

On the Anti-Eavesdropping Broadcast

Steve Liu and Yiping Shen
Computer Science Department
Texas A&M University
College Station, TX 77843-3112

1 Abstract

In this paper, we present an anti-eavesdropping broadcast protocol for unicast and multicast communications. In our scheme, multicast groups need not transmit session keys over the network, but rather, use digital signatures to identify specific *secret-sharing* schemas collectively, so that nodes in the same group can determine session keys independently. When a node needs to initiate a communication session, it sends out the digital signature of a target group, and only nodes in the target group can determine the session key through the secret-sharing schema associated with the digital signature. After cold start communicating parties exchange data using only broadcast packets to conceal traffic patterns and counter traffic analysis attacks. This way, both unicast and multicast messages can be transmitted through an identical procedure.

To prevent the eavesdroppers from intercepting full messages too easily, we further shuffle and fragment messages before transmission. Message fragments are sent to the destination(s), and multiple intermediate nodes using broadcast packets, but only nodes that have proper keys can decode them. Shuffling rules are sent in a similar fashion as keys, so that only selected (intermediate) nodes can extract the needed fragments, encrypt remaining fragments (with another set of session keys,) and broadcast them again. Broadcast ceases in pre-determined rounds, and only the unicast/multicast recipients who have the initial correct secret-sharing values would be able to reassemble the plaintext message. Our scheme finds its applications in key distribution, anti-traffic-analysis group communications.

2 Background

Encrypted data exchange is the most widely adopted technology for private communications. Subject to regulatory and performance requirements, users need to choose a common encryption solution and acceptable key strengths to achieve the mission goals. While data encryption provides adequate protection for common applications, traffic patterns often unveil the nature of certain activities. For instance, the communication patterns of a command and control center could unveil the coordination patterns between the center and its subordinates. If eavesdroppers can predict presence of (on-line) key refreshing messages, they would have a much better chance to crack the coding system. In this paper, we propose a simple data transport technique to conceal traffic flow patterns. By fragmenting and dispersing a critical message along different paths, we increase the level of guesswork for the eavesdroppers to acquire complete messages.

Chiou and Chen [6] defined the "secure broadcasting problem." Moyer [8] proposed evaluation criteria about key management solutions. Most existing secure multicast schemes [1][6][10] assume that members in a multicast group share at least one initial secret to carry on

the following group key management or secure transmission sessions, but they do not consider the means for members to acquire this shared secret. Most designs assume that the encryption algorithm adequately protects the confidentiality of data communications.

Our scheme is designed to exchange critical information among the parties that has established cryptographic algorithms, authentication architecture and secret sharing schemes. It can work with both symmetric (IDEA or DES) and asymmetric (RSA 512 bits or elliptic curve) encryption algorithms, respectively. We assume that the *credential* of a *principal* (which can be either an end user or organization) is bound to a public key, on the basis of a public key infrastructure managed by some *trusted entities*. The trusted entities set guidelines, certify new principals, and validate the binding process. Most public key infrastructures, such as X.509 and PGP, follow this model, but differ in their operational details. We assume the use of a secret sharing scheme like that of Shamir [2] and Blakley [3], so that a recipient can recover the message when m -out-of- n of the message pieces, i.e., *shares* or *shadows*, become available. Different choices for the values of m and n reflect the tradeoff between security and reliability.

