A Group Secure Key Generation and Transfer Protocol Based on ICMetrics

Hasan Tahir, Klaus McDonald-Maier
School of Computer Science and Electronic Engineering
University of Essex
Colchester, United Kingdom
htahir@essex.ac.uk, kdm@essex.ac.uk

Abstract – **Secure group communications are more prone to attacks as compared to the conventional one to one communication. Every client in a group can be seen as a single source of attack, therefore it is important to design a robust security scheme that will protect all the individual clients and hence the entire group. In this paper a novel security architecture has been presented, that provides a secure group key generation and transfer protocol that is based on ICMetrics. The salient features of the protocol include a single collaborative key generation scheme that is initiated through client authentication. Also provided is a rekeying procedure that is important in maintaining the freshness of the key and offers perfect forward secrecy. The above features are based on the use of ICMetrics to provide a security protocol that is scalable and secure. The presented protocol has been simulated for varying group population sizes using C++ and Maple. The resulting running times for various stages of the protocol have been studied.**

Keywords – ICMetrics; group secure communications; key generation; group key; group security.

I. INTRODUCTION

Group based security is an aggravation of one to one security. Security provision is a delicate matter, as overprovisioning can cause an increase in the amount of resources required and ultimately in the inefficiency of the entire operation. When studying group based security algorithmic efficiency is important, as we are concerned with a large client population, therefore a slight decrease in the efficiency of the system will result in poor system security/performance. The importance of having secure group communications is more important because there are many clients involved and every client can be considered a single point of failure. Hence having a larger number of clients implies having more weak points in the group.

The complexity in group security exists, because group security requires specialised procedures for key generation, transportation and key administration among other prominent activities. Client to client communication is generally based on only two clients, i.e. a sender and a receiver [1]. However, group communications can be composed of a mesh based architecture, master client architecture or a self-organizing mesh network. In any case the complexity involved is evident due to the high number of entities involved.

A novel technique for enforcing security is underway that promises an effective and ingenious solution to our current security related problems. The technique is called Integrated Circuit Metrics (ICMetrics) and provides security by using individual features or characteristics of a device [2]. Conventionally, a user is required to generate a key pair, which can involve the use of prime numbers and random numbers along with mathematical functions. The same key generation is achieved by using ICMetrics and it allows the generation of a unique number that is based on the characteristics of the device. The characteristics of a device can include things like identification numbers, addresses, PC counter data and other device related data. Research is underway that allows a device to generate a unique, yet stable number that can be used for further generation of a key. This number is known as the ICMetric and is further used in combination with complex algorithms and techniques to provide a comprehensive security solution. This technique is not an alternative way of producing a random number; because the number being produced is generated from hardware and software based feature extraction at real time, hence no ICMetrics related data needs to be stored on the system. The only way of extracting the ICMetrics data is to tamper with the constituent hardware unit, which will result in the failure of the device.

Much research [2][3][4] is underway in designing and extending ICMetrics, so that it can be fully implemented. Researchers are already beginning to acknowledge the added benefits ICMetrics has to offer as compared to conventional security. Perhaps the greatest advantage of using ICMetrics, is that an attempt to extract the ICMetric data will fail, as the hardware that generates ICMetrics can be designed to be tamper resistant [5].

To this end, we propose a multicast security protocol that provides a key generation, key transfer and rekeying procedure. We then explore the usefulness of the protocol by simulating its operations for varying sized client populations.

This paper begins by first discussing contributions that have already been made to the field of group communications. Then, two prominent keying perspectives have been presented, that explain the various aspects of keying in a group setting. Section IV provides a detailed description of the protocol and how it operates. Also provided in section IV are details relating to the rekeying protocol. In section V we provide a standalone and a comparative analysis of the protocol with other rivalling protocols.

II. RELATED WORK

Perhaps the most prominent and widely acknowledged work in the domain of key generation and exchange has been done by Diffie and Hellman [6]. They effectively proposed a key exchange protocol that allows two parties to exchange data and in result exchange keys without transmitting the keys over the medium. The problem with the Diffie Hellman key exchange is not only it being prone to attacks (bucket brigade attack), it is also designed for key exchange between no more than two parties.

