American Journal of Applied Sciences 6 (9): 1714-1717, 2009
ISSN 1546-9239

Group Re-keying Protocol Based on Modular Polynomial Arithmetic Over Galois Field GF(2^n)

[1]Sundaram Sudha, [1]Azman Samsudin and [2]Mohammad Ahmad Alia
[1]School of Computer Sciences, University Sains Malaysia, Malaysia
[2]Department of Computer Information System, Al-Zaytoonah University of Jordan, Jordan

Abstract: Problem statement: In this study we propose a group re-keying protocol based on modular polynomial arithmetic over Galois Field GF(2^n). Common secure group communications requires encryption/decryption for group re-keying process, especially when a group member is leaving the group. **Approach:** This study proposes secret keys multiplication protocol based on modular polynomial arithmetic (SKMP), which eliminates the need for the encryption/decryption during the group re-keying. **Results:** The implementation based on modular polynomial arithmetic over Galois Field GF(2^n) offers fast re-keying process (about 50% faster than Secret Keys Multiplication Protocol (SKM) for 128 bit key) and compact key size representation against other secret keys multiplication protocols. With SKMP group re-keying is handled more efficiently through modular polynomial arithmetic manipulation rather than the expensive encryption/encryption which need to be done on every membership change.

Key words: Multicast, group re-keying, public-key, Polynomial arithmetic, Galois Field GF(2^n)

INTRODUCTION

In the modern world, most user-based network applications such as multimedia streaming, multi-party video conferencing, pay per view of digital media content and others; need efficient, scalable and secure group communication. One of the network protocols that can meet such requirements is the multicast communication.

Multicast communication is a network protocol that is being used for communication among users to deliver data from a sender to multi-receivers efficiently. Multicast over unicast has an advantage that it utilizes less network resources. In multicast communication, data is delivered only to a group of anathematized users which is denoted as multicast group. One of the prominent needs in multicast communication is security. To achieve secure multicast communication, encryption/decryption is normally being employed. However changing membership within a multicast group can post a serious performance degradation problem, especially when membership changes are frequent.

Secret Keys Multiplication Protocol (SKM)[1] introduces a simple re-keying method that eliminates the need for encryption/decryption. In this study secret key multiplication protocol based on modular polynomial arithmetic over Galois Field GF(2^n) (SKMP) is proposed. The key in the proposed protocol is transmitted to the users in the form of modular polynomial over Galois Field GF(2^n). Through the use of modular polynomial arithmetic, faster re-keying process is achieved, and with a much more compact key size.

Multicast network system: Network system is defined as a communication between computers. The multicast network is the group of interested users. Figure 1 shows a secure multicast whereby the sender transmits data to receivers via a multicast network. The control server is responsible for generating and distributing keys to both the sender and receivers.

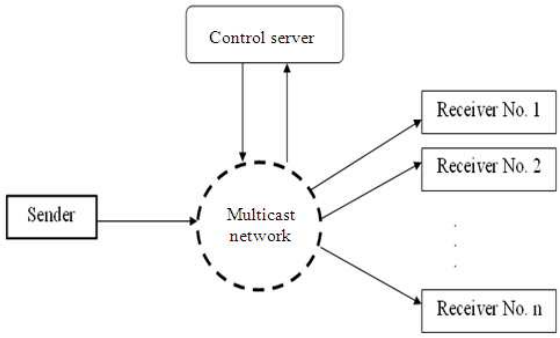

Fig. 1: An example of multicast network system

Corresponding Author: Mohammed Alia, Faculty of Sciences and IT, Al-Zaytoonah University of Jordan, 11733 P.O. Box 130, Amman, Jordan Tel: +962-6-4291511 Fax: +962-6-4291432

