International Journal of Engineering Science Invention
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org Volume 2 Issue 11‖ November 2013 ‖ PP.15-21

Secret Key Sharing Scheme Based On Key Generation Centre For Authenticated Exchange Of Messages

B.V. Baiju
Department of Information Technology,Hindustan Institute of Technology & Science, Chennai, India

***ABSTRACT** : Group communication is becoming the basis for a growing number of applications. Group communication can benefit from a Key Generation Center to achieve a scalable exchange of messages. Key Generation Center can broadcast group key information to all group members at once under the secret shared with each user during registration and sends the cipher text to each group member separately. Key transfer protocols rely on a mutually trusted key generation center(KGC) to select session keys and transport session keys to all communication entities secretly. Most often, KGC encrypts session keys under another secret key shared with each entity during registration. In this paper, we propose an authenticated key transfer protocol based on secret sharing scheme that KGC can broadcast group key information to all group members at once and only authorized group members can recover the group key; but unauthorized users cannot recover the group key. The confidentiality of this transformation is information theoretically secure the strength of the proposed protocol is authentication and confidentiality. Security goals and analysis against inside and outside attacks can be achieved. . It also provide authentication for transporting this group key. Goals and security threats of our proposed group key transfer protocol will be analyzed in detail.*

***KEYWORDS:** Key Generation Center, Vandermonde matrix, Group key, Hash function, Encryption*

I. INTRODUCTION

When two entities are communicating with each other, and they do not want a third party to listen to their communication, then they want to pass on their message in such a way that nobody else could understand their message. This is known as communicating in a secure manner or secure communication. Secure communication includes means by which people can share information with varying degrees of certainty that third parties cannot know what was said. Other than communication spoken face to face out of possibility of listening, it is probably safe to say that no communication is guaranteed secure in this sense, although practical limitations such as legislation, resources, technical issues (interception and encryption), and the sheer volume of communication are limiting factors to surveillance. The purpose of this article is to describe the various means by which security is sought and compromised, the differing kinds of security possible, and the current means and their degree of security readily available. With many communications taking place over long distance and mediated by technology, and increasing awareness of the importance of interception issues, technology and its compromise are at the heart of this debate. For this reason, this article focuses on communications mediated or intercepted by technology. In this paper, I propose a solution based on this approach and provide confidentiality and authentication for distributing group keys. And only authorized group members can recover the group key; but unauthorized users cannot recover the group key. The confidentiality of this transformation is information theoretically secure.

II. PREVIOUS WORK

2.1 Diffie Hellman based Key Agreement Protocols

In key agreement protocols, all communication entities are involved to determine session keys. The most commonly used key agreement protocol is Diffie-Hellman (DH) key agreement protocol. In DH protocol, the session key is determined by exchanging public keys of two communication entities. Since the public key itself does not provide any authentication, a digital signature can be attached to the public key to provide authentication. However, DH public key distribution algorithm can only provide session key for two entities; not for a group more than two members.

2.2 Group Key Protocols

The class of centralized group key management protocols is the most widely used group key management protocols. Harney et al. Proposed a group key management protocol that requires O(n), where n is the size of group, encryptions to update a group key when a user is evicted or added if backward and forward secrecy are required. A set of scalable hierarchical structure-based group key protocols have been proposed.

Most distributed group key management protocols took natural generalization of the DH key agreement protocol, for example, Ingemarsson et al., Steer et al., Burmester and Desmedt, and Steiner et al. followed this approach. In 2006, Bohli developed a framework for robust group key agreement that provides security against malicious insiders and active adversaries in an unauthenticated point-to-point network. Then, in 2007, Bresson et al. constructed a generic authenticated group DH Key exchange and the algorithm is provably secure. Also, in 2007, Katz and Yung proposed the first constant-round and fully scalable group DH protocol which is provably secure in the standard model (i.e., without assuming the existence of "random oracles"). The main feature of the group DH key exchange is to establish a secret group key among all group members without relying on a mutually trusted KGC.Secret sharing has been used to design group key distribution protocols. There are two different approaches using secret sharing: one assumes a trusted offline server active only at initialization, and the other assumes an online trusted server, called the key generation centre, always active. The first type of approach is also called the key predistribution scheme. The main disadvantage of this approach is to require every user to store a large size of secrets. The second type of approach requires an online server to be active and this approach is similar to the model used in the IEEE 802.11i standard that employs an online server to select a group key and transport it to each group member.

