeated prime numbers as $n = p_1 \times p_2 \times p_3 \times p_4 \times p_1 \times p_2 \times p_3 \times p_4$.
7. Server S shares the common public-key pair (e, n) to the participants U_1, U_2, U_3. The public-key pairs (e_1, n_1), (e_2, n_2), (e_3, n_3), respectively, are shared with the individual users U_4, U_5, U_6, respectively.

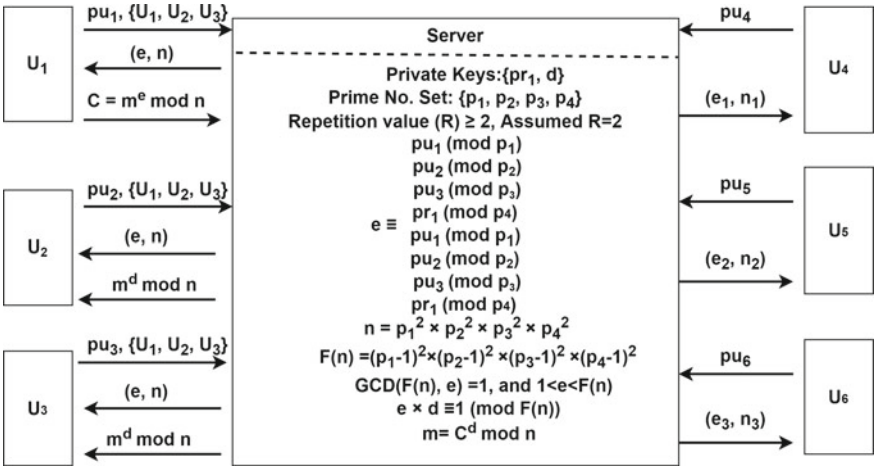


Fig. 5 Group communication using multi-party modified RSA

3.2 Group Communication

In this phase, after getting the common key, the group communication is performed as shown in Fig. 5. The group communication is illustrated in the following steps:

1. Server S separately generates the $F(n) = (p_1 - 1) \times (p_2 - 1) \times (p_3 - 1) \times (p_4 - 1) \times (p_1 - 1) \times (p_2 - 1) \times (p_3 - 1) \times (p_4 - 1)$.
2. If $\gcd(F(n), e) \equiv 1$ and $1 < e < F(n)$ then only the next step is followed, otherwise change the value of pr_1 again, until we get $\gcd(F(n), e) \equiv 1$ and $1 < e < F(n)$.
3. Server S calculates the private key d by satisfying the constraint as $d \times e \equiv 1 \bmod F(n)$.
4. If any participant of a group $\{U_1, U_2, U_3\}$ wants to share a message m to the group, firstly, that message is shared in encrypted form to the server S as $C = m^e \bmod n$.
5. Later, the server S decrypts the cipher text as $m = C^d \bmod n$, and shares that decrypted message m in encrypted form as $m^d \bmod n$ to the group's participants U_2 and U_3.

4 Performance Analysis

In performance analysis, we are comparing the proposed approach with the existing approaches in relation to group key generation with the parameters such as rounds, message, multiplication, exponentiation as shown in Table 1, where the proposed approach has a zero exponential operation, making it more computation efficient than all other existing approaches, as the computation cost of exponentiation operation is

Table 1 Complexity analysis

Approach	Rounds	Messages	Multiplication	Exponentiation	Time complexity
Michael et al. [9]	$k+1$	$2k-1$	–	$5k-6$	$O(k)$
Naresh and Murthy [10]	$k+1$	$k+1$	–	k	$O(k)$
Murthy and Naresh [11]	1	$2k$	k^2	k^2	$O(k^2)$
Mandal and Mohanty [8]	$k+1$	$k+1$	–	k	$O(k)$
Rawat and Deshmukh [7]	$\log_2(k)+1$	$2k-2$	$2\log k + c$	–	$O(k)$
Proposed approach	2	$2k$	$2Rk+1$	–	$O(R)$

dominant over multiplication. The proposed approach's multiplication cost, $2Rk + 1$, is a little bit more expensive than some of the existing approaches, where k is the number of participants in a group and R is the number of repetitions applied in the proposed approach. The number of messages interchanged is $2k$, which is almost equivalent to the existing approaches. The number of rounds required by this approach is 2, more efficient than the existing approaches. The time complexity of this new approach is also more efficient than all, taking $O(R)$ as unit cost.

5 Conclusion and Future Scope

The proposed approach is Chinese remainder theorem based multi-party key generation using modified RSA protocol. The Chinese remainder theorem provided a common key for the group without violating the security in less computation cost and modified RSA provided the secure communication for the group. The approach is computation and communication efficient for generating the common key for the group, because there is no use of modular exponentiation operation in key generation, only by applying the modular multiplication operation we make the approach computationally efficient. The key generation of a group is more suitable for a low configuration system, because there is no expensive operation. In complexity analysis, we found that our approach is giving better results in exponentiation, time complexity, rounds, a little bit expensive in multiplication cost, and equivalent performance in message transmission than the existing approach making it a communication-efficient protocol. The approach is giving much security than other existing approaches and is also very helpful for the low configuration system in terms of key generation of a group. The modified RSA reduced the threat of the factoring attack as the repetition R value is increased. The major limitation of the approach is that it is not suitable for a low configuration system when communication starts, because it uses expensive modular exponential operations for encryption and decryption, but it provided better security for the group. In the future we will also improve the computation cost in the communication phase with equal security.

References

1. Boyd C, Mathuria A (2013) Protocols for authentication and key establishment. Springer Science and Business Media
2. Forouzan BA, Mukhopadhyay D (2011) Cryptography and network security (Sie). McGraw-Hill Education
3. Stallings W (2006) Cryptography and network security, 4/E. Pearson Education India
4. Stinson DR (2005) Cryptography: theory and practice. Chapman and Hall/CRC
5. Diffie W, Hellman M (1976) New directions in cryptography. IEEE Trans Inform Theory 22(6):644–654
6. Khatarkar S, Kamble R (2015) A survey and performance analysis of various RSA based encryption techniques. Int J Comput Appl 114(7)
7. Rawat AS, Deshmukh M (2019) Efficient extended diffie-hellman key exchange protocol. In International conference on computing. power and communication technologies (GUCON). IEEE
8. Mandal S, Mohanty S (2014) Multi-party key-exchange with perfect forward secrecy. In 2014 international conference on information technology. IEEE
9. Michael S, Tsudik G, Michael W (1996) Diffie-Hellman key distribution extended to group communication. In Proceedings of the 3rd ACM conference on computer and communications security
10. Naresh VS, Murthy NVES (2010) Diffie-Hellman technique extended to efficient and simpler group key distribution protocol. Int J Comput Appl 4(11):1–5
11. Murthy NVES, Naresh VS (2010) Extended diffie-Hellman technique to generate multiple shared keys at a time with reduced keos and its polynomial time complexity. IJCSI Int J Comput Sci Issues 7
12. Rawat A, Deshmukh M (2020) Tree and elliptic curve based efficient and secure group key agreement protocol. J Inform Secur Appl 55:102599