To resolve these issues many researchers have strived to fine tune the Diffie Hellman Key exchange and also extend the protocol so that it can be operated in a group setting [7][8]. Many protocols require the services of a Key Distribution Centre (KDC) and a Key Generation Centre (KGC) for handling rogue entities and also assist in managing entities that join or leave the group at any moment. Most modern protocols acknowledge that group based security should use the services of an administrative body like the KDC and KGC. When we view the individual activities of a group performing communications, we see that member admittance, member removal, key renewal and key distribution are some of the major operations that will take place in the group.

Once the above stated procedures are functional, the individual keys need to be managed in relation with their owners. This in itself is a complex procedure, because no matter what data structure is used, the size of a group and the dynamics of the environment cannot be predicted in advance. Therefore, researchers have worked on a tree based structure [9][10] to store keys in an efficient manner, so that when members join or leave the group the key management activity can be performed both efficiently and accurately. Researchers have successfully attempted to refine and make efficient these tree based data structures.

Papoutsis [17] in his work has performed an investigation into the potential of generating encryption keys using ICMetrics. The research provides methods and experiments that explain the process of generation of the ICMetric number.

Tahir et al have proposed a secure key generation protocol that is based on the use of ICMetrics. They have designed a strong key pair key generation scheme that uses SHA-2 and session token to generate a public-private key pair that is used in one to one secure communications.

Individual and dispersed work has been done in the individual areas relating to multicast security. Based on the expected advantages of ICMetrics, it is important to consider the use of ICMetrics in multicast security.

III. KEYING PERSPECTIVES

Key generation in a group protocol is a unique affair as group communication can require the use of group keys and individual keys. The ease of using group keys is, that a single key can be generated that will govern communications between the entire group. Of course if the key is leaked, then the entire operation becomes insecure. Researchers [11][12] emphasize on the use of contributory keys, as they believe that it provides more randomness, key freshness and also ensures that once a person leaves or joins, a new key is produced, owing to which a guarantee of perfect forward secrecy can be given. Key agreement is generally of two types, namely initial key agreement and auxiliary key agreement. For a group security protocol to be fully operational it must possess both the initial and auxiliary protocols.

A. Initial Agreement Protocol

The initial key agreement protocol is the most important key agreement protocol because this is used when a new client attempts to join the group. This protocol needs to be efficient, as many clients may be attempting to join a group communication at a single time. At the time of group conception, a large number of clients may be joining the group in parallel, hence streamlined performance of this protocol is essential. During the initial key agreement it is important to also authenticate the incoming client and also obtain relevant data for generation of the contributory key. These individual events, although seemingly simple are complex and hence require guaranteed efficiency.

B. Auxiliary Protocol

The auxiliary key agreement is a secondary protocol that will come in effect when a rekeying procedure may be needed. This protocol is utilized when client(s) leave the group communication or a simple rekey is required. Since group communications are dynamic in nature, the services of the auxiliary protocol are required when there is a membership change in the group. In the absence of the auxiliary protocol, there is no defined mechanism for the provision of perfect forward secrecy. Hence, even if a client leaves the group he will still have access to the current key as his departure did not trigger a rekeying process. This implies that the auxiliary protocol is required to overwrite the previous keys that are prescribed by the group controller.

IV. PROPOSED WORK

Before formally introducing the proposed protocol, the target security goals have to be understood. These goals, if fully addressed, can provide a comprehensive protocol that can be easily adopted.

- The protocol should be based on the use of ICMetrics for the generation of ICMetric Keys.
- The keys should be updated frequently to ensure group secrecy.
- Perfect Forward Secrecy – Whenever a client leaves a group, it should not have access to future keys and communications taking place in the group [13].
- Backward Secrecy - Whenever a new client is admitted to a group, it should not have access to previous keys and communications taking place in the group [14].
- If any data required for key generation is sent unencrypted, then it should not become a threat, if the data is captured by an adversary, i.e. the attacker should not be able to bring down the entire system just because he could access a part of the data.

