Secure distribution of secret key to group nodes using SPSS model for Threshold Cryptography in MANETs

P.Swetha [a], P.Premchand [b]

[a] Associate Professor, JNTUH College of Engineering Jagityal, Dept. Of CSE, JNTU, TS, India

[b] Professor, University College of Engineering, Dept. of CSE, Osmania University, TS, India

Abstract

In mobile ad hoc networks, the intruder nodes can cause malfunctioning of the network as a consequence, the entire communication can be disrupted. Threshold cryptography is used\ to establish a secure communication among the nodes, where a secret key is distributed by a cryptographic function to multiple nodes as shares. These shares are used by the nodes for the reconstruction of the original secret key. This system only considers the condition when all the nodes taking part are authentic shareholders. Conventionally, the dealer (master node) and other nodes are trustworthy, however, this does not work every time. This paper proposes a new model for the group construction and picking a Group Head (master node), where the group head is responsible for the distribution of the secret key among the group members. This paper also proposes a Secure Proactive Secret Sharing SPSS method to identify the legitimate nodes for the secret key reconstruction using homomorphic commitments and for the encryption and decryption of messages. Secure Proactive Secret Sharing(SPSS)lets all shares of the secret key to be refreshed by producing a new set of shares for the same secret key. All shareholders need to cooperate with the PSS process for the protocol stability. In addition to the above, this paper also proposes a protocol for PSS synchronization.

KEYWORDS: Threshold Cryptography, Proactive Secret Sharing, Homomorphic Commitments, Verifiable Secret Sharing, Secret Reconstruction.

Introduction

A Mobile Ad hoc NETwork (MANET) is a organisation of wireless mobile nodes that dynamically establish arbitrary and temporary network topologies. People can be internetworked without a preexisting communication infrastructure. In the mobile ad hoc network, nodes can openly communicate with all the other nodes within their radio ranges; whereas nodes that not in the direct communication range use intermediary node(s) to communicate with each other. In these two circumstances, all the nodes that take part in the communication spontaneously form a wireless network, consequently this kind of wireless network can be viewed as mobile ad hoc network. Because of the unreliability of wireless links among nodes , frequently varying topology, lack of incorporation, the mobile ad hoc networks are more likely to suffer from the malicious activities than the traditional wired networks. Therefore, more attention to the security issues is required in the mobile ad hoc networks.

Security is the most desired feature in Mobile Ad hoc NETworks (MANET) [1]. The communication in a mobile network can be safeguarded by confirming that a secret key can be disclosed only to the two communicating parties. The most critical and composite issue is the dissemination of a secret group key [2] [3] to the authentic

nodes in a secure manner which is used to encrypt the data. For this Threshold Cryptography (TC) [4] [5] is used. In (n, t)TC, a secret key is distributed into 'n' shares using a cryptographic process and these shares are circulated to the nodes taking part in communication. The secret can be rebuilt only if the threshold number 't' out of 'n' shares are combined together. The distinct shares alone cannot serve the purpose of rebuilding the original secret key. A secret group key cannot be created with lesser than 't' shares. Nevertheless, there is a probability for a malicious node to create the secret key by stealing 't' or more shares from the participating nodes, withlengthyduration of time. Shamir's (*t ,n)* threshold secret sharing scheme[6] is a method in which the secret *s* is distributed to *n* shares by the master node and is shared among *n* shareholders in a fashion that (a) the secret can be recreated only if there are *t* or more than *t* shares; and (b) the secret cannot be completed if there are fewer than *t* shares. In the secret reconstruction, the nodes taking part can be either authentic shareholders or malicious nodes. It is vital that the shares are given to authentic nodes only.

This paper proposes a Secure Proactive Secret Sharing SPSS which illustrates how the group is constructed, how the Group Head (Master node) is elected for distributing the shares among the group members. An additional module, Proactive Secret Sharing PSS[7][8] lets a set of nodes holding shares to refresh all shares by producing a new set of shares for the original secret key from the old shares without rebuilding the secret key.

SPSSModel:

This model confers about the way the nodes in the network form the group and how the Group Head (Dealer) is picked who constructively allots the shares of the secret key amongst the group members. When the group formation is completed, the dealer has to execute operations like Key Generation, Key Dissemination and Reconstruction. Later, the keys are refreshed.

Group Construction

Consider a network of *N* nodes wherein, $N = \{ N_1, N_2, \ldots N_n \}$. Every node is certified by the Certificate Authority (CA) and the Certificates supplied to these nodes are created using ECDSA[9] method.