3 Anti-Eavesdropping Broadcast

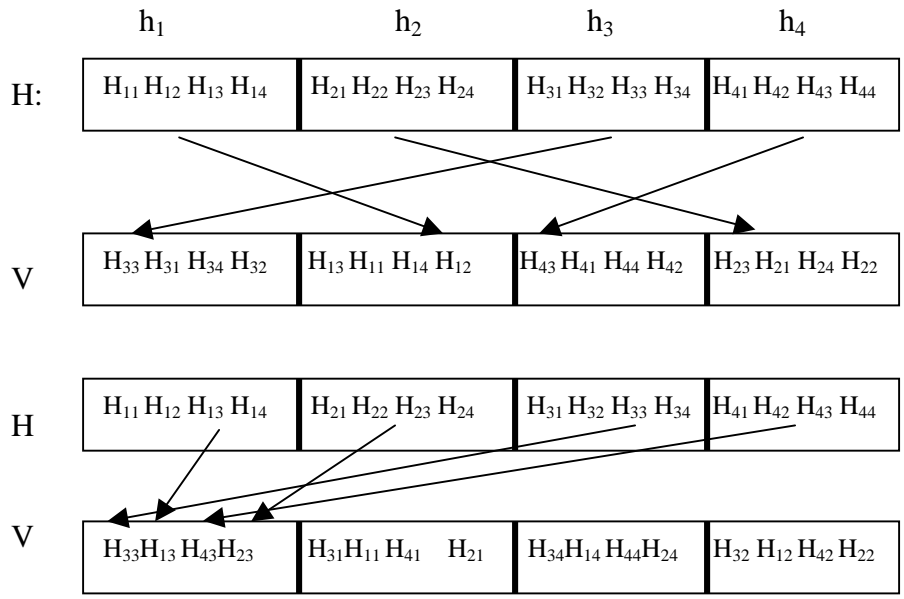
To prevent traffic analysis attacks, our approach is to map all unicast/multicast messages into broadcast packets, but only nodes that have the proper unicast/multicast keys can decode the packets into plaintext messages. Adequately dispersing unicast messages into fragment multicast packets will also smooth out the typical interaction bursts between hosts in order to conceal certain significant events. A reliable multicast (data dissemination) [5] service such as Muse [15], MDP[16], RMTP[17], MFTP [18] is used to broadcast the message. There are four major protocols in our scheme. Suppose that a broadcast server, denoted S , needs to broadcast a secret message M to n users. Let $U = \{U_1, U_2, \dots, U_n\}$ be a group of n users. When S needs to send a message, it first sends out the digital signature of a secret-sharing rule, so that only the selected subset of group members can determine the session key.

Protocol 1: Shared key generation

- Step 1. S randomly select a bulk of data G , S partitions the message G into n fragments G_1, G_2, \dots, G_n .
- Step 2. S calculates digests [19] $D_i = \text{Hash}(G_i), i=1,2,\dots,n$
- Step 3. S uses a secret sharing scheme [11] (k, n) with $D_1 \dots D_n$ as inputs, and the shared key is K
- Step 4. S sends its timestamp to all recipients' clocks and initialize broadcast channels. Recipients update local timestamps and acknowledge S .
- Step 5. S transfers G_i to U_j using an n complete bipartite matching graph, $i, j=1,2,\dots,n$. G_i is transmitted in packets $G_{i,1}, G_{i,2}, \dots, G_{i,n}$. After a recipient receives its fragment, it acknowledges S .
- Step 6. S retransmits lost packets to the corresponding recipients, until all recipients get the packets.
- Step 7. After S receives all the acknowledgements, it commands all recipients to start broadcast; U_j broadcasts G_i to $U_j (j=1,2,3,\dots,n \ j \neq i)$.
- Step 8. After U_j received from all other $n-1$ users, it repeats steps 2 and 3 to get the shared key K , $K = \text{Function1}(G, n)$.

Protocol 2: Fragmented broadcast

For simplicity, we illustrate our scheme using the perfect shuffling rule to demonstrate the shuffling procedure. Before we broadcast a message M , we encrypt M using K and shuffle the encrypted message. Suppose $H = E_K(M)$ and we use a shuffling rule $R = (r_1, r_2, r_3, \dots, r_n)$, in which position i is mapped to r_i , to shuffle the message fragments $H = (h_1, h_2, h_3, \dots, h_n)$, and $h_i = (h_{i1}, h_{i2}, h_{i3}, \dots, h_{in})$. The following figure depicts the shuffling rule (3142) for mapping of H and h . Let the shuffling outputs be denoted as $V = (v_1, v_2, v_3, \dots, v_n)$, we send V_i to U_i respectively ($i=1, 2 \dots n$).



Protocol 2 uses protocol 1 for data exchange, except that its input is $R(E_K(M))$, and each user U_i will calculate another shared key K' , $K' = \text{Function1}(R(E_K(M)), n)$, after they received all V_i from other $n-1$ users.

