Protected cluster Communication Using AES Contributory Key Agreement

C.Lakshmi Devi[1], Mr.J S Ananda Kumar[2]
[1]*Student, Dept. of MCA, KMM Institute of Post Graduate Studies, Tirupati,A.P*
[2]*Assistant Professor, Dept. of MCA, KMM Institute of Post Graduate Studies, Tirupati,A.P*

***Abstract*- Many collaborative settings such as audio and videoconferencing, white-boards, clustering, and replication applications, require services which are not provided by the current network infrastructure. A typical collaborative application operates as a peer group where members communicate via reliable many-to-many multicast, sometimes requiring reliable ordered message delivery. In some settings, group members must be aware of the exact (agreed upon) group membership. Since group communication systems provide these services, many collaborative applications use group communication systems (GCS) as the underlying messaging infrastructure. Advanced encryption strategy protocols generate group keys based on contributions of all group members. Particularly appropriate for relatively small collaborative peer groups, these protocols are resilient to many types of attacks. Unlike most group key distribution protocols, advanced encryption strategy offer strong security properties such as key independence and perfect forward secrecy. We prove that it provides both Virtual Synchrony and the security properties of Group Diffie-Hellman, in the presence of any sequence of (potentially cascading) node failures, recoveries, network partitions, and heals. We implemented a secure group communication service, Secure Spread, based on our AES key agreement protocol and Spread group communication system.**

***Index Terms*- Security and protection, fault tolerance, network protocols, distributed systems, group communication, AES, cryptographic protocols.**

I. INTRODUCTION

Any cooperative settings such as audio and videoconferencing, white-boards, clustering, and replication applications, need services that aren't provided by the present network infrastructure. A typical cooperative application operates as a peer group wherever members communicate via reliable many-to-many multicast, sometimes requiring reliable ordered message delivery. In some settings, cluster members should remember of the precise (agreed upon) cluster membership. Since cluster communication systems give these services, several cooperative applications use cluster communication systems (GCS) because the underlying electronic messaging infrastructure. Security is crucial for distributed and cooperative applications that operate in an exceedingly dynamic network atmosphere and communicate over insecure networks reminiscent of the Internet. Basic security services required in such a bunch setting area unit mostly constant as in point-to-point communication: data secrecy and integrity, and entity authentication. These services cannot be earned while not secure, efficient, and sturdy cluster key management. Several vital applications (e.g., military and financial) need that every one internal communication stay confidential. Consequently, not only sufficiently advanced encryption should be used to shield intergroup messages, however the underlying cluster key management must conjointly give robust security guarantees. Group keys will be viewed as a sequence of values sorted by time of use, with every key adores a unique "snapshot" of a bunch. a bunch secret's modified whenever the cluster changes or a periodic rekey is required. The strongest familiar security guarantees area unit key independence and perfect forward secrecy (PFS). Key independence states that a passive adversary—who, within the worst case, might know all cluster keys except one—cannot use its knowledge to discover the one key that's missing. PFS demands that the compromise of cluster members' long keys should not lead to the compromise of any previously used cluster keys. Contributory group key agreement protocols that compute a group key as a (usually, one-way) function of individual contributions from all members can provide both key independence and

PFS properties. At the same time, contributory group key agreement presents a tough practical challenge: Its multiround nature must be reconciled with the possibility of crashes, partitions, and other events affecting group membership that can occur during the execution of the group key agreement. Therefore, this paper focuses on AES contributory group key agreement.

II. GROUP KEY MANAGEMENT

Traditional centralized key management relies on a single fixed key server to generate and distribute keys to the group. This approach is not well-suited for group communication systems that guarantee continuous operation in any possible group subset and any arbitrary number of partitions in the event of network partitions or faults. Although a key server can be made constantly available and attack-resistant with the aid of various fault tolerance and replication techniques, it is very difficult (in a scalable and efficient manner) to make a centralized server present in every possible group subset. We note that centralized approaches work well in one-to-many multicast scenarios since a key server (or a set thereof) can support continued operation within an arbitrary partition as long as it includes the source.

