On the Incorporation of Secure Filter in ICMetrics Group Communications

Hasan Tahir*, Gareth Howells†, Huosheng Hu*, Dongbing Gu*, Klaus McDonald-Maier*

* School of Computer Science and Electronic Engineering, University of Essex, United Kingdom
htahir@essex.ac.uk, hhu@essex.ac.uk, dgu@essex.ac.uk, kdm@essex.ac.uk

† School of Engineering and Digital Arts, University of Kent, United Kingdom
W.G.J.Howells@kent.ac.uk

Abstract - **Secure group communications present a unique environment where there can be multiple clients and hosts are trying to communicate securely within the group. As the number of clients and hosts increases the complexity of the communication security also increases. Group communications are based on a dynamic environment where the clients may join or leave the group at any moment. Hence it is important to ensure that only permitted entities have access to the group and those that have left the group or are not part of the group have no access to the group communications. This paper explores the delineation of a secure communication filter function that is applicable to group communications and is based on the latest Integrated Circuits Metrics (ICMetrics). The proposed scheme is based on the use of hash functions. To test the scalability of the scheme it has been implemented using SHA1 and SHA2.**

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

I. INTRODUCTION

Much research has been done in designing protocols and techniques for secure one to one communication. Group communications differ from the classic one to one communication at an architectural level and hence at the complexity level. This is why group communications can be considered as an aggravation of the one to one communication. A typical group communications architecture is composed of at least three group members and a group controller. Other similar architectures include mesh based highly connected clients and also the conventional ring topology which is also referred to as conference communication. Although many researchers debate over the need for a specialized group controller, researchers generally agree that group communications can be more complex and insecure. The reason for this is that every group member is a potential point of attack for an attacker. Which implies that the group is only as strong as the weakest member in the group. Hence there will always be need for designing and implementing group communication protocols that are robust and reliable [1].

An emerging and promising technology in the field of security is ICMetrics. This technology attempts to provide an identity to computational devices. The technology exploits the fact that every device is unique in its internal environment hence unique characteristics can be used by hardware or software modules on the device to generate a unique identification for individual devices. The unique characteristics of a device can include measurable features like serial numbers, addresses, data in the program counter, data in the cache and many more. These unique features can be used to generate a number that is stable yet unique and can be used to provide identification for computation devices.

In this paper a novel filter based security scheme has been proposed that fulfils two security requirements, i.e. access control and key exchange for secure group communications. The paper proposes the use of a secure filter function that uses timestamps and hashing to generate a group key that provides access to legitimate group clients only. The proposed security filter is resilient against the man-in-the-middle attack.

This paper begins with an introduction to the ICMetrics technology followed by a study of other contributions in the area of group communications. An explanation of keying perspective and how they effect group communications is then presented. Section V provides a detailed description of the proposed secure filter function and how it operates. In the end we perform a standalone and comparative evaluation of the proposed security scheme using SHA1 and SHA2.

II. INTEGRATED CIRCUIT METRICS (ICMetrics)

ICMetrics is a unique solution because it allows a device to recognize itself based on its internal hardware and software environment. In conventional security systems related data such as keys are stored on the devices. This meant that if the key is compromised then the security of the system is also compromised [3]. ICMetrics advocates the use of unique device attributes for the generation of an ICMetric basis number that can then be used to generate security data. An advantage of this technique is that a client will never have to store the cryptographic keys on its system because they can be extracted using the ICMetric basis number. The ICMetric basis number is also discarded after use because whenever it is needed it can be regenerated. Hence it can be safely said that since security related data is not stored on the system it is possible to greatly reduce the chances of having a key compromise. Another mechanism which sets ICMetrics apart from other security techniques is that it is hardware and software based so any physical attempt to extract ICMetric data will render the ICMetric module inoperable, hence the system is protected from key leakages.

The ICMetric number is based on the use of feature sets that are specific to a system. Generation of the ICMetric number is a complex

operation because it involves operations that require change analysis, mean value deviation analysis and correlations between different feature sets. The ICMetric basis number can be generated by using two different techniques; both techniques have their positive and negative points. The choice of technique depends upon the required size and stability of the number. Both techniques require the extraction of features and the application of normalization maps to provide basis number stability. The first technique called the feature addition-combination technique allows the addition of individual feature values and hence generates a small yet stable basis number. The second technique is called the feature concatenation-combination because it generates a basis number by using the concatenation operation and hence generates a longer yet less stable basis number [3] [20].

