ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Hybrid Network: Secure Data Retrieval Using Cipher Text Policy Attribute Based Encryption

N. Deshai, M.Tech
Assistant Professor,
Department Information Technology,
S.R.K.R Engineering College,
Bhimavaram.

P. Penchala Swamy
PG Scholar (M.Tech),
Department Information Technology,
S.R.K.R Engineering College,
Bhimavaram.

Dr. G. P. Saradhi Varma
B.Tech (CSE), M.Tech (CSE), Ph.D.(CSE),
Department CSE,
S.R.K.R Engineering College,
Bhimavaram.

ABSTRACT:

Encrypted information of broadcast is the scheme that a sender encrypts messages for a designated group of receivers, and sends the cipher texts by broadcast over the networks. Many research papers have done it using elliptic curve cryptography. In this paper, we propose the broadcast encryption scheme based on braid groups cryptography which is an alternative method in the public key cryptography and can reduce the computational cost. Here new ancient, a gathering of individuals arrange a typical open encryption key while every part holds an unscrambling key. A sender seeing people in general gathering encryption key can confine the unscrambling to a subset of individuals from his decision. We present a new BE scheme that is aggregately. The aggregatability property is shown to be useful to construct advanced protocols.

Keywords:

Broadcast encryption, Braid groups, Asymmetric group key Agreement, Contributory Broadcast Encryption, and Provable Security.

INTRODUCTION:

A.Fiat and M. Naor [1] first proposed the concept of broadcast encryption in 1993. In this scheme, sender allows to send a cipher text to some designated groups whose members of the group can decrypt it with his or her private key. However, nobody outside the group can decrypt the message. Broadcast encryption is widely used in the present day in many aspects, such as VoIP, TV subscription services over the Internet, communication among group members or from someone outside the group to the group members. This type of scheme also can be extended in networks like mobile multi-hop networks, which each node in these networks has limitation in computing and storage resources. However, neither conventional symmetric GKA nor the newly introduced asymmetric GKA allow the sender to unilaterally exclude any particular member from reading the plaintext. Hence, it is essential to find more flexible cryptographic primitives allowing dynamic broadcasts without a fully trusted dealer. This paper investigates a close variation of the above mentioned problem of one-round group key agreement protocols and focuses on “how to establish a confidential channel from scratch for multiple parties in one round”. We provide a short overview of some new ideas to solve this variation.

Asymmetric GKA Observe that a major goal of GKAs for most applications is to establish a confidential broadcast channel among the group. We investigate the potentiality to establish this channel in an asymmetric manner in the sense that the group members merely negotiate a common encryption key (accessible to attackers) but hold respective secret decryption keys. We introduce a new class of GKA protocols which we name asymmetric group key agreements (ASGKAs), in contrast to the conventional GKAs. A trivial solution is for each member to publish a public key and withhold the respective secret key, so that the final cipher text is built as a concatenation of the underlying individual ones. However, this trivial solution is highly inefficient: the cipher text increases linearly with the group size; furthermore, the sender has to keep all the public keys of the group members and separately encrypt for each member. We are interested in nontrivial solutions that do not suffer from these limitations. Group key agreement (GKA) is another well-understood cryptographic primitive to secure group-oriented communications. A conventional GKA allows a group of members to establish a common secret key via open networks. However, whenever a sender wants to send a message to a group, he must first join the group and run a GKA protocol to share a secret key with the intended members. More recently introduced asymmetric GKA in which only a common group public key is negotiated and each group member holds a different decryption key. However, neither conventional symmetric GKA nor the newly Introduced asymmetric GKA allow the sender to unilaterally exclude any particular member from reading the plaintext1. Hence, it is essential to find more flexible cryptographic primitives allowing dynamic broadcasts without a fully trusted dealer.

POTENTIAL APPLICATIONS:

A potential application of our ConBE is to secure data exchanged among friends via social networks. Since the Prism scandal , people are increasingly concerned about the protection of their personal data shared with their friends over social networks. Our ConBE can provide a feasible solution to this problem. Indeed, Phan et al. underlined the applications of our ConBE to social networks. In this scenario, if a group of users want to share their data without letting the social network operator know it, they can use our ConBE scheme. Since the setup procedure of our ConBE only requires one round of communication, each member of the group just needs to broadcast one message to other intended members in a send-and-leave way, without the synchronization requirement. After receiving the messages from the other members, all the members share the encryption key that allows any user to selectively share his/her data to any subgroup of the members. Furthermore, it also allows sensitive data to be shared among different groups.