- The group key protocol should allow clients to join or leave a group at any time.
- The group key protocol should be scalable, i.e. it should be computationally feasible if the size of the group increases.
- The protocol should protect from current clients that have become rogue.
- The protocol should easily interoperate with existing technologies and protocols.

A. User Registration

The protocol is initiated, when a user wishes to get registered for group communication. Each client, when it is about to get registered, is in fact a single entity upon which the conventional rules of client to client communication apply. Once the client is admitted into the group only, then it can enjoy the benefits of the group communications. A vital question which is often not asked is which clients are going to be participating in the group communication. The first client to initiate a group will host the group communication and will "name" those members that will be involved in the group communication. This first member of the group will place the public keys with the group controller of those clients that will be part of the group. This means, when a client intends to join the group the group controller can generate a Turing test encrypted with the clients public key and will expect a correct response, if the client is genuine and has the right private key for decryption. So if an incorrect response is obtained for the extracted Turing test then it means that the entity is either not human or an attacker. Hence bot based attacks can be reduced at source.

For the purpose of enhanced security, the group controller will periodically perform the above test on all clients that are part of the group communication. This will ensure that any client that has abruptly disconnected, is kicked out of the communication and any client that can pose to be some other entity is automatically filtered out.

B. Key Generation

The key generation protocol is a 9 step protocol, which consists of producing a group communication key by exchanging data between the individual group members and the group controller. The important fact here is, that through the use of this protocol the generated key is a contributive key, which is generated through individual contributions by the clients. The clients cannot contact each other for generating the key. The individual clients generate their individual data and transmit to the group controller. Given below are the steps that will be performed by the individual clients:

Consider n registered members in a group communication system $(N_1, N_2, \dots, N_n)$

Step 1: Each member N_i, for $i = 1,2, \dots, n$ generates its own ICMetric number IC_i.

Step 2: Every N_i computes an X_i, by taking the XOR of IC_i with a generated random number R_i, i.e.

$$X_i = IC_i \oplus R_i, i = 1,2, \dots, n \quad (1)$$

Step 3: With the Group Controller's public key GC_{Pub} each N_i encrypts X_i, which also serves the purpose of authentication of group controller,

$$Y_i = E_{GC_{Pub}}(X_i), i = 1,2, \dots, n \quad (2)$$

Once this step is completed Y_i is sent to the group controller for further computation and generation of the group key. Hence the following steps (4-9) will be performed by the group controller.

Step 4: The Group Controller (GC) will decrypt each Y_i with its own private key GC_{Priv}

$$Z_i = D_{GC_{Priv}}(Y_i), i = 1,2, \dots, n \quad (3)$$

Step 5: GC selects an arbitrary number K where $K < Z_i, \ \forall i = 1,2, \dots, n$, i.e.

$$K < Z_1, K < Z_2, \dots, K < Z_n \quad (4)$$

Step 6: Now GC generates the Message M

$$M = (Z_1 * Z_2 * \dots * Z_n) + K \quad (5)$$

Step 7: GC encrypts M with every member's public key $E_{N_{i_{pub}}}$ and sends the encrypted message S_i individually to all members N_i where $i = 1,2, \dots, n$

$$S_i = E_{N_{i_{pub}}}(M), \quad i = 1,2, \dots, n \quad (6)$$

Step 8: Each N_i decrypts the received S_i with its own private key $D_{N_{i_{Priv}}}$

$$D_{N_{i_{Priv}}}(S_i), \quad i = 1,2, \dots, n \quad (7)$$

Step 9: Now every group member N_i derives the key K at its own end, i.e.

$$K = S_i mod \, X_i, \quad i = 1,2, \dots, n \quad (8)$$

Upon derivation every group member will be equipped with a symmetric group communication key that will be used to communicate within the group. Any entity that is not registered in the group will not possess the group key and hence will not be able to communicate within the group.

Fig. 1 provides a sequential flow that shows the phases and their relationship with the client and the controller.