III. STATE OF THE ART

Much work has been done to incorporate security in the classical one to one communication environment. Researchers often base their secure group communication design on extensions of one to one secure communications. Most research work has used variants of the basic Diffie Hellman key exchange [4] to provide security in groups. Diffie Hellman was designed for key exchange between only two entities. To use Diffie Hellman successfully in a group environment it had to be extended. Therefore many variants of the Diffie Hellman key exchange protocol have been suggested that provide security in group communications [5][6].

Steiner et al have proposed schemes that assist in group key generation in dynamic peer groups. Their proposed schemes target both initial key generation and auxiliary keying which is needed when a client(s) leaves or joins and existing group[7][8].

Research [18][19] has also been done to design data structures that can connect communication keys with individual clients. As the number of clients grows the task of the group controller becomes complicated and intensive. A solution to simplifying the task of the group controller is to use a tree based data structure that breaks down the entire group into small segments of multiple clients. The keys of each client form the leaf of the tree hence reducing the search time and also optimizing the rekeying operation.

Hashing algorithms allows the conversion an arbitrary strength to a fixed length string. This ability has been used to produce signatures relating to a file or a text. Common uses of hashing are for data corruption detection, fingerprinting of messages and comparison of large data objects [21]. Many hash algorithms have been proposed but the most commonly used are the SHA1 and the SHA2. Both algorithms are algorithmically very different from each other. The SHA1 algorithm produces a hash of 160 bit length while the SHA2 produces an output of 256 bit length [22].

Research has also been done on various applications of ICMetrics. One of the prominent is the integration of ICMetrics into an intelligent wheel chair formally known as the SYSIASS project [9]. Research has also been done on how ICMetrics can be incorporated into specialized areas like document archival in cloud computing [10] and also wireless sensor networks [17].Concurrent work is also underway that promotes the use of ICMetrics in one to one communications[11]. But until now no work has been done on the use of ICMetrics for secure group communications.

IV. KEYING PERSPECTIVES

Keying in group communications is divided into three categories namely centralized, distributed and decentralized [12]. The choice of category depends on the architecture, resource constraints and most of all level of security required. It is impossible to advocate a particular category because each has its positive and negative points. We will highlight each category briefly below.

A. Dictative Keying

In the dictative approach keying responsibilities are given to a group leader or a key generation controller. It is not necessary that the leader has to be a fixed entity. A client can be given the additional responsibility of being a group controller with privileges. The problem with the centralized technique is that the controller needs to be protected from attacks because if the controller is compromised then the group communication is dismantled. Furthermore, a monolithic architecture can be more devastating if an attack is successful on the group.

B. Contributive Keying

In the contributive keying protocols, clients are given the responsibility of generating a key from their individual contributions. This does not necessarily mean that there is no group controller. There are protocols that need a group controller for the distribution of keys once the key has been generated. A vast advantage of the distributed protocol is that it allows the generation of a contributive key which is generated by taking inputs from individual clients in the group. Security analysts have discovered that seemingly secure protocols falling in this category are more prone to the man-in-the-middle attack. Simple yet ingenious mechanisms can be devised to counter the man-in-the-middle attack.

C. Clustered Keying

In clustered keying protocols the entire group is divided into smaller clusters. Each cluster has its own controller and is responsible for key generation and distribution. The complexity involved in this technique is the selection/ election of the controller. Once fully operational, it has been observed that the protocol is more optimized because of the low key distribution latency. Considering an aggressors viewpoint, clustered keying provides a target rich environment by presenting multiple targets that can be attacked. We must also consider the fact that brining down a single client or controller results in partial success for the aggressor. Hence many potential targets can mean low aggression impact on the group.

V. PROPOSED WORK

The design of the secure filter is based on the following principles:

- The protocol is based on the use of ICMetrics. All security related data must be generated by using ICMetrics.
- Perfect Forward Secrecy – once a client leaves a group it must not have access to the group and its activities. This implies that it should not be able to guess the group keys after its departure from the group.
- Backward Secrecy –If a new client is admitted to a group then it should not have access to the previous keys and data of the group.

- Collusion freedom – the attacker should not be able to deduce the keys even if it manages to capture some of the communications of the group.
- Clients may join and leave a group as they wish.
- Scalable – the secure group keying should be scalable.
- The protocol should interoperate with existing technologies and techniques.

A. *Sequence of Flow*

The scheme is based on contributive computation from individual clients and the group controller. The group controller is responsible for collecting data from individual clients, perform computation on the data and then transmit the new computations back to the clients for generation of the group key. The scheme provides an access key and hence filters those entities that may not be part of the group.