III. PROPOSED PROTOCOL DESCRIPTION

In this project work, I propose a protocol based on this approach that provides confidentiality and authentication for distributing group keys. Furthermore, I classify attacks into insider and outsider attacks separately, and analyze our protocol under these attacks in detail

3.1 Group key transfer protocol using secret sharing scheme

The following are the unique features of our proposed group key transfer protocol using secret sharing scheme.

1) Each user needs to register at KGC to subscribe the group key transfer service and to establish a secret with KGC. Thus, a secure channel is needed initially to share this secret with each user. Later, KGC can transport the group key and interact with all group members in a broadcast channel.
2) The confidentiality of group key distribution is information theoretically secure; that is, the security of this transfer of group key to each group member does not depend on any computational assumption.
3) The authentication of the group key is achieved by broadcasting a single authentication message to all group members.

Group key transfer protocol relies on one trusted entity, KGC, to choose the key, which is then transported to each member involved. Each user is required to register at KGC for subscribing the key distribution service. The KGC keeps tracking all registered users and removing any unsubscribed users. During registration, KGC shares a secret with each user. In most key transfer protocol, KGC encrypts the randomly selected group key under the secret shared with each user during registration and sends the ciphertext to each group member separately. An authenticated message checksum is attached with the ciphertext to provide group key authenticity. In this approach, the confidentiality of group key is ensured using any encryption algorithm which is computationally secure. In cryptography, a key-agreement protocol is a protocol whereby two or more parties can agree on a key in such a way that both influence the outcome This protocol uses secret sharing scheme to replace the encryption algorithm. A broadcast message is sent to all group members at once. The confidentiality of group key is information theoretically secure. In addition, the authentication of broadcasting message can be provided as a group authentication. This feature provides efficiency of our proposed protocol.

3.2.1 Advantages

The main security advantages for our group key transfer protocol are: 1) key freshness; 2) key confidentiality; and 3) key authentication. Key freshness is to ensure that a group key has never been used before. Thus, a compromised group key cannot cause any further damage of group communication. Key confidentiality is to protect the group key such that it can only be recovered by authorized group members; but not by any un-authorized user. Key authentication is to provide assurance to authorized group members that the group key is distributed by KGC, but not by an attacker. In our protocol, I focus only on protecting group key information broadcasted from KGC to all group members. The service request and challenge messages from users to KGC are not authenticated. Thus, an attacker can impersonate a user to request for a group key service. In addition, attacker can also modify information transmitted from users to KGC without being detected. In security analysis, we will conclude that none of these attacks can successfully attack to authorized group members since attackers can neither obtain the group key nor share a group key with authorized group members. User/message authentication and key confirmation can be easily incorporated into our protocol since each user

has shared a secret key with KGC during registration. However, these security features are beyond the scope of our fundamental protocol.

IV. SYSTEM DEVELOPMENT

The proposed authenticated group key transfer protocol consists of three Processes: initialization of KGC, user registration, and group key generation and distribution. The detailed description is as follows:

A. Initialization of KGC.

The KGC randomly chooses two safe primes p and and compute n = pq. n is made publicly known.

B. User Registration

Each user is required to register at KGC for subscribing the key distribution service. The KGC keeps tracking all registered users and removing any unsubscribed users. During registration, KGC shares a secret, (xi , yi) with each user Ui,.

C. Group key generation and distribution

Upon receiving a group key generation request from any user, KGC needs to randomly selects a group key and access all shared secrets with group members. KGC needs to distribute this group key to all group members in a secure and authenticated manner. All communication between KGC and group members are in a broadcast channel. For example, we assume that a group consists of t members, (U1; U2; . . . ; Ut), and shared secrets are (xi, yi), for i = 1; . . . t.

The key generation and distribution process contains five steps.

Step 1: The initiator sends a key generation request to KGC with a list of group members as {U1, U2, . , U}.

Step 2: KGC broadcasts the list of all participating members, {U1; U2; . . . ; Ut}, as a response.

Step 3: Each participating group member needs to send a random challenge, Ri to KGC.