Other applications may include instant messaging among family members, secure scientific research tasks jointly conducted by scientists from different places, and disaster rescue using a mobile ad hoc network. A common feature of these scenarios is that a group of users would like to exchange sensitive data but a fully trusted third party is unavailable. Our ConBE provides an efficient solution to these applications.

RELATED WORK:

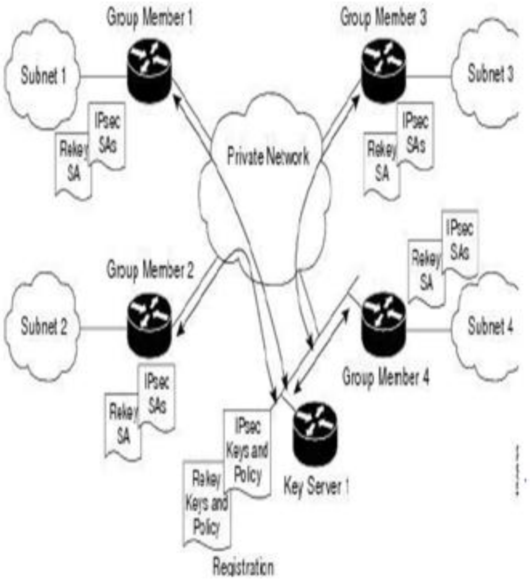
A number of works have addressed key agreement protocols for multiple parties. The schemes due to Ingemarsson et al. [2] and Steiner et al. are designed for n parties and require O(n) rounds. Tree key structures have been further proposed, reducing the number of rounds to O(log n) [8], [9], [10]. Multi round GKA protocols pose a synchronism requirement: in order to complete the protocol, all the group members have to stay online simultaneously. How to optimize the round complexity of GKA protocols has been studied in several works (e.g., [11], [12], [13]). In [14], Tzeng presented a constant-round GKA protocol that can identify cheaters. Subsequently, Yi [15] constructed a fault-tolerant protocol in an identity based setting.

Burmester and Desmedt [16] proposed a two-round n-party GKA protocol for n parties. The Joux protocol [17] is one round and only applicable to three parties. The work of Boneh and Silverberg [18] shows aoneround (n+1)- party GKA protocol with n-linear pairings. Dynamic GKA protocols provide extra mechanisms to handle member changes. Bresson et al. extended the protocol in to dynamic GKA protocols that allow members to leave and join the group. The number of rounds in the set-up/join algorithms of the Bresson et al.’s protocols is linear with the group size, but the number of rounds in the leave algorithm is constant. The theoretical analysis shows that for any tree-based group key agreement scheme, the lower bound of the worst-case cost is O(log n) rounds of interaction for a member to join or leave.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Without relying on a tree-based structure, Kim et al. proposed a two-round dynamic GKA protocol. Recently, Abdalla et al. presented a two-round dynamic GKA protocol in which only one round is required to cope with the change of members if they are in the initial group. Jarecki et al. presented a robust two-round GKA protocol in which a session key can be established even if some participants fail during the execution of the protocol. Observing that existing GKA protocols cannot handle sender/member changes efficiently, Wu et al. Presented a group key management protocol in which a change of the sender or monotone exclusion of group members does not require extra communication, and changes of other members require one extra round. BE is another well established cryptographic primitive developed for secure group communications.

SYSTEM ARCHITECTURE:



SECURITY PROPERTIES:

Confidentiality:

Communicated data is protected from non-members.

Sender authentication and non-repudiation:

Participants can authenticate message senders.

Membership dynamism:

It is possible to form groups and to add/remove participants.

Perfect Forward Security:

Compromise of long term keys of a member does not compromise earlier communication of that member.

Group Forward and Backward Secrecy:

Secrecy of new communication from revoked members, and old communication from new members.