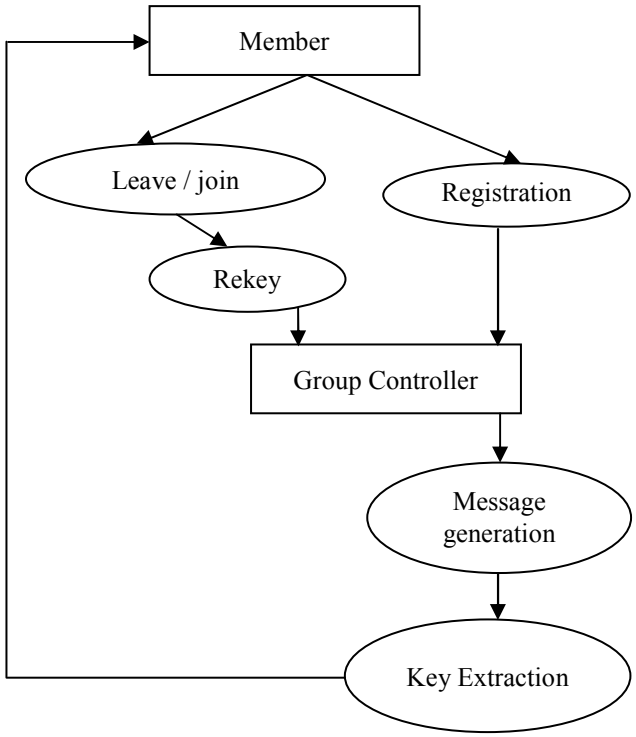

Fig. 1. System design and activities

C. *Rekeying Procedure*

The rekeying procedure is a necessary part of group communications. Conventionally it is believed that the rekeying procedure is only required for post welcome and farewell of a client. However, researchers [10] have presented other scenarios where such a protocol may be required. The full list of scenarios is member addition, member deletion, mass join, mass leave, group division and group fusion.

It is evident that in the absence of such a procedure, the entire group is at the mercy of the departing group member for not disclosing the key to an entity outside the group. The rekeying procedure is also important for giving admission to an entity that may wish to join the group in communications. When designing the rekeying protocol we propose adherence to the same scheme with slight sequential modification. By following the same scheme the source code can be designed by using modular functions which can be called according to the need of the algorithm. The rekeying procedure is based on the same protocol but with slight sequence modifications. Fig. 2 shows the sequence of events required for the rekeying procedure in a group setting.

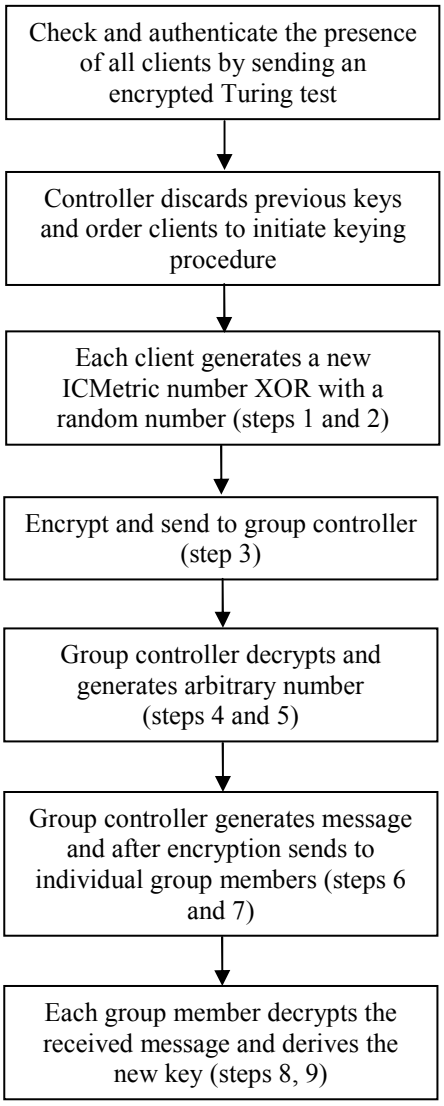

Fig. 2. The rekeying procedural steps

V. SIMULATION AND RESULTS

It is standard practice to simulate and evaluate a newly designed protocol to prove its efficiency. To do so, the key exchange protocol was simulated in C++ and MAPLE on a 3rd Generation Core i5 system with 6GB RAM. It was discovered that the conventional data types did not offer sufficient length/size to deal with numbers on the scale of 20 digits. Therefore the unsigned long long data type has been used for this purpose.