1. Initially every node in the network produces its own arbitrary binary value [0,1].
2. A Certificate authority (CA) [10] occasionally broadcasts its announcement to the whole nodes in the networks.
3. The nodes that are ready to take part in the communication replies to the announcement by sending a SEND_CERTIFICATE _REQUEST message to CA using its key and node_id.
4. CA produces the certificate which comprises of node_id, private key, public key, signature, hash value with the timestamp and sends to the demanded nodes.
5. Nodes save the received certificate.

All the nodes with their corresponding CAs form in group for having secure communication between the two entities.

Picking up Group Head (Dealer)

Upon group construction, all the nodes have to pick one node as the Group Manager or Group Head (Dealer), which holds the responsibility of Key Generation & Key Distribution.

1. All nodes in the group announces HELLO message with its own arbitrary value
2. Neighboring nodes obtains the announcements of other nodes and stores them with their corresponding arbitrary values
3. A node which has the highest arbitrary value amongst the neighboring nodes is picked as the Group Manager or Group Head (Dealer or master node)
4. The GH node publicizes its arbitrary value and its status as Dealer to entire nodes in the group.
5. The Dealer's node_id becomes its arbitrary value and the other nodes in the group are called as share holder nodes.
6. Dealer (GH) sends JOIN message to its shareholder where all nodes are combined to a group
7. As soon as all the nodes form in a group it starts request timer to request for shareholder.

Share Computation, Distribution and verification

The freshly picked Master Node or Group head or Dealer has to produce the secret key for distributing among the share holder nodes [11].

Let S_k be the secret key. The share holders holds the shares of S_k and the dealer D knows the secret of S_k. The dealer D distributes the shares of S_k amongN_1, N_2,….$N_n$andthe share holders nodes have to reconstruct the original secret key S_k.

Share generation

Let S_k be the secret used for sharing. The dealer D initiates the procedure by committing to S_k .The secret $S_k$has to be disseminated amongst$N$ shareholders where $N \in \{ N_1, N_2 \ldots N_n \}$.If any share holder$N_i$ismalicious, then it can input a incorrect share. In such case the reconstructed secret S_k will be in accurate. There is also a risk of malicious dealers who might give inaccurate shares to some share holder nodeN_i .Here Verifiable Secret Sharing (VSS) is used[12][13], where share holders commit to the shares they have obtained. A malicious share holder $N_i$can send incorrect share of the key S_k to any node and the accurate shares of S_k to others.

Key Distribution

Every user couldwork out a correct pair (s_{ki}, r_{ki}) on the committed polynomials. This method involves Pedersen's Commitments [14][15][16]. The dealer Drefers to the encrypted commitments along with their signatures:

$\rho\{C_{skj}\}$, $\rho\{C_{rkj}\}$, $\rho\{Commit_{skj}\}$ for all $j \in [1,n]$ and $\rho\{Commit_{sk}\}$ to every share holder.

Verification

Verifiable Secret Sharing VSS are built on Shamir's work. These systems allow shareholders to find whether the dealer provided them with valid shares of the secret or not, hence letting them to come to an agreement regarding whether the secret was

shared successfully. In this context, the dealer is not fully trusted; it does not disclose the secret, but it maytry to fool servers in accepting an incorrect share of the secret. Verifiable secret sharing is a significant module in several distributed secret sharing protocols having untrusted nodes since the protocols normally involve every server acting as a partially-trusted dealer to all of the others.

Every share holder N_i on getting the signed commitments [17][18] from *D*validates if the committed shares $\{commit_{skj}\}_{j\epsilon[1,n]}$ consists Shamir's sharing of the secret committed in $Commit_{sk}$ with threshold *t.* In specific, a share holder N_i takes $commit_{sk}$ and $\{Commit_{skj}\}_{j\epsilon[1,t]}$, homomorphically calculates the commitments of the share and random polynomial. Using the above commitments, the node N_i then calculates the commitments of the left over$n-t$ points and matches them with $\{commit_{skj}\}_{j\epsilon[t+1,n]}$,i.e., confirm if $\{commit_{skj}\}_{j\epsilon[1,n]}$ and $\{commit_{sk}\}$ define unique t-degree polynomials [19][20].The dealer *D*forms a signature $\rho_D$to support its claim of right sharing and broadcasts it. A share holder continues further merely on reception of a valid signature $\rho_D$from *D.* Each share holder$N_i\epsilon\ N$ on getting the broadcasted signature $\rho_D$checks the signature of *D*. On successful verification, if the trio { $\rho\{\square_{\square\square\square}\}$, $\square\ \{\square_{\square\square\square}\}$, $\square\ \{\square\square\square\square\square\square_{\square\square\square}\}$} [21] are received from *D*, then implement step 1 else step 2.

