ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

Encompassing the Power of Data from Different Clients Hosted on Separate Networks

V.Thirumurugan[1] and P.Petchimuthu[2]

II ME (CSE), SCAD College of Engineering and Technology, Cheranmahadevi, Tamilnadu, India[1]

Professor, Department of Computer Science and Engineering, SCAD College of Engineering and Technology, Cheranmahadevi, Tamilnadu, India[2]

ABSTRACT: To ensure the securely, efficiently, and flexibility of data, the in-transit Protocol Data Units of data protocol the Consultative Committee for Network Data Systems Data Security Protocol (DSP) specification suggests four IPSec style security headers to provide four aspects of security services. However, this specification leaves key management as an open problem. Aiming to address the key establishment issue for data protocol, in this paper, we utilize a time-evolving topology model and two channel cryptography to design efficient and no interactive key exchange protocol. A time-evolving model is used to formally model the periodic and predetermined behaviour patterns of network, and therefore, a node can schedule when and to whom it should send its public key. Meanwhile, the application of two-channel cryptography enables nodes to exchange their public keys or revocation status information, with authentication assurance and in a no interactive manner. The proposed scheme helps to establish a secure context to support for DSP, tolerating high delays, and unexpected loss of connectivity of network.

KEYWORDS: Time evolving topology model, Two Channel cryptography, Storage and data sharing.

1. INTRODUCTION

Nowadays network storage is gaining more popularity. In enterprise settings, we see the rise in demand for data outsourcing, which assists in the strategic management of corporate data. It is also used as a core technology behind many online services for personal applications. Data from different clients can be hosted on separate virtual machines (VMs) but reside on a single physical machine. Data in a target VM could be stolen by instantiating another VM co resident with the target one. Regarding availability of files, there are a series of cryptographic schemes which go as far as allowing a third-party auditor to check the availability of files on behalf of the data owner without leaking anything about the data or without compromising the data owner's anonymity [1].

Likewise, network users probably will not hold the strong belief that the network server. There are two extreme ways for her under the traditional encryption paradigm:

- Alice encrypts all files with a single encryption key and gives Bob the corresponding secret key directly.
- Alice encrypts files with distinct keys and sends Bob the corresponding secret keys.

Data sharing in the Network is for the data owner to encrypt his data before storing into the Network, and hence the data remain information-theoretically secure against the Network provider and other malicious users. When the data owner wants to share his data to a group, he sends the key used for data encryption to each member of the group. Any member of the group can then get the encrypted data from the Network and decrypt the data using the key

In recent years, increased concern over personal information and privacy protection has led to the development of a number of techniques such as randomization and homomorphism encryption. Such techniques have been suggested to ensure that data mining can be performed while maintaining the preservation of private information protection. We consider privacy-preserving data mining, in which *m* parties collaborate to compute some function of their input, and no party in the system will learn any additional information separate from the functions and the parties' own input.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

Large numbers of conclusions in privacy-preserving data mining have been obtained by researchers, most of which are based on an assumption that each party is semi-honest. In particular, a party is deemed semi-honest when the party follows the protocol properly with the exception that it keeps a record of all its intermediate computation results and then tries to deduce further information in addition to the protocol result. Moreover, researchers also assume that every party does not collude or share its record with any other party. However, since intermediate messages are exchanged between parties throughout this method, these messages may be used to deduce private information of other parties.

II. PROBLEM STATEMENT

A key-aggregate encryption scheme consists of five polynomial-time algorithms as follows. The data owner establishes the public system parameter via Setup and generates a public/master-secret3 key pair via KeyGen. Messages can be encrypted via Encrypt by anyone who also decides what cipher text class is associated with the plaintext message to be encrypted. The data owner can use the master-secret to generate an aggregate decryption key for a set of cipher text classes via Extract. The generated keys can be passed to delegates securely (via secure e-mails or secure devices) finally any user with an aggregate key can decrypt any cipher text provided.