Two separate user defined functions were designed for the encryption and decryption process. This enables the reuse of code, since the encryption and decryption is being done twice in the entire process. The outputs at each step have been fully traced and verified for correctness. Since there are steps in the process that may encounter communication delay or a simple delayed response. These times have not been simulated, since they are dependent upon a variety of other factors (network delay, propagation delay, system delay), which cannot be produced in a simulation environment.

Person to person protocols are just simulated for two persons, however in group based communication the protocol needs to be run for varying sized populations. The protocol has been simulated starting with a small group size of 25 and grows to a large size group of 200 members.

In the first simulation, the entire protocol was simulated to study the time the protocol should take, if it was all inclusive. The results are shown in the graph in Fig. 3 and indicate a moderate time of 200ms for a maximum group size of 200 clients

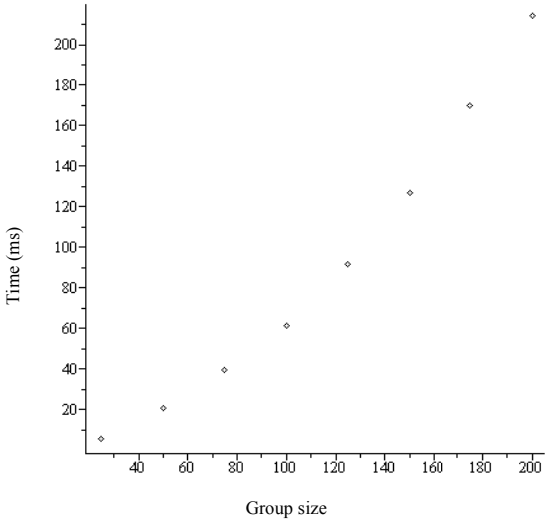

Fig. 3. Time (ms) taken for key generation and distribution in client population of 25-200

The client activities from step 1 to step 3 are important in the sense that they form the basis of the entire operation. When simulated, these steps were fairly intensive and consume much of the time in our defined group population of 25 to 200 clients. As shown in Fig. 4, these initial steps although seemingly simple consume a major portion of the time required in the entire key generation.

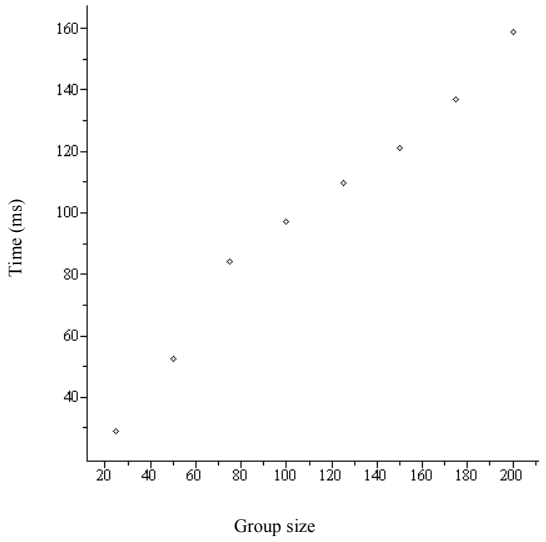

Fig. 4. Time (ms) taken for the clients to perform steps 1 to 3 for client population of 25-200

The message generation process is perhaps the most crucial and intensive operation in the protocol, since it is composed of computations performed by collecting data from all participants. Then, the message has to be encrypted for submission to the individual clients. The message generation and encryption was simulated and the projected time in milliseconds is simulated in Fig. 6.