III.AES –ALGORITHM

Cryptography plays an important role in the security of data. It enables us to store sensitive information or transmit it across insecure networks so that unauthorized persons cannot read it. The basic unit for processing in the AES algorithm is a byte (a sequence of eight bits), so the input bit sequence is first transformed into byte sequence. In the next step a two-dimensional array of bytes (called the State) is built. The State array consists of four rows of bytes, each containing Nb bytes, where Nb is the block size divided by 32 (number of words). All internal operations (Cipher and Inverse Cipher) of the AES algorithms are then performed on the State array, after which its final value is copied to the output (State array is transformed back to the bit sequence). The input and output for the AES algorithm each consist of sequences of 128 bits (digits with values of 0 or 1). These sequences will sometimes be referred to as blocks and the number of bits they contain will be referred to as their length. The Cipher Key for the AES algorithm is a sequence of 128, 192 or 256 bits. The AES algorithm consists of ten rounds of encryption, as can be seen in Figure 3 First the 128-bit key is expanded into eleven so-called round keys, each of them 128 bits in size. Each round includes a transformation using the corresponding cipher key to ensure the security of the encryption.

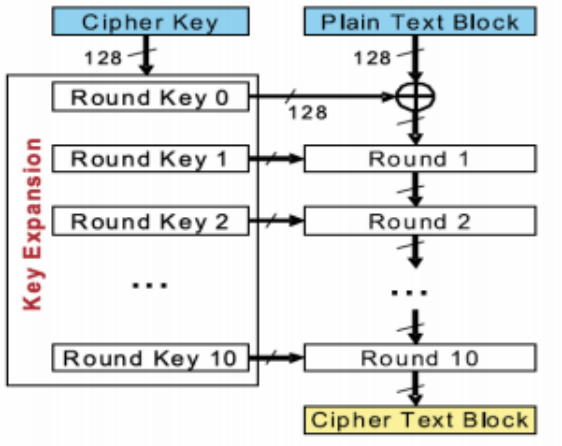

Figure1. AES Algorithm Structure

After an initial round, during which the first round key is XORed to the plain text (Addroundkey operation), nine equally structured rounds follow. Each round consists of the following operations: · Substitute bytes · Shift rows · Mix columns · Add round key The tenth round is similar to rounds one to nine, but the Mix columns step is omitted.

Substitute Bytes:
There are different ways of interpreting the Sub bytes operation. In this application report, it is sufficient to consider the Sub bytes step as a lookup in a table. With the help of this lookup table, the 16 bytes of the state (the input data) are substituted by the corresponding values.

Shift Rows
As implied by its name, the Shift rows operation processes different rows. A simple rotate with a different rotate width is performed. The second row of the 4x4 byte input data (the state) is shifted one byte position to the left in the matrix, the third row is shifted two byte positions to the left, and the fourth row is shifted three byte positions to the left. The first row is not changed.

Mix Columns

Opposed to the Shift rows operation, which works on rows in the 4x4 state matrixes, the Mix columns operation processes columns. Transformation in the Cipher that takes all of the columns of the State and mixes their data (independently of one another) to produce new columns.

Add Round Key

The Add round key operation is simple. The corresponding bytes of the input data and the expanded key are XORed.

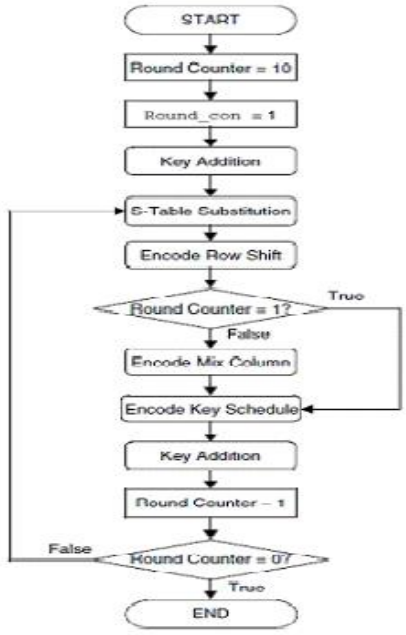

Figure :2 .AES Encryption Flow Chart

the flow chart of AES encryption algorithm round counter for 128 bit is 10. Then all other operation like key addition, S-Table Substitution, Encode Row Shift, Encode Mix Column are performed.