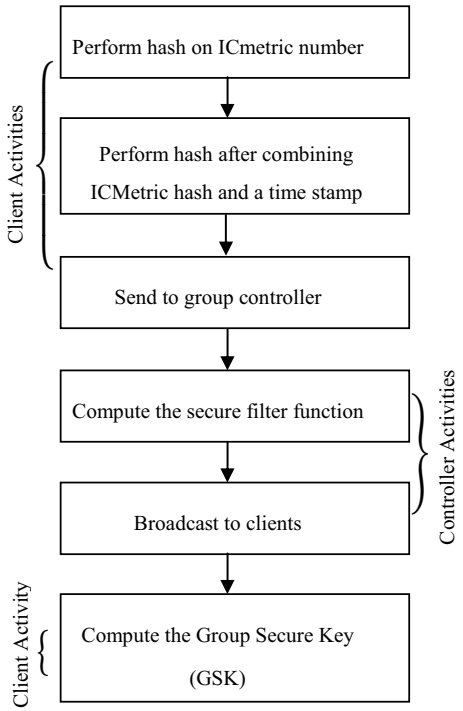

Fig. 1. Sequence of flow.

B. *One Way Hash Function*

Our proposed scheme is fundamentally based on a one way hash function [13][14] which possesses the property that if $h(x) = y$ is a hash function then it should not be computationally feasible to compute the message x given the value y. In our proposed scheme the hash has been used twice to increase the diffusion.

C. *Group Access Key Distribution*

In secure group communications, the important most stage is the generation and distribution of keying material. Since this protocol is a distributed protocol, our aim is to generate a key that can be generated using contributions from individual clients. The proposed secure filter function$f_C(x)$ is primarily based on the secret sharing scheme that gives correct results to only legitimate users. In the equation GSK_C is the group secure key. The positive integer i is a count of the number of clients that are present in the group communication. The secure function is composed of a hash function $h(\ \ )$which can be based on any hashing technique. The outer hash function takes as input a dividing combination composed of timestamp $ts$and the ICMetric basis number IC for client C_i.

$$f_C(x) = \begin{cases} GSK_C & ,\prod_{i=1}^{n}\left(x - h\left(ts_{C_i} | \, h(IC_{C_i})\right)\right) = 0 \\ undefined\ value, & \prod_{i=1}^{n}\left(x - h\left(ts_{C_i} | \, h(IC_{C_i})\right)\right) \neq 0 \end{cases} \quad (1)$$

When a group controller receives input from the individual clients it will perform the computation as shown in equation 2.

$$f_c(x) = \left\{\left(x - h\left(ts_{C_1} | \, h(IC_{C_1})\right)\right) \times \left(x - h\left(ts_{C_2} | \, h(IC_{C_2})\right)\right) \times \cdots \times \left(x - h\left(ts_{C_n} | \, h(IC_{C_n})\right)\right)\right\} + GSK_C \quad (2)$$

VI. RESULTS AND ANALYSIS

To prove the feasibility of an algorithm it is important to implement and study the scheme both from a security and computational standpoint. The scheme is composed of two hash function. First of all the inner most hash is computed on the ICMetric number. This is done so that the ICMetric basis number does not have to be transmitted in its original form. The second hash is computed by dividing the timestamp with the hash of the ICMetric number. The filter function is secure from the man-in-the-middle attack because to generate the key the attacker must have access to the time stamp. The time stamp is only known by the individual clients and is never broadcast. This also implies that the group communications are secure from basic impersonation attacks.As with any controller based scheme the only concern is that if the group controller is successfully attacked then future key generations will be hampered. Therefore having a stable and secure group controller is of great importance.

The simulations and evaluations were carried out on a 1.7GHz Intel 3rd generation CORE i5 system with 6GB RAM. The algorithm was programmed in C++ and the results have been analysed using Maple 13. Our analysis and previous studies [15] have shown that employing hash functions is computationally intensive and their use should be discouraged in systems with resource constraints. In group communications algorithmic complexity is very important, because for large size populations computationally intensive operations can prove to be counterproductive. We programmed the algorithm by using both SHA1 and SHA2. The results have shown that although SHA2 is more secure it requires significant computational effort, which make it impractical for large sized groups. To counter the extreme time

requirement of SHA2 we considered the use of SHA1 for the same group populations.SHA1is a lower effort algorithm but lacks in the provision of security because it generates a 160 bit hash which is prone to collisions and matching.

When the algorithm is run using SHA2 the average increase in the time as compared to SHA1 is 2021%. This number indicates the phenomenal increase in the amount of time taken by the algorithm to fully execute. The entire program was executed using SHA2 and the projected results can be seen in figure 2. Clearly as the population size increases then using the SHA2 is not feasible if time is of great importance.