III. Encryption Data for Cryptography

A canonical application of KAC is data sharing. The key aggregation property is especially useful when we expect the delegation to be efficient and flexible. The schemes enable a content provider to share her data in a confidential and selective way, with a fixed and small ciphertext expansion, by distributing to each authorized user a single and small aggregate key.

3.1 Cryptographic Keys for a Predefined Hierarchy

We take the tree structure as an example. Alice can first classify the cipher text classes according to their subjects like Fig. 1. Each node in the tree represents a secret key, while the leaf node represents the keys for individual cipher text classes. Filled circles represent the keys for the classes to be delegated and circles circumvented by dotted lines represent the keys to be granted. Note that every key of the nonleaf node can derive the keys of its descendant nodes.
Fig. 1, if Alice wants to share all the files in the “personal” category, she only needs to grant the key for the node “personal,” which automatically grants the delegate the keys of all the descendant nodes (“photo,” “music”). This is the ideal case, where most classes to be shared belong to the same branch and thus a parent key of them is sufficient. However, it is still difficult for general cases. As shown in if Alice shares her demo music at work (“work”! “Casual”! “Demo” and “work”! “Confidential”! “demo”) with a colleague who also has the rights to see some of her personal data, what she can do is to give more keys, which leads to an increase in the total key size.

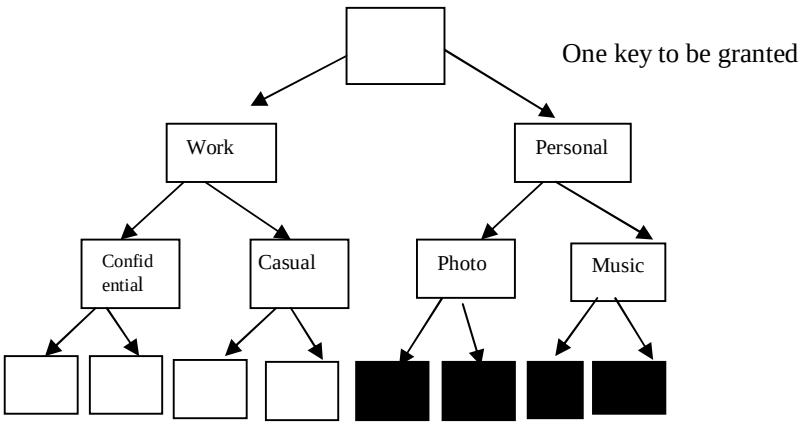

Fig 1. Hierarchy Structure

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

3.2 Compact Key in Identity-Based Encryption (IBE)

IBE is a type of public-key encryption in which the public-key of a user can be set as an identity string of the user. There is a trusted party called private key generator in IBE which holds a master-secret key and issues a secret key to each user with respect to the user identity. The encrypt or can take the public parameter and a user identity to encrypt a message. The recipient can decrypt this cipher text by his secret key.
To build IBE with key aggregation. One of their schemes assumes random oracles but another does not. In their schemes, key aggregation is constrained in the sense that all keys to be aggregated must come from different "identity divisions." While there are an exponential number of identities and thus secret keys, only a polynomial number of them can be aggregated. Most importantly, their key-aggregation comes at the expense sizes for both cipher texts and the public parameter, where n is the number of secret keys which can be aggregated into a constant size one. This greatly increases the costs of storing and transmitting cipher texts, which is impractical in many situations such as shared network storage. As we mentioned, our schemes feature constant cipher text size, and their security holds in the standard model.
It maintains each cipher text to be associated with an attribute, and the master-secret key holder can extract a secret key for a policy of these attributes so that a cipher text can be decrypted by this key.
In fuzzy IBE, one single compact secret key can decrypt cipher texts encrypted under many identities which are close in a certain metric space, but not for an arbitrary set of identities and, therefore, it does not match with our idea of key aggregation.

3.3 Other Encryption Schemes

- Setup
- KeyGen
- Encrypt