Step-1:Produce the share pair (s_{kj},r_{kj}) and send the trio { $\rho\ \{C_{skj}\}$, $\rho\ \{C_{rkj}\}$, $\rho\ \{commit_{skj}\}$} to N_j where $N_j\ \epsilon\ N$ and

s_{kj}=$D_{pkj}\{C_{skj}\}$

r_{kj}=$D_{pkj}\{C_{rkj}\}$

Step-2: Wait for the trio to be sent by some node. On receiving calculate the share pair (s_{kj},r_{kj}) as in step1. Send the share pair (s_{kj},r_{kj}) to eachN_j where $N_j\ \epsilon\ N$,check whether

$$commit_{skj} \stackrel{?}{=} commit(s_{kj}\ ,r_{kj})$$

$$commit_{skj}=f^{skj}g^{rkj}$$

If verification is successful, then enclosess_{kj} in the Session List (SL_j), elseeliminate the share holder from the Session List. Ultimately every true share holder will get*'t+1'* forwarded committed shares. By the property of homomorphic commitments [22] [23] each share holder will calculate the left over committed shares and$Commit_{sk}$, thus having all the necessary information of [s_k].

Reconstruction:

1) Every single node directs its share pair to all the nodes,which checks with the corresponding commits existing with the nodes
2) When*'t+1'* correct share pairs are established, the sharing polynomial, hence s_k is rebuilt.
3) If $|SL_j|$ = t+1, create a *t*-degree polynomial *f(x)* by interpolating$\{(j, s_{kj})\}_{skj\epsilon SLj}$.
   As a result*f(0)*= S_k where the secret key is rebuilt.

When the true dealer and true shareholders exists , the new session key is produced for encrypting the messages. Every shareholder has the partial session key, from which the session key is created for decrypting the message.

The essential requirement is that each share need not be revealed in the share transmission process; if *t* or more shares are stolen by malicious nodes in a long duration of time, the secret key is last lyproduced by them. Proactive Secret Sharing (PSS) reasonably offers the method to escape from threats of revealing the secret key [24][25] . PSS lets the nodes to refresh all shares by producing a new set of shares for the same secret key from the old shares without reconstructing the secret key, and then the old share is unusable after the refreshment of the share [26].
All the shares need be in consistent state when refreshing the new shares. That is, if the share holders are not consistent in the PSS process then certain share holders might use old shares while the others might use new shares. If this inconsistency is agreed then the secret key cannot be recreated. In order to accomplish this all share holders need be synchronized. For this a simple technique is used for initiating PSS.

Initiating PSS

1. Share holders asks the Dealer by sending a PSS_REQ request message with multicast address[27]
2. Dealer when receives PSS_REQ message checks the multicast address and computes the next sec interval and makes shareholders to wait by PSS_WAIT message
3. The dealer sends PSS_WAIT by encrypting with the session key and calculates hash value
4. Share holders gains the session key and decrypts the cipher by session key to check the data [28][29]
5. Later share holders starts waiting for nextSEC time interval

Share Refreshment

1. Now Share holders computes new share values and sends NOTIFY message to the dealer [30]
2. When the dealer receives NOTIFY from all the share holder itexecutes PSS_SUCCESS
3. Otherwise executes PSS_SUSPEND
4. Share holders periodically produces the to kenstoinitiate PSS procedure
5. Share holders obtains the initiated PSS broadcast and initiates PSS_REQ
6. When the PSS is completed all share holders jump to use its new subshares
7. If PSS is put off then all share holders halt rebroadcasting the token

Results

The SPSS protocol is simulated using ns-2. This protocol is executed for iterative share calculation and parallel share calculation. The results are depicted below.