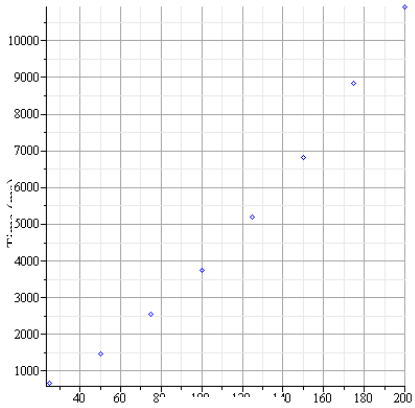

Fig. 2.Execution time (ms) using SHA2.

The same scheme was run by using the SHA1 with an identical population size and increment. We conclude that the time consumption is more acceptable when using SHA1. The projected graph using SHA1 is given in figure 3.

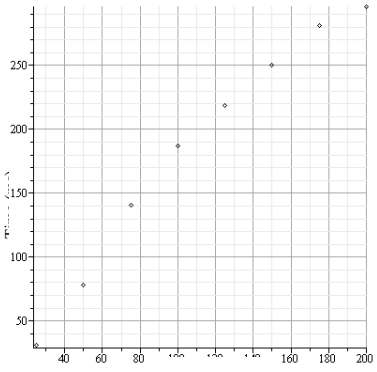

Fig. 3.Execution time(ms) using SHA1.

When comparing both techniques it is clear that SHA1 has a perfect advantage and outperforms SHA2 when considering time consumption. A merged comparative graph is given in figure 4 that shows the great increase in the time consumption for the same client population size. When comparing both algorithms, the SHA1 (represented by a point) has a much lower execution time as compared to the rapid increase demonstrated by the SHA2 (represented by cross) algorithm .One should not deduce by just considering the time consumption that SHA2 is incompatible with the secure function. SHA2 should be used with the proposed filter function if the population size is small (up to 100 clients) or if time can be sacrificed for added security.

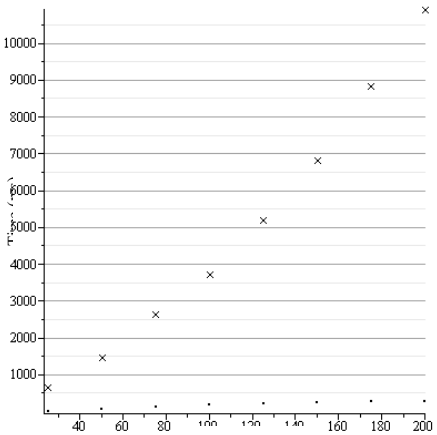

Fig. 4. Running time (ms) using SHA1(point plot) and SHA2(cross plot).

Huang and Medhi [16] propose a similar key distribution scheme for hierarchal mobile adhoc networks. Just like the scheme presented here, their scheme is also based on a decentralized key management infrastructure. Their scheme utilizes hashing for key generation hence the research provides an attractive comparison model. There are some points at which their scheme differs such as it utilizes hierarchies and trees which makes it more streamlined. Secondly the number of key derivation operations is$log_2 n - 1$ which shows the algorithmic efficiency of the scheme. The researchers have simulated their work using both SHA1 and SHA2. In their research they have only provided the time for a single run of the hash function. Our single run of the hash function takes 0.00094 seconds as compared to their 0.0003 seconds. We can conclude that using SHA1 in our scheme with ICMetrics is fully justifiable as the benefits of having ICMetrics greatly outweigh the drawbacks. Further SHA2 is computationally intensive and should only be used for small sized groups.

CONCLUSION

Secure group communications need special attention because multi-client environments are more prone to attacks. Provisioning security for multi-client environment is a delicate task because over provisioning can result in slowdowns at the group level. To fully test a secure group communication protocol it must be tested for large sized group populations because smaller sized populations do not effectively reflect the time requirements of a security scheme. In this paper a secure filter function has been proposed that is man-in-the-middle attack resilient. The filter function is based on a contributive secret sharing scheme that can be used for the generation and distribution of a group secure key.

To make the scheme more secure we have based the design on the latest ICMetrics technology. ICMetrics advocates the use of unique system attributes to generate a single ICMetrics number that can be used to identify a unique system. This ICMetrics number cannot be directly transmitted due to security concerns therefore in our scheme we have proposed passing the number through a hash function. The proposed security scheme is based on time stamping and hashing therefore to test its efficiency we have simulated the algorithm for large sized group populations using both SHA1 and SHA2. Both programmed hash algorithms possess unique attributes and the complexity of the SHA2 is reflected in the execution times of our proposed scheme. The results show that SHA1 with ICMetrics is fully scalable and can be practically used with the experimented maximum size of 200 participants. Whereas SHA2 is algorithmically complex and should be used for populations of upto 100 group members. To further study the protocol for performance we have done a comparative study against a protocol that has similar attributes but is not based on the ICMetrics technology. The results are insightful as a clear trade off can be made between the choice of hashing technique.