Protocol 3: Shuffling rule broadcast

S broadcasts $R' = E_{K'}(E_K(R))$ to $U_i (i=1, 2 \dots n)$ using protocol 1. After each user receives R' , it decrypts R' and gets the shuffling rule $R = D_{K'} D_K(R')$ using the two shared secret key K, K'

Protocol 4: Fragment relay and reassembly

After a node obtains K and R , it will be able to reconstruct the message M . In the example shown in the following figure, we have $A, B, C,$ and D in the same group, and communication links exist for $S \rightarrow A, S \rightarrow B, S \rightarrow C,$ and $S \rightarrow D$.

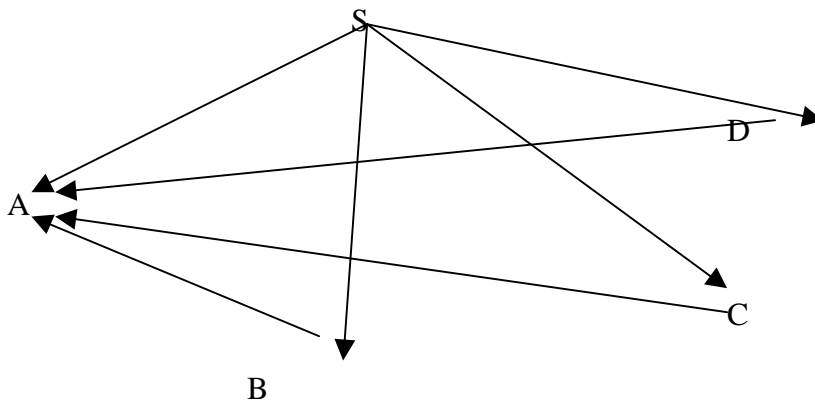


Figure 2. Two-hub relay of message fragments from S to A.

After S sets up the shared key K with A, B, C, and D, S begins to transfer message fragments to A, B, C and D. For A to recover the message, B, C, and D need to forward their message fragments to A. A then uses the decrypted R and K to assemble the fragments in sequence. To break a communication message from S to A, an eavesdropper needs to intercept and decode E(SA), E(SB), E(SC), and E(SD), the encryption key K, and the shuffling rule R, failing of any of the steps will not unveil the plaintext message.

4. Performance Analysis

The degree of protection provided by this scheme increases with the size of the broadcast group. Suppose that there are N nodes in the group, and the broadcast root S distributes secret information to n members. Even though we assume eavesdroppers know the entire topology, the possibility of the eavesdroppers know the n group members is:

$$\begin{aligned}
 & \text{Prob } (G_i | \text{ Given } n \text{ known nodes in the broadcast network}) \\
 & = 1 / (N! / (N-n)! * n!) = ((N-n)! * n!) / N! \\
 & = n! / N * (N-1) * (N-2) * \dots * (N-n+1) \\
 & \approx (n! / N^n) \\
 & \approx N^{-n} \quad (N \gg n)
 \end{aligned}$$

In addition to encryption protection, Eavesdroppers also need to know the phase sequences among n group members. In the worst case, if the attacker indeed controls all multiple common channels and get all information in protocols 1 or 2, he could crack the correct shared key K or K'. We could increase the protection by using protocol 1 multiple times to derive the real K, possibly with several other bogus keys. By sufficiently randomizing the transmission sequences of packets in each broadcast round, one could interleave packet transmission in different phases, making it difficult for eavesdroppers to differentiate the message fragments. Of course, by increasing the message sizes, we could also increase the randomness of data packets.

The packet confidentiality based on combined symmetric encryption (such as DES) and asymmetric encryption algorithm (RSA) is hard to crack within limited time. As a result, it is easy to protect authenticity of raw packets by setting the time-to-live parameters. Our proposed scheme can be verified using authentication specifications [4].

When eavesdroppers do not know the shuffling and fragmentation rules, this scheme increases the time complexity to the decryption task by polynomial order. Eavesdroppers can obtain the rules only if they get the shared key K, K' in protocols 1 and 2, and can control multicast channels in protocol 3. This could occur only if the eavesdroppers could correctly receive all packets in protocols 1 and 2 in all multicast channels.