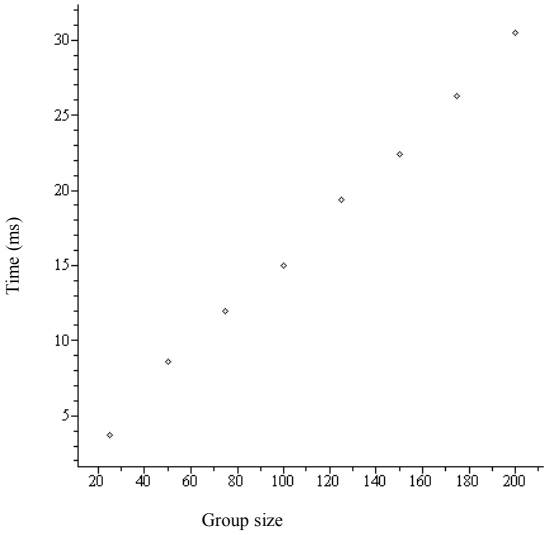

Fig. 5. Time (ms) taken for message generation and encryption by group controller for client population of 25-200

A closer comparison of this protocol with other rivalling protocols has shown that this protocol outperforms other protocols even though it has additional ICMetrics features. Most security protocols have been simulated to a relatively small group sizes ranging from 25 to 40. Such a small population size should not be used for simulation purposes because the time difference between the upper and lower limit of the population is seen as insignificant. Rao et al have presented a secure key transfer protocol for group communications. The operating time for 25 group members is around 45 milliseconds, which in our case is 40 milliseconds. The authors have not projected the running time for group size greater than 25 clients.

Table 1 provides a summary of increase in time in comparison with group size. It is interesting to note that as the group size increases there is a sharp increase in the time taken for key generation and distribution.

Table 1: Projected times for 100 and 200 participants

Simulation	Time (ms) for 100 participants	Time(ms) for 200 participants
Key generation and distribution	70	210
Message generation and encryption	15	32

Venkatesulu et al have presented a similar protocol that provides secure group communications in grid environments. This protocol takes a key extraction time of 410 milliseconds for a group size of 200 clients. Our protocol takes 210

milliseconds for running the entire protocol. These comparisons prove that our proposed protocol is both secure and efficient.

VI. CONCLUSION

Often group based security is seen as a large scale communication between many small groups of two clients. This notion is misguided, as it does not consider the complexities involved in the creation of a group. Whenever a group is formed, there has to be a group controller that can administer the individual group members. This paper presents a novel security protocol that provides first an authentication mechanism for group members and then it provides a group key generation and transfer technique based on ICMetrics. This proposed protocol is unique, because it produces a single group communication key that is generated through individual contributions from each client in the group. To make the protocol even more resilient, we have based the protocol on the use of ICMetrics. ICMetrics is a breakthrough in the area of system identification and security provision. ICMetrics promotes the use of a hardware/ software feature extraction that can be installed on every device and hence generate a single number that is different for each device. This number will typically vary significantly from device to device because every device has a unique environment/software/hardware profile.

Our proposed protocol uses the generated ICMetric and performs a series of computations and communications between the clients and group controller to generate a single group key. Also presented in the paper is a group rekey protocol that is effective when clients are admitted or removed from the group. To prove the effectiveness of the protocol, it has been simulated for varying sized client population. To start with, the protocol has been studied for a small group of 25 participants. Then the population of the group is increased with increments of 25 up to a maximum of 200 participants. Based on the simulated results, it can be safely said that the algorithm is fully scalable and is expected to perform well even for groups that exceed 200 clients.

ACKNOWLEDGEMENT

This research is partially financially supported by the EU Interreg IVA 2 Seas Programme (SYSIASS project: Autonomous & Intelligent Healthcare System, http://www.sysiass.eu/), and Franco-British programme (COALAS Project) that has been selected in the context of the INTERREG IVA France (Channel) England European cross-border co-operation programme which is co-financed by the ERDF.

REFERENCES

[1] A. Shabtai, Y. Elovici, L. Rokach, A Survey of Data Leakage Detection and Prevention Solutions, Edition of book, Springer US, 2012.

[2] A. Kokosy, T. Floquet, G. Howels, H. Hu, M. Pepper, M. Sakel, C. Donzé, "SYSIASS – an intelligent powered wheelchair," 1st International Conference on Systems and Computer Science, Villeneuve d 'Ascq, Lille France, September 2012.