ACKNOWLEDGEMENT

This research is financially supported by the COALAS Project, http://www.coalas-project.eu/, 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] M. J. Moyer, J. R. Rao, P. Rohatgi, "A survey of security issues in multicast communications,"Network, IEEE, vol.13, no.6, pp.12-23, Nov.-Dec. 1999.

[2] D. Ramsbrock, R. Berthier, M. Cukier, "Profiling Attacker Behavior Following SSH Compromises", Dependable Systems and Networks, 2007. DSN '07. 37th Annual IEEE/IFIP International Conference on, pp.119-124, 25-28 June 2007

[3] 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, Sept 5-7, 2012.

[4] W. Diffie; M. Hellman, "New Directions In Cryptography, "IEEE Transactions of Information Theory, vol. 22, no. 6, pp. 644-654, Nov 1976.

[5] E. Bresson; O. Chevassut; D. Pointcheval, "The Group Diffie-Hellman Problems"Lecture Notes on Computer Science. Springer Berlin Heidelberg, vol. 2595, pp. 325-338, August 2002.

[6] E. Bresson, O. Chevassut, D. Pointcheval, "Dynamic Group Diffie-Hellman Key Exchange Under Standard Assumptions" Lecture Notes on Computer Science. Springer Berlin Heidelberg, vol. 23-32, pp. 321-336, May 2002.

[7] 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, Aug 2000.

[8] 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.

[9] 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, Sept. 2012.

[10] H. Tahir, R. Tahir, K. McDonald-Maier, "A Novel Private Cloud Document Archival System Architecture Based on ICmetrics," Emerging Security Technologies (EST), 2013 Fourth International Conference on, pp.102,106, 9-11 Sept. 2013.

[11] R. Tahir, K. McDonald-Maier, "An ICMetrics Based Lightweight Security Architecture Using Lattice Signcryption,"Emerging Security Technologies (EST), 2012 Third International Conference on, pp.135,140, 5-7 Sept. 2012.

[12] D. Devi; P. Ganapathi, "Secure Multicast Key Distribution for Mobile AdhocNetworks,"International Journal of Computer Science and Information Security, vol. 7, no. 2, pp. 218-223, 2010.

[13] I. Mironov, "Hash functions: Theory, attacks, and application,"Microsoft Research, 2005.

[14] NIST, Federal Information Processing Standards Publication 180-3, Secure Hash Standards (SHS), October 2008.

[15] B. Preneel, R. Govaerts, J. Vandewalle, "Cryptographic Hash Functions: an Overview" Proceedings of the 6th International Computer Security and Virus Conference (ICSVC 1993), vol. 19, 1993.

[16] D. Huang, D. Medhi, "A Secure Group Key Management Scheme for Hierarchical Mobile ad hoc Networks" Ad Hoc Networks, vol. 6, no. 4, pp. 560-577, June 2008. [Accessed 2014].

[17] R. Tahir, K. McDonald-Maier, "Improving Resilience Against Node Capture Attacks in Wireless Sensor Networks Using ICmetrics" Emerging Security Technologies (EST), 2012 Third International Conference on,pp.127-130, 5 Sept. 2012.

[18] J. Fan, P. Judge, M. H. Ammar, "HySOR: Group Key Management With Collusion-Scalability Tradeoffs Using a Hybrid Structuring of Receivers" Computer Communications and Networks, 2002. Proceedings. Eleventh International Conference on, vol., no., pp.196-201, 14-16 Oct. 2002

[19] H. Aslan, "A Scalable and Distributed Multicast Security Protocol Using a Subgroup-Key Hierarchy,"Computers & Security, Vol. 23, no. 4, 320-329, June 2004.

[20] 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.

[21] M. Singh, D. Garg, "Choosing Best Hashing Strategies and Hash Functions" Advance Computing Conference, 2009. IACC 2009. IEEE International, pp.50-55, 6-7 March 2009

[22] B. Coskun, N. Memon, "Confusion/Diffusion Capabilities of Some Robust Hash Functions,"Information Sciences and Systems, 2006 40th Annual Conference on, pp.1188-1193, 22-24 March 2006