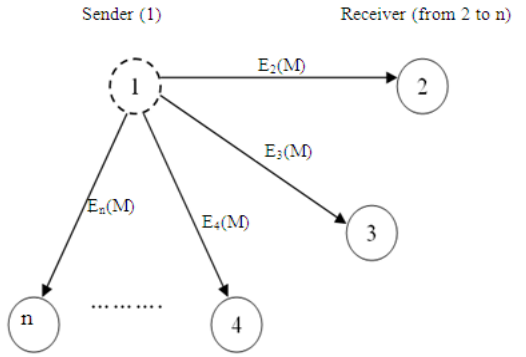

Fig. 2: Transmission of the message M through four point-to-point connections

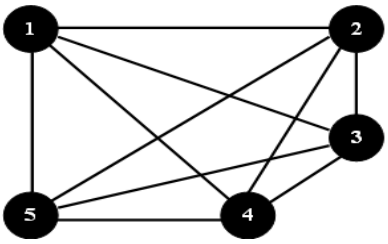

Fig. 3: Communication graph with n(n-1)/2 individual point-to-point connections

Figure 2 shows an example of a multicast network where message M is being transmitted over four points. Point 1 is the sender while the rest of the nodes are the receivers in the network. Point 1 sends the four different encrypted messages ($E_1(M)$, $E_2(M)$, $E_3(M)$ and $E_4(M)$) to the receivers respectively. Figure 3 shows one possibility how the members in the group can be connected. Generally, there will be n(n-1)/2 individual point-to-point connections for the multicast scheme.

Existing secrets keys multiplication protocol: There are several proposed scalability multicast group re-keying protocols. Among them are group key approach[1-3], contributory key agreement supported by Diffie Hellman algorithm[4], and logical key tree based approach[5,6]. Among the group re-keying methods mentioned above, SKM[1] is one method that does not depend on encryption/decryption for its group re-keying process.

Secret keys multiplication for scalable group re-keying (SKM)[1]: In secure group communication users of a group share a common group key. Normally in a secure group communication protocol, the group controller sends to the group members a new key to authorize new users as well as performs the group re-keying for group users whenever the key changes.

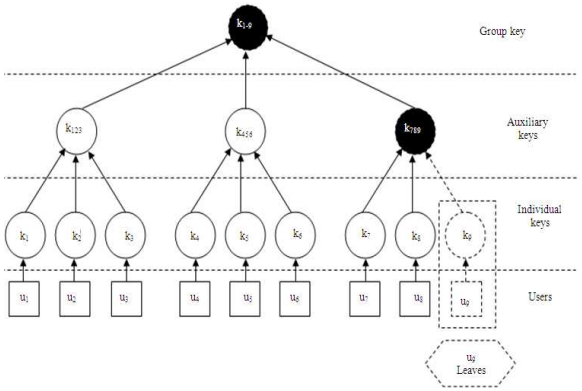

Fig. 4: An example of key tree with nine users

The SKM protocol uses the logical tree hierarchy of key exchange among the group members by multiplying the group secret keys. For data protection, SKM protocol uses a modular arithmetic which is applied to the individual key.

SKM protocol uses secret key multiplication in conjunction with the key tree approach. This approach is managed by the trusted server called Group Controller (GC). The trusted GC, who owns the private key k_c, uses the key tree for group key management. To provide multicast security, each secured multicast group is associated with one trusted server for managing the group communication. An example of a key tree is shown in Fig. 4.

As shown by Fig. 4, the u nodes are the users of a group, the k nodes are the keys and the group key (group secret key) is the session key The user u_1 to u_9 holds individual keys as k_1 to k_9. k_{123} is the auxiliary key share by user u_1, u_2 and u_3. Similarly, k_{456} and k_{789} are shared by their users, u_{4p}-u_6 and u_7-u_9, respectively. k_{1-9} is the session key and is known to all the group members. As shown by Fig. 4, the individual keys are located at level-2 of the tree, the auxiliary keys are located at level-1 of the tree and the session key k_{1-9} is located at level-0 of the tree. These levels are managed with a condition as that level-2 keys must be greater in value than level-1 and level-0 keys. Similarly, the level-1 keys must be greater in value than level-0 keys.

To explain the SKM group re-keying protocol, assume user u_9 wants to leave the group. Then the GC has to change the secret key which is known to u_9, as well as other users. To manage the keys, a re-keying process has to be done. k_{1-9} will be changed to k_{1-8}, k_{789} is changed to k_{78} and k_9 will be deleted from the tree. Before generates a new secret key, the GC changes its private key from k_c to k_c'. After performing the calculation as shown by Eq. 1, the GC will multicast the

values X and Y to the rest of the group members (u_1-u_8). Users u_7 and u_8 recover the new auxiliary key, $k_{78,}$ by using their individual private keys k_7 and k_8 respectively (Eq. 2 and 3). With the auxiliary key k_{78} users u_7 and u_8 can recover the new session key, $k_{1\text{-}8}$, by executing Eq. 3. Similarly, users u_1-u_3 and u_4-u_6 can recover the new session key by using either the respective auxiliary keys, k_{123} or k_{456}:

$$X = k_7 \times k_8 \times k_c' + k_{78}$$
$$Y = k_{123} \times k_{456} \times k_{78} \times k_c' + k_{1\text{-}8} \quad (1)$$

$$k_{78} = X \bmod k_7$$
$$k_{78} = X \bmod k_8 \quad (2)$$

$$k_{1\text{-}8} = Y \bmod k_{78} \quad (3)$$

MATERIALS AND METHODS

The proposed scalable group re-keying method based on modular polynomial arithmetic: This study proposes an enhancement to SKM by implementing SKM with modular polynomial arithmetic over Galois Field GF(2^n). We are comparing the proposed protocol with the existing SKM protocol to show the enhancement in the computation speed. Figure 5 shows an example of a logical arrangement of the users and the nodes in the proposed method. The key structure is stored in a hierocracy tree form similar to the SKM key tree structure.

In Fig. 5, U_{n1} to U_{n9} denote the user keys. These 9 user keys are connected with three subgroup keys S_{n1}, S_{n2} and S_{n3}, and the subgroup keys are further connected to the session key, R_n. Similar to SKM, k_c and k_c' are secret keys own by the Group Controller. The secret key k_c (and its derivations) is a random number that should be changed for every re-keying process. Each key (U_{ni}, S_{ni}, R_n) is represented by a binary string, $b_{n-1}...b_1b_0$. This binary string is then further represented in its polynomial form, $P(b_{n-1}\ldots b_1b_0) = b_{n-1}x^{n-1}\times\cdots\times b_1x^1\times b_0x^0 = \sum_{i=0}^{n-1} b_i x^i$, which is used in the calculation as stated in Eq. 4-6.

If a new user joins or an existing user leaves in any one of the subgroup, the Group Controller will change the corresponding subgroup key and transfer the new subgroup key to the respective users in a secured way similar to the SKM methods. However the calculation of the new keys will be done in modular polynomial arithmetic over the Galois Field GF(2^n).

In the proposed re-keying method, we design new modular polynomial equations for transmitting the key in a secured manner (Eq. 4-6). Note that we use $\oplus$ and $\otimes$ to denote modular polynomial addition and multiplication over Galois Field GF(2^n), respectively:

$$P(X) = P(U_{n7}) \otimes P(U_{n8}) \otimes P(k_c') \oplus P(S_{n3}')$$
$$P(Y) = P(S_{n1}) \otimes P(S_{n2}) \otimes P(S_{n3}') \otimes P(kc') \oplus P(R_n') \quad (4)$$

$$P(S_{n3}') = P(X) \bmod P(U_{n7})$$
$$P(S_{n3}') = P(X) \bmod P(U_{n8}) \quad (5)$$

$$P(R_n') = P(Y) \bmod P(S_{n3}') \quad (6)$$

As shown by Fig. 5, the keys for each user and the subgroup are assigned by the Group Controller. The key value is given as a binary input which is then transform to its equivalent polynomial form for the re-keying process. Therefore, the bit length of the proposed method is totally reduced compared to the SKM method where the key is in a form of integer in Finite Field Z_n, where n has to be large for security reason.

As shown by Fig. 5, if the user U_9 is leaving then the group, the GC (group controller) has to change the session key which is known by the U_9, as well as other users. In the re-keying process, the subgroup key, S_{n3}, is changed to S_{n3}' by executing Eq. 4 and the session key, R_n, is changed to R_n', while U_{n9} is being deleted from the tree. For security reason, the Group Controller also changes its private key k_c to k_c'.

The re-keying calculation structure in SKMP is similar to the re-keying calculation structure found in SKM. As shown by Eq. 4, after creating the new values S_{n3}' and R_n', the Group Controller multicasts X and Y to the group members (U_1-U_8). The new value of S_{n3}' is embedded in Y while the new session key R_n' is embedded in Y. To recover the new values S_{n3}'and R_n', users U_7 and U_8 can execute Eq. 5 and 6, respectively. Similar, users U_1 to U_6 can recover the R_n' by using S_{n1} and S_{n2} in Eq. 6.