[3] Y. Kovalchuk, K. D. McDonald-Maier, G. Howells "Overview of ICmetrics Technology – Security Infrastructure for Autonomous and Intelligent Healthcare System". International Journal of u- and e- Service, Science and Technology vol. 4, no. 3, pp. 49,60 September 2011.

[4] R. Tahir, K. D. McDonald-Maier, "Improving Resilience against Node Capture Attacks in Wireless Sensor Networks using ICMetrics", IEEE Conference on Emerging Security Technologies, Portugal, September 5-7, 2012.

[5] X. Zhai, K. Appiah, S. Ehsan, H. Hu, D. Gu, K. D. McDonald Maier, "Application of ICmetrics for Embedded System Security,"Fourth International Conference on Emerging Security Technologies (EST) 2013, pp. 89, 92, 9-11 Sept 2013.

[6] W. Diffie, M. Hellman, "New Directions In Cryptography, "IEEE Transactions of Information Theory, Vol. 22, no. 6, pp. 644,654, November 1976.

[7] E. Bresson, O. Chevassut, D. Pointcheval, "Provably Authenticated Group Diffie-Hellman Key Exchange — The Dynamic Case, "Springer Lecture Notes in Computer Science, vol. 2248 , pp. 290,309, 2001.

[8] H. Krawczyk. "HMQV: A High-Performance Secure Diffie-Hellman Protocol". In Proc. of CRYPTO'05, LNCS.

[9] Y. Kim, A. Perrig, and G. Tsudik. "Tree-Based Group Key Agreement". ACM Transactions on Information and System Security, vol. 7, no. 1, pp. 60,96, February 2004.

[10] M. Howarth, S. Iyengar, Z. Sun, H. Cruickshank, "Dynamics of Key Management in Secure Satellite Multicast," IEEE Journal On Selected Areas In Communications, vol. 22, no. 2, pp. 308,319, February 2004.

[11] M. Steiner, G. Tsudik, M. Waidner, "Key agreement in dynamic peer groups," Parallel and Distributed Systems, IEEE Transactions on, vol.11, no.8, pp.769,780, August 2000.

[12] M. Steiner, G. Tsudik, M. Waidner, "CLIQUES: a new approach to group key agreement," Distributed Computing Systems, 1998. Proceedings. 18th International Conference on, pp.380,387, 26-29 May 1998.

[13] Ai-fen Sui, Hui, L. C K; Yiu, S.M.; Chow, K.P.; Tsang, W. W.; Chong, C. F.; Pun, K. H.; Chan, H. W., "An Improved Authenticated Key Agreement Protocol with Perfect Forward Secrecy for Wireless Mobile Communication," Wireless Communications and Networking Conference, 2005 IEEE, vol.4, , pp. 2088, 2093 vol. 4, 13-17 March 2005.

[14] C. C. Chang, J.S. Lee, C. C. Chen, "On the forward and backward secrecy of HLL group key exchange mechanism," Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference on, pp.702, 705, 21-23 September 2005.

[15] R. Rao, K. Selvamani, R. Elakkiya, "A Secure Key Transfer Protocol For Group Communication," Advanced Computing: An International Journal (ACIJ), vol. 3, no. 6, pp. 83, 90, Nov 2012.

[16] M. Venkatesulu, K. Kartheeban, "An Efficient Certificate-Free Key Distribution Protocol for Secure Group Communication in Grid Environment," Journal of Computer Science, vol. 7, no. 6, pp. 917,923, 2011.

[17] E. Papoutsis, "Investigation Of The Potential Of Generating Encryption Keys For ICMmetrics," Ph.D. Dissertation, Dept. Electronics, Kent Univ, 2009.

[18] R. Tahir, H. Hu, D. Gu, K. McDonald-Maier, G. Howells, "A Scheme for the Generation of Strong ICMetrics Based Session Key Pairs for Secure Embedded System Applications," Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on, pp. 689, 696, 25-28 March 2013.