Step 4: KGC randomly selects a group key, k, and generates an interpolated polynomial f(x) with degree t to pass through (t + 1) points, (0, k) and (xi, yi XOR Ri), for i =1, . ,t. KGC also computes t additional points, Pi, for i = 1, ... t, on f(x) and Auth = h(k;U1; . . . ; Ut; R2; . . .;Rt; P1; . . . ; Pt), where h is a one-way hash function. All computations on f(x) are over Zn. KGC broadcasts Auth to all group members. All computations are performed in Zn.*Step 5:* For each group member, Ui, knowing the shared secret, (xi, yi XOR Ri), and t additional public points, Pi, for i = 1; . . . ; t, on f(x), is able to compute the polynomial f(x) and recover the group key k = f(0). Then, Ui computes h(k; U1; . . . ; Ut;R1; . . .; Rt; P1; . . . ; Pt) and checks whether this hash value is identical to Auth. If these two values are identical, Ui authenticates the group key is sent from KGC.

V. SYSTEM IMPLEMENATION

To perform this three main algorithms are used. They are SHA1, Lagrange Interpolation Polynomial Algorithm and Advanced Encryption Standard.

5.1 SHA1 – Secured Hash Algorithm

SHA1 is used for authentication. There are several similarities in the evolution of hash function and that of symmetric block ciphers. We have seen that the increasing power of brute-force attacks and advances in cryptanalysis have led to the decline in the popularity of DES and in the design of newer algorithm with longer key lengths and with features designed to resist specific cryptanalytic attacks. Newer hash algorithm have been developed with longer hash code length and with features designed to resist specific cryptanalytic attacks.

DES is based on the Feistel cipher, which in turn is based on the Substitution-permutation network proposal of Shannon. Similarly, most important modern hash functions follow the basic structure. This has proved to be a fundamentally sound structure and newer designs simply refine the structure and add to the hash code length. MD5, SHA-1, and RIPEMD-160. We then look at an internet-standard message authentication code.

Hash Function A hash function H is a transformation that takes a variable-size input m and returns a fixed-size string, which is called the hash value h (that is, h = H(m)). Hash functions with just this property have a variety of general computational uses, but when employed in cryptography the hash functions are usually chosen to have some additional properties.

The basic requirements for a cryptographic hash function are:

a. The input can be of any length.
b. The output has a fixed length.
c. H(x) is relatively easy to compute for any given x
d. H(x) is one-way.

e. H(x) is collision-free.

SHA is a cryptographic message digest algorithm similar to the MD4 family of hash functions developed by Rivest. SHA1, also known as SHA160, is a hash algorithm which was developed by the National Institute of Standards. SHA1 is commonly used on the internet to verify the integrity of software archives, as a unique identifier, and for digital signatures. The SHA takes a message of less than 264 bits in length. It is based on design of MD4 with a few key differences.

SHA-1 ALGORITHM
Step 1: Initialize variables:
h0 := 0x67452301
h1 := 0xEFCDAB89
h2 := 0x98BADCFE
h3 := 0x10325476
h4 := 0xC3D2E1F0

Step 2: Pre-processing:
append the bit '1' to the message
append k bits '0', where k is the minimum number >= 0 such that the resulting message length (in bits) is congruent to 448 (mod 512)
append length of message (before pre-processing), in bits, as 64-bit big-endian integer.

Step 3: Process the message in successive 512-bit chunks:
break message into 512-bit chunks for each chunk
break chunk into sixteen 32-bit big-endian words w[i], $0 \leq i \leq 15$

Step 4 : Extend the sixteen 32-bit words into eighty 32-bit words:
For i from 16 to 79
w[i] := (w[i-3] xor w[i-8] xor w[i-14] xor w[i-16]) leftrota
Step 5: Initialize hash value for this chunk:
a := h0
b := h1
c := h2
d := h3
e := h4

Main loop:
for i from 0 to 79
if $0 \leq i \leq 19$ then
f := (b and c) or ((not b) and d)
k := 0x5A827999
else if $20 \leq i \leq 39$
f := b xor c xor d
k := 0x6ED9EBA1
else if $40 \leq i \leq 59$
f := (b and c) or (b and d) or (c and d)
k := 0x8F1BBCDC
else if $60 \leq i \leq 79$
f := b xor c xor d
k := 0xCA62C1D6
temp := (a leftrotate 5) + f + e + k + w[i]
e := d, d := c
c := b leftrotate 30
b := a
a := temp

Step 6: Add this chunk's hash to result so far:
h0 := h0 + a
h1 := h1 + b

h2 := h2 + c
h3 := h3 + d
h4 := h4 + e

Step 7: Produce the final hash value (big-endian):
digest = hash = h0 append h1 append h2 append h3 append h4