IV. CONCLUSION

In this paper, we have a tendency to show that, though troublesome, it is possible to harden security protocols to form them strong to asynchronous network events. Above all, we have a tendency to demonstrate how strong conducive key agreement protocols can be designed by taking advantage of cluster communication services. we have a tendency to presented 2 such strong protocols primarily based on the GDH key protocol suite and therefore the Virtual synchroneity group communication semantics. we have a tendency to conjointly showed however such protocols may be accustomed style secure cluster communication services and argued that, by group action them with a GCS supporting Virtual synchronicity, group communication membership and ordering guarantees are preserved. we have a tendency to exemplified by presenting Secure unfold, a client library that uses spread as its GCS and depends on a group key management protocol that's strong to method crashes and network partitions and merges and protects confidentiality of the information even once long-run keys of the participants are compromised.

REFERENCES

[1] W. Diffie and M.E. Hellman, "New Directions in Cryptography," IEEE Trans. Information Theory, vol. 22, pp. 644-654, Nov. 1976.

[2] G. Ateniese, O. Chevassut, D. Hasse, Y. Kim, and G. Tsudik, "Design of a Group Key Agreement API," Proc. DARPA Information Security Conf. and Exposition, Jan. 2000.

[3] Y. Amir, "Replication Using Group Communication over a Partitioned Network," PhD dissertation, Inst. of Computer Science, The Hebrew Univ. of Jerusalem, Israel, 1995.

[4] Y. Kim, A. Perrig, and G. Tsudik, "Simple and Fault-Tolerant Key Agreement for Dynamic Collaborative Groups," Proc. Seventh ACM Conf. Computer and Comm. Security, pp. 235-244, Nov. 2000.

[5] Y. Kim, A. Perrig, and G. Tsudik, "Communication-Efficient Group Key Agreement," Proc. Int'l Conf. Information Security IFIP SEC, June 2001.

[6] D. Steer, L. Strawczynski, W. Diffie, and M. Wiener, "A Secure Audio Teleconference System," Proc. Conf. Advances in Cryptology, Aug. 1990.

[7] M. Burmester and Y. Desmedt, "A Secure and Efficient Conference Key Distribution System," Proc. Conf. Advances in Cryptology, May 1994.

[8] D. Boneh, "The Decision Diffie-Hellman Problem," Lecture Notes in Computer Science, vol. 1423, pp. 48-63, 1998.

[9] H. Harney and C. Muckenhirn, "Group Key Management Protocol (GKMP) Specification," RFC 2093, July 1997.
[10] T. Hardjono, B. Cain, and I. Monga, "Intradomain Group Key Management Protocol," Sept. 2000.
[11] D. Harkins and N. Doraswamy, "A Secure Scalable Multicast Key Management Protocol (MKMP)," Nov. 1997.
[12] T. Ballardie, "Scalable Multicast Key Distribution," RFC 1949, 1996.
[13] Abdel-Karim R. Al Tamimi, "Security in Wireless Data Networks".
[14] William Stallings, "Cryptogrphy and network Security principles and practices",2007 pp 134-165.
[15] Tsang-Yean Lee, Huey-Ming Lee, Homer Wu, Jin-Shieh Su, "DataTransmission Encryption and Decryption Algorithm in Network Security".
[16] Kean, T. Duncan, A, "DES key breaking, encryption and decryption in wireless Communication"August 2008.