5. Conclusion

In this paper, we proposed a traffic-concealing, anti-eavesdropping communication protocol for secure data exchanges. By using shared secrets and digital signatures, our scheme need not exchange keys over the network explicitly for data encryption. By using simple shuffling and ordering of message fragments, we disperse the interaction communication patterns among the multicast participants to counter eavesdropping and traffic analysis attacks. Our immediate next step is to consider other operational issues, such as membership changes, traffic pattern analysis, etc. to guarantee the quality of protection provided by this scheme.

6 References

- [1] Deng, R.H. Tjhung, T.T. "Novel approach to secure broadcast in distributed systems" Security Technology, 1995. Proceedings. Institute of Electrical and Electronics Engineers 29th Annual 1995 International Carnahan Conference on , 1995 .
- [2] Shamir, "How to share a secret", Communications of the ACM 22 (1979), 612-613.
- [3] G. R. Blakley, Safeguarding cryptographic keys, in "Proceedings of the National Computer Conference, 1979", American Federation of Information Processing Societies Proceedings 48 (1979), 313-317.
- [4] Lowe, G. A Hierarchy of Authentication Specifications. 10th IEEE Computer Security Foundations Workshop, 1997. June 10-12. Rockport, Massachusetts.
- [5] Katia Obraczka , Multicast Transport Protocols: A Survey and Taxonomy, Nov. 1997
- [6] Guang-Huei Chiou; Wen-Tsuen Chen "Secure broadcasting using the secure lock" Software Engineering, IEEE Transactions on , Volume: 15 Issue: 8 , Aug. 1989
- [7] Jan, J.K.; Yu, C.D." Yet another approach for secure broadcasting based upon single key concept" Security Technology, 1991. Proceedings. 25th Annual 1991 IEEE International Carnahan Conference on, 1991
- [8] Moyer, M.J.; Rao, J.R.; Rohatgi, P. " A survey of security issues in multicast communications " IEEE Network , Volume: 13 Issue: 6 , Nov.-Dec. 1999
- [9] C. A. Asmuth and G. R. Blakley. Pooling splitting and restituting information to overcome total failure of some channels of communications. In Proceedings of the 1982 Symposium on Security and Privacy, pages 156--169, New York, 1982. IEEE Society.
- [10] Chung Kei Wong; Gouda, M.; Lam, S.S. "Secure group communications using key graphs "Networking, IEEE/ACM Transactions on , Volume: 8 Issue: 1 , Feb. 2000
- [11] Shimshon Berkovits. "How to broadcast a secret". In D. W. Davies, editor,

- Advances in Cryptology -- EUROCRYPT 91, volume 547 of Lecture Notes in Computer Science, pages 535-541. Springer-Verlag, 8-11 April 1991.
- [12] Ballardie, A., "Scalable Multicast Key Distribution", RFC 1949, July 1996.
 - [13] R. Canetti and B. Pinkas, An updated version of A taxonomy of multicast security issues, internet draft draft-irtf-smug-taxonomy-00.txt, 2000.
 - [14] Brad Mann "HOW MANY TIMES SHOULD YOU SHUFFLE A DECK OF CARDS"
 - [15] K. Lidl, J. Osborne, and J. Malcolm. Drinking from the firehose: Multicast USENET news. Proceedings of the 1994 Winter USENIX Conference, 1994
 - [16] J. Macker and W. Dang. The multicast dissemination protocol (mdp) framework. Internet Draft, Internet Engineering Task Force, 1996
 - [17] J.C. Lin and S. Paul. Rmtp: A reliable multicast transport protocol. Proceedings of the IEEE INFOCOM'96 pages 1414-1424, March 1996
 - [18] K. Miller, K. Robertson, A. Tweedly, and M. White. Starburst multicast file transfer protocol (mftp) specifications. Internet Draft, Internet Engineering Task Force, January 1997
 - [19] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, MIT and RSA Data Security, Inc., April 1992.