5.2 Lagrange Interpolation polynomial algorithm

This algorithm is used for key generation. In numerical analysis, polynomial interpolation is the interpolation of a given data set by a polynomial: given some points, find a polynomial which goes exactly through these points. Polynomials can be used to approximate more complicated curves, for example, the shapes of letters in typography, given a few points. A relevant application is the evaluation of the natural logarithm and trigonometric functions: pick a few known data points, create a lookup table, and interpolate between those data points. This results in significantly faster computations. Polynomial interpolation also forms the basis for algorithms in numerical quadrature and numerical ordinary differential equations.
Suppose that the interpolation polynomial is in the form

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots\ldots + a_2 x^2 + a_1 x + a_{0.} \qquad (1)$$

The statement that p interpolates the data points means that

$$p(x_i) = y_i \qquad \text{for all } i \in \{0,1,\ldots,\ n\}.$$

If we substitute equation (1) in here, we get a system of linear equations in the coefficients ak. The system in matrix-vector form reads

$$\begin{bmatrix} x_0^n & x_0^{n-1} & x_0^{n-2} & . & . & . & x_0 & 1 \\ x_1^n & x_1^{n-1} & x_1^{n-2} & . & . & . & x_1 & 1 \\ . & . & . & & & & . & . \\ . & . & . & & & & . & . \\ x_n^n & x_n^{n-1} & x_n^{n-2} & . & . & . & x_1 & 1 \end{bmatrix} \begin{bmatrix} a_n \\ a_{n-1} \\ . \\ . \\ a_0 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ . \\ . \\ y_n \end{bmatrix}$$

It is necessary to solve this system for ak to construct the interpolant p(x). The matrix on the left is commonly referred to as a Vandermonde matrix. The condition number of the Vandermonde matrix may be large, causing large errors when computing the coefficients ai if the system of equations is solved using Gaussian elimination. These methods rely on constructing first a Newton interpolation of the polynomial and then converting it to the monomial form above.
Definition:
Given a set of k + 1 data points

$$(x_0, y_0),\ldots, (x_j, y_j),\ldots, (x_k, y_k)$$

where no two xj are the same, the interpolation polynomial in the Lagrange form is a linear combination of Lagrange basis polynomials

$$L(x) := \sum_{j=0}^{k} y_j l_j(x)$$

$$l_j(x) := \prod_{\substack{0<m<k \\ m \neq j}} \frac{x - x_m}{x_j - x_m} = \frac{(x - x_0)}{(x_j - x_0)} \cdots \frac{(x - x_{j-1})}{(x_j - x_{j-1})} \frac{(x - x_{j+1})}{(x_j - x_{j+1})} \cdots \frac{(x - x_k)}{(x_j - x_k)}$$

Note how, given the initial assumption that no two xi are the same $x_j - x_m \neq 0$ so this expression is always well-defined. The reason pairs $x_i = x_j$ with $y_i \neq y_j$ are not allowed is that no interpolation function L such that $y_i = L(x_i)$ would exist; a function can only get one value for each argument x_i. On the other hand, if also $y_i = y_j$, then those two points would actually be one single point.

For all $j \neq i$, $l_j(x)$, include the term $(x - x_i)$ in the numerator, so the whole product will be zero at $x = x_i$:

$$l_{j\neq i}(x_i) := \prod_{m\neq j} \frac{x_i - x_m}{x_j - x_m} = \frac{(x_i - x_0)}{(x_j - x_0)} \cdots \frac{(x_i - x_j)}{(x_j - x_i)} \cdots \frac{(x_i - x_k)}{(x_j - x_k)} = 0$$

On the other hand,

$$l_i(x_i) := \prod_{m\neq j} \frac{x_i - x_m}{x_j - x_m} = 1$$

In other words, all basis polynomials are zero at x = xi, except $l_i(x) = 1$, because it lacks the $(x - x_i)$ clause.It follows that $y_i l_i(x_i) = y_i$, so at each point x_i , $L(x_i) = y_i + 0 + 0 + \ldots + 0 = y_i$, showing that L interpolates the function exactly.

5.3 Advanced Encryption Standard Algorithm

AES algorithm is used for security services. AES is an iterated block cipher with a fixed block size of 128 and a variable key length. The different transformations operate on the intermediate results, called state. The state is a rectangular array of bytes and since the block size is 128 bits, which is 16 bytes, the rectangular array is of dimensions 4x4). The cipher key is similarly pictured as a rectangular array with four rows. The number of columns of the cipher key, denoted Nk, is equal to the key length divided by 32.