GROUP KEY MANAGEMENT:

The new key management paradigm ostensibly requires a sender to know the keys of the receivers, which may need communications from the receivers to the sender as in traditional group key agreement protocols. However, some subtleties must be pointed out here. In traditional group key agreement protocols, the sender has to simultaneously stay online with the receivers and direct communications from the receivers to the sender are needed. This is difficult for a remote sender. On the contrary, in our key management paradigm, the sender only needs to obtain the receivers' public keys from a third party, and no direct communication from the receivers to the sender is required, which is implementable with exactly the existing PKIs in open networks. Hence, this is feasible for a remote sender. In our scheme, it is almost free of cost for a sender to exclude a group member by deleting the public key of the member from the public key chain or, similarly, to enroll a user as a new member by inserting that user's public key into the proper position of the public key chain of the receivers. After the deletion/addition of certain member, a new logical public-key ring naturally forms. Hence, a trivial way to enable this change is to run the protocol independently with the new key ring. If the sender would like to include a new member, the sender just needs to retrieve the public key of this user and insert it into the public key chain of the current receiver set.

By repeatedly invoking the member addition operation, a sender can merge two receiver sets into a single group. Similarly, by repeatedly invoking the member deletion operation, a sender can partition one receiver set into two groups. Both merging and partitioning can be done efficiently. In this module shows the deletion of member from the receiver group. Then, the sender and the remaining receivers need to apply this change to their subsequent encryption and decryption procedures.

CERTIFICATE AUTHORITY MODULE:

In this module, each receiver has a public/secret key pair. The public key is certified by a certificate authority, but the secret key is kept only by the receiver. A remote sender can retrieve the receiver's public key from the certificate authority and validate the authenticity of the public key by checking its certificate, which implies that no direct communication from the receivers to the sender is necessary. Then, the sender can send secret messages to any chosen subset of the receivers.

ALGORITHMS:

KeyGen(param;U) is an interactive protocol between the users in the set U. After the protocol run, it returns the public encryption key EK and a list Reg of the registered users with additional public information. Each user u 2 U eventually gets a secret decryption key dku.

Join(v; fu(dku)gu2U; Reg; EK) is an interactive protocol run between a user v and the set of users U, described in Reg. Each user takes as input his secret key and/or some random coins, the list Reg, and the encryption key EK. After the protocol, Reg and EK are updated, and each user (including v) has a secret decryption key.

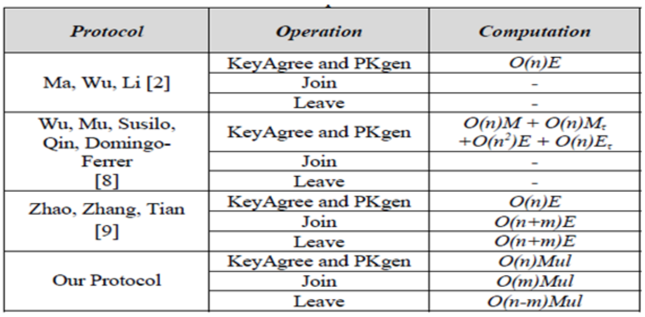
Enc(EK; Reg; S) takes as input the encryption key EK, the user register Reg, and a target set S. It outputs a key header H and a session key K 2 f0; 1gk.

Dec(dku; S;H) takes as input the target set S and a user decryption key dku together with a key header H. If dku corresponds to a recipient user, it outputs the session key K, else it outputs the error symbol ?.

The correctness requirement is that for all N, any target set S _ UN = [1;N] and for any u 2 UN, if u 2 S then the decapsulation algorithm gives back the key. A decentralized scheme requires that no authority is involved in the KeyGen and Join protocols.

COMPUTATION COST:

The computation cost is shown in Table 2. The values in the table are measured in Big-O notation. Our protocol has only multiplication in braid groups while the others have both multiplication in G or Gτ , and also exponentiation.



Protocol	*Operation*	*Computation*
Ma, Wu, Li [2]	KeyAgree and PKgen	$O(n)E$
	Join	-
	Leave	-
Wu, Mu, Susilo, Qin, Domingo-Ferrer [8]	KeyAgree and PKgen	$O(n)M + O(n)M_\tau$ $+O(n^2)E + O(n)E_\tau$
	Join	-
	Leave	-
Zhao, Zhang, Tian [9]	KeyAgree and PKgen	$O(n)E$
	Join	$O(n+m)E$
	Leave	$O(n+m)E$
Our Protocol	KeyAgree and PKgen	$O(n)Mul$
	Join	$O(m)Mul$
	Leave	$O(n-m)Mul$

n: the total number of members in the protocol; m: the number of members who want to join/leave the group; G: element in G; Gτ: element in Gτ ; M: multiplication (or division) in G; E: exponentiation in G; Mτ: multiplication (or division) in Gτ; Mul: multiplication in braid groups.