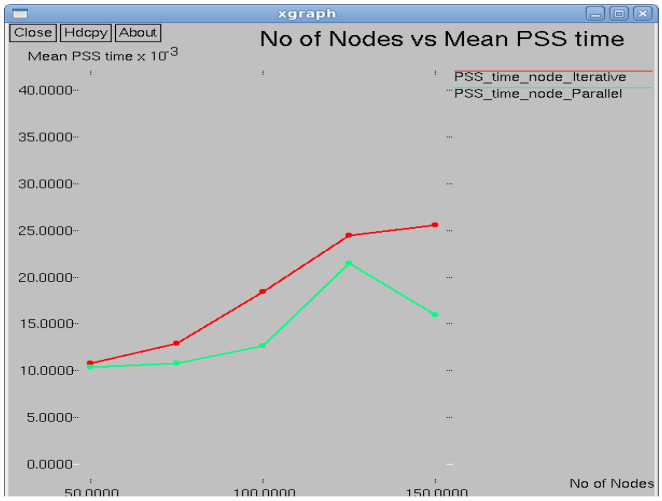

Fig.1 Nodes vs Mean PSS time

The graph for no. of nodes versus Mean PSS time shows that the Mean PSS time increases with the increase in nodes. The Mean PSS time is slightly more in iterative subshare calculation than parallel subshare calculation.

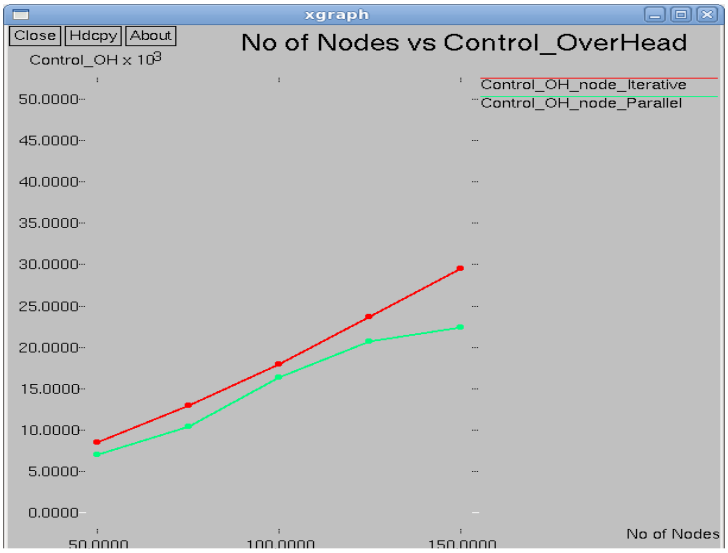

Fig.2 No. of Nodes Vs Control Overhead

The above graph indicates that as the no. of nodes increases, the control overhead increases in both parallel as well as iterative subshare calculation because of computation involved in performing operations like Key Generation, Key Dissemination and Reconstruction after which the keys are refreshed.

Conclusion

This paper illustrates the method to find true or authentic shareholders and the method for sharing secret key homomorphic commitments . The Threshold cryptography is implemented along with Proactive Sharing Scheme that permits the group nodes to refresh all shares by producing a new set of shares for the same secret key from the previous shares without rebuilding the secret key. Furthermore a new PSS synchronization pattern is used with a secret key sharing technique on top of threshold

cryptography. This is how the messages are decrypted using the collection of partial session keys. This paper shows that the proposed approach reduces in consistences compared to traditional refreshing scheme.

References

[1] L. Zhou and Z. J. Haas, Securing Ad Hoc Networks, IEEE Network Magazine, vol.13, no.6, Nov/Dec 1999.
[2] J. Hubaux, L. Buttyan and S. Capkun, The Quest for Security in Mobile Ad Hoc Networks, Proc. ACM MobiHOC, October 2001.
[3] M. Narasimha, G. Tsudik and J. Hyun, On the Utility of Distributed Cryptography in P2P and MANETs, Proc. IEEE ICNP, November 2003.
[4] Hitoshi Asaeda, Musfiq Rahman, Yoshihiro Toyama, Structuring proactive secret sharing in mobile ad-hoc networks ,IEEE 2006 1st International Symposium on Wireless Pervasive Computing 0-7803-9410-0/06/©2006 IEEE.

[5] P. Swetha, Implementation of Threshold Cryptography in MANETS, International Journal of Engineering Research & Technology (IJERT)Vol. 3 Issue 1, January – 2014 IJERT ISSN: 2278-0181

[6] A. Shamir, How to share a secret, Communication of the ACM, vol.22, 1979.

[7]. A. Herzberg, J. Stanislaw, H. Krawczyk, M. Yung, Proactive Secret Sharing or: How to Cope with Perpetual Leakage , Proc. 15th Annual International Cryptology Conference on Advances in Cryptology, 1995, pp. 339-352.
[8]. A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, Proactive secret sharing or: How to cope with perpetual leakage . In D.Coppersmith, editor, Advances in Cryptology—Crypto‘95, the 15th Annual International Cryptology Conference, Santa Barbara, CA USA, August 27–31, 1995, Proceedings, volume963 of Lecture. Notes in Computer Science. Springer, 1995.
[9] . L. Bassham, D. Johnson and T. Polk, Representation of Elliptic Curve Digital Signature Algorithm (ECDSA) Keys and Signatures in Internet X.509 Public Key Infrastructure Certificates, Internet Draft, June 1999. Available at http://www.ietf.org

[10] D. Eastlake, 3rd and P. Jones, ÒUS Secure Hash Algorithm 1 (SHA1), RFC3174, September 2001.
[11] E. Ben-Sasson, S. Fehr, and R. Ostrovsky. Near-Linear Unconditionally-Secure Multiparty Computation with a Dishonest Minority. In CRYPTO, pages 663–680, 2012.
[12] G. Bracha. An Asynchronous [(n-1)/3]-Resilient Consensus Protocol. In PODC, pages 154–162, 1984.
[13] C. Cachin, K. Kursawe, A.Lysyanskaya, and R. Strobl. Asynchronous Verifiable Secret Sharing and Proactive Cryptosystems. In CCS, pages 88–97, 2002.
[14] J.Brandt, I.Damga˚ard, P.Landrock and T.Pedersen: Zero-Knowledge Authentication Scheme with Secret Key Exchange, J.Cryptology, vol 11(1998), 147-160.
[15] M.Ben-Or, O.Goldreich, S.Goldwasser, J.H˚astad, J.Kilian, S.Micali and P.Rogaway: Everything Provable is Provable in Zero-Knowledge, Proceedings of Crypto 88, Springer Verlag LNCS series, 37–56.
[16] Blum, De Santis, Micali and Persiano: Non-Interactive Zero-Knowledge, SIAM J.Computing, Vol.20, no.6, 1991.

[17] R. Canetti. Studies in Secure Multiparty Computation and Applications. PhD thesis, The Weizmann Institute of Science, 1996.
[18] D. Chaum, C. Crépeau, and I. Damgård. Multiparty Unconditionally Secure Protocols. In STOC, pages 11–19, 1988.
[19]] B. Chevallier-Mames, P. Paillier, and D. Pointcheval. Encoding-Free Elgamal Encryption without Random Oracles. In PKC, pages 91–104, 2006.
[20] A. Choudhury, M. Hirt, and A. Patra. Asynchronous Multiparty Computation with Linear Communication Complexity. In DISC, pages 388–402, 2013.
[21]] P. Swetha , Dr. P. Premchand , " Secret Key Sharing Using Homomorphic Commitments and its application to Threshold Cryptography" , International Journal of Applied Engineering Research ISSN 0973-4562 Volume 11, Number 9 (2016) pp 6598-6602.
[22] J. C. Benaloh, "Secret Sharing Homomorphisms: Keeping Shares of a Secret Secret," Advances in Cryptology—6th Annual International Cryptology Conference (CRYPTO '86), Santa Barbara, 17-21 August 1987, pp. 251-260.
[23] E. F. Brickle and D. R. Stinson, "The Detection of Cheaters in Threshold Schemes," Advances in Cryptology—9th Annual International Cryptology Conference (CRYPTO '88), Santa Barbara, 21-25 August 1988, pp. 564-577.
[24] T. A. ElGamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," IEEE Transactions on Information Theory, Vol. 31, No. 4, 1985, pp. 469-472.
[25] H. Y. Lin and L. Harn, "Fair Reconstruction of a Secret," Information Processing Letters, Vol. 55, No. 1, 1995, pp. 45-47. http://dx.doi.org/10.1016/0020-0190(95)00045-E
[26] Wu, B., Wu, J., Fernandez, E., Magliveras, S., and Ilyas, M. (2005). Secure and Efficient Key Management in Mobile Ad HocNetworks. Proc. of 19th IEEE International Parallel &Distributed Processing Symposium, Denver.
[27] A. Yao. Protocols for secure computations. In FOCS, pages 160–164, 1982.
[28]] Wu, B., Wu, J., Fernandez, E., Ilyas, M., and Magliveras, S. (2005). Secure and Efficient Key Management Scheme in Mobile AdHoc Networks. Journal of Network and Computer Applications (JCNA).
[29] Stallings, W. (2002). Wireless Communication and Networks, Pearson Education.
[30] Nichols, R. and Lekkas, P. (2002). Wireless Security-Models, Threats, and Solutions, McGraw Hill, Chapter 7.
[31] Kaufman, C., Perlman, R., and Speciner, M. (2002). Network Security Private Communication in public World, Prentice HallPTR.