A state:

a0,0	a0,1	a0,2	a0,3
a1,0	a1,1	a1,2	a1,3
a2,0	a2,1	a2,2	a2,3
a3,0	a3,1	a3,2	a3,3

A key:

k0,0	k0,1	k0,2	k0,3
k1,0	k1,1	k1,2	k1,3
k2,0	k2,1	k2,2	k2,3
k3,0	k3,1	k3,2	k3,3

It is very important to know that the cipher input bytes are mapped onto the the state bytes in the order a0,0, a1,0, a2,0, a3,0, a0,1, a1,1, a2,1, a3,1 ... and the bytes of the cipher key are mapped onto the array in the order k0,0, k1,0, k2,0, k3,0, k0,1, k1,1, k2,1, k3,1 ... At the end of the cipher operation, the cipher output is extracted from the state by taking the state bytes in the same order. AES uses a variable number of rounds, which are fixed: A key of size 128 has 10 rounds. A key of size 192 has 12 rounds. A key of size 256 has 14 rounds.

During each round, the following operations are applied on the state:

1. SubBytes: every byte in the state is replaced by another one, using the Rijndael S-Box
2. ShiftRow: every row in the 4x4 array is shifted a certain amount to the left
3. MixColumn: a linear transformation on the columns of the state
4. AddRoundKey: each byte of the state is combined with a round key, which is a different key for each round and derived from the Rijndael key schedule

In the final round, the MixColumn operation is omitted. The algorithm looks like the following (pseudo):

```
AES(state, CipherKey)
{KeyExpansion(CipherKey, ExpandedKey);
AddRoundKey(state, ExpandedKey);
for (i = 1; i < Nr; i++)   { Round(state, ExpandedKey + Nb*i);
} FinalRound(state, ExpandedKey + Nb * Nr);
        }
```

Observations:

[1] The cipher key is expanded into a larger key, which is later used for the actual operations
[2] The roundKey is added to the state before starting the with loop
[3] The FinalRound() is the same as Round(), apart from missing the MixColumns() operation.
[4] During each round, another part of the ExpandedKey is used for the operations

[5] The ExpandedKey shall always be derived from the Cipher Key and never be specified directly.

VI. CONCLUSION

The proposed an efficient group key transfer protocol based on secret sharing. Every user needs to register at a trusted KGC initially and preshare a secret with KGC. KGC broadcasts group key information to all group members at once. Security analysis for possible attacks is also included. In the proposed protocol, we only focus on protecting group key information broadcasted from KGC to all group members. Here briefly explained that how to provide user authentication and authenticate messages transmitted from group members to KGC.
In future an improved authenticated key transfer protocol based on Shamir's secret sharing is used to achieve key confidentiality. Furthermore, the proposed scheme resists against both insider and outsider attacks.

REFERENCES

[1] I.F. Akyildiz and I.H. Kasimoglu, Wireless Sensor and Actor Networks: Research Challenges, Elsevier Ad Hoc Network J., vol. 2, 2004, pp. 351-367.
[2] K. Akkaya and M. Younis, Coverage-Aware and Connectivity- Constrained Actor Positioning in Wireless Sensor and Actor Networks, Proc. 26th IEEE Int'l Performance Computing and Comm. Conf. (IPCCC '07), Apr. 2007.
[3] K. Ozaki et al., A Fault-Tolerant Model for Wireless Sensor-Actor System, Proc. Second IEEE Int'l Workshop Heterogeneous Wireless Sensor Networks (HWISE '06), Apr. 2006.
[4] P. Basu and J. Redi, Movement Control Algorithms for Realization of Fault-Tolerant Ad Hoc Robot Networks, IEEE Networks, vol. 18, no. 4, Aug. 2004, pp. 36-44.
[5] K. Akkaya, M. Younis, and M. Bangad, Sink Repositioning for Enhanced Performance in Wireless Sensor Networks, Computer Networks, vol. 49, 2005, pp. 434-512.
[6] X. Liu, L. Xiao, A. Kreling, and Y. Liu, Optimizing Overlay Topology by Reducing Cut Vertices, Proc. ACM Int'l Workshop Network and Operating System Support for Digital Audio and Video (NOSSDAV '06), May 2006.
[7] K. Akkaya et al., Distributed Recovery of Actor Failures in Wireless Sensor and Actor Networks, Proc. IEEE Wireless Comm. and Networking Conf. (WCNC'08), Mar. 2008.