CONCLUSION:

We propose a broadcast encryption scheme based on braid groups cryptography. Our scheme is asymmetric group key agreement protocol and it is an encryption scheme in which the sender can broadcast an encrypted message over the networks by using his or her private key together with the public group key. The receivers which are the group members can decrypt it with their own private keys together with the public key of the sender. Our scheme makes the constant of cipher text and public key.

The computation cost of our scheme is only one serial number of braid group multiplication when a new member joins the group, and equal to n-2 when any member leaves the group.

REFERENCES:

[1] Qianhong Wu, Member, IEEE, Bo Qin, Lei Zhang, Member, IEEE, Josep Domingo-Ferrer, Fellow, IEEE Oriol Farr`as, and Jes´us A. Manj´on, "Contributory Broadcast Encryption with Efficient Encryption and Short Ciphertexts", IEEE Transactions On Computers, Vol. Xxx, No. Xxx, Xxx 2015.

[2] A. Fiat and M. Naor, "Broadcast Encryption," in Proc. Crypto 1993, 1993, vol. LNCS 773, Lecture Notes in Computer Science, pp. 480-491.

[3] I. Ingemarsson, D.T. Tang and C.K. Wong, "A Conference Key Distribution System," IEEE Transactions on Information Theory, vol. 28, no. 5, pp. 714-720, 1982.

[4] Q. Wu, Y. Mu, W. Susilo, B. Qin and J. Domingo-Ferrer, "Asymmetric Group Key Agreement," in Proc. Eurocrypt 2009, 2009, vol. LNCS 5479, Lecture Notes in Computer Science, pp. 153-170.

[5] http://en.wikipedia.org/wiki/PRISM surveillance program, 2014.

[6] Q. Wu, B. Qin, L. Zhang, J. Domingo-Ferrer and O. Farr`as, "Bridging Broadcast Encryption and Group Key Agreement," in Proc. Asiacrypt 2011, 2011, vol. LNCS 7073, Lecture Notes in Computer Science, pp. 143-160.

[7] D. H. Phan, D. Pointcheval and M. Strefler, "Decentralized Dynamic Broadcast Encryption," in Proc. SCN 2012, 2011, vol. LNCS 7485, Lecture Notes in Computer Science, pp. 166-183.

[8] M. Steiner, G. Tsudik and M. Waidner, "Key Agreement in Dynamic Peer Groups," IEEE Transactions on Parallel and Distributed Systems, vol. 11, no. 8, pp. 769-780, 2000.

[9] A. Sherman and D. McGrew, "Key Establishment in Large Dynamic Groups Using One-way Function Trees," IEEE Transactions on Software Engineering, vol. 29, no. 5, pp. 444-458, 2003.

[10] Y. Kim, A. Perrig and G. Tsudik, "Tree-Based Group Key Agreement," ACM Transactions on Information System Security, vol. 7, no. 1, pp. 60-96, 2004.

[11] Y. Mao, Y. Sun, M. Wu and K.J.R. Liu, "JET: Dynamic Join-Exit-Tree Amortization and Scheduling for Contributory Key Management," IEEE/ACM Transactions on Networking, vol. 14, no. 5, pp. 1128-1140, 2006.

[12] W.-G. Tzeng, "A Secure Fault-Tolerant Conference-Key Agreement Protocol," IEEE Transactions on Computers, vol. 51, no.4, pp. 373-379, 2002.

[13] X. Yi, "Identity-Based Fault-Tolerant Conference Key Agreement," IEEE Transactions Dependable Secure Computing vol. 1, no. 3, 170- 178, 2004.

[14] M. Burmester and Y. Desmedt, "A Secure and Efficient Conference Key Distribution System," in Proc. Eurocrypt 1994, 1994, vol. LNCS 950, Lecture Notes in Computer Science, pp. 275-286.

[15] A. Joux, "A One Round Protocol for Tripartite Diffie-Hellman," Journal of Cryptology, vol. 17, no. 4, pp. 263-276, 2004.

[16] D. Boneh and A. Silverberg, "Applications of Multilinear Forms to Crytography," Contemporary Mathematics, vol. 324, pp.71-90, 2003.