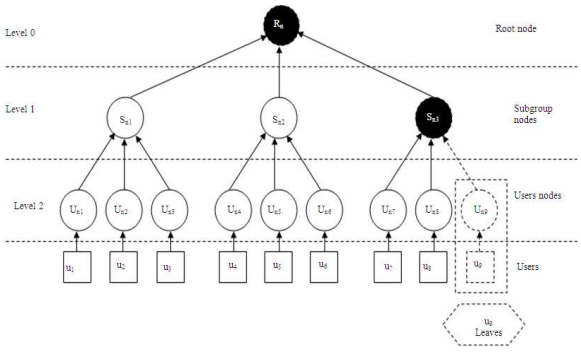

Fig. 5: Tree based structure of the proposed system

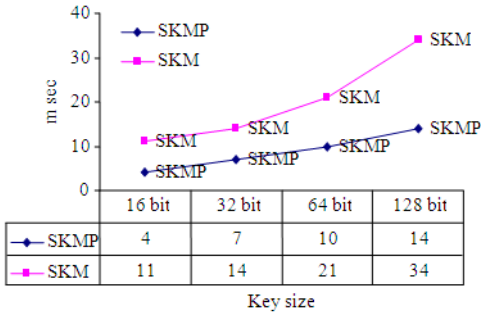

Fig. 6: Overall time comparison between the execution of SKMP and SKM protocols

Table 1: Performance evaluation between SKMP and SKM protocols

Key size (bit)	SKMP time (m sec)	SKM time (m sec)
16	4	11
32	7	14
64	10	21
128	14	34

RESULTS

We compared the performance of the modular polynomial arithmetic based secret keys multiplication (SKMP) against exiting secret keys multiplication protocol (SKM) (Fig. 6). Table 1 shows the performance for both approaches. Both protocols were coded in Turbo C with NTL library[7]. The NTL library is used to handle the polynomial arithmetic operation. Both protocols were run on a computer with 1.6 GHz Intel® M Pentium processor and 256MB RAM.

DISCUSSION

The comparison between SKMP and SKM protocol shows that SKMP protocol performs better than SKM in general. As Fig. 6 indicates, the secret keys multiplication based modular polynomial arithmetic provides higher level of security at a much lower cost, both in term of key size and execution time.

CONCLUSION

This study has shown the possibility of establishing a method of multicast group re-keying based on polynomial arithmetic operation for data transmission in order to reduce the computational cost. As the result, the proposed modular polynomial secret keys multiplication protocol requires a much lower cost of execution time and performs at a high level of security compared to the existing Secret Keys Multiplication protocol (SKM).

ACKNOWLEDGMENT

The researchers would like to thank the Al-Zaytoonah University of Jordan and the Universiti Sains Malaysia for supporting this study.

REFERENCES

1. Mohamed, Y. and A. Kara, 2003. Secret keys multiplication for scalable group re-keying. Proceeding of the 3rd International Conference on Information Technology, (CIT'03), ACM/IGMOD, pp: 247-251. http://www.pubzone.org/dblp/conf/cita/MohamedK03
2. Yang, Y.R., X.S. Li, X.B. Zhang and S.S. Lam, 2001. Reliable group rekeying: A performance analysis. Proceedings of the 2001 Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, (ATAPCC'01), ACM Press, San Diego, California, United States, pp: 27-38. http://portal.acm.org/citation.cfm?id=383062
3. Rafaeli, S. and D. Hutchison, 2000. A decentralized architecture for group key management. http://eprints.kfupm.edu.sa/17283/
4. Diffie, W. and M. Hellman, 1976. New directions in cryptography. IEEE Trans. Inform. Theor., 6: 644-654. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1055638
5. Chang, I., R. Engel, D. Kandlur, D. Pendarakis and D. Saha, 1999. Key management for secure internet multicast using boolean function minimization techniques. Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies, Mar. 21-25, IEEE Xplore Press, New York, USA., pp: 689-698. DOI: 10.1109/INFCOM.1999.751455
6. Yang, C. and C. Li, 2004. Access control in a hierarchy using one-way hash functions. Comput. Secur., 8: 659-664. DOI: 10.1016/j.cose.2004.08.004
7. Shoup, V., 2009. NTL: A library for doing number theory. http://www.symbolicnet.org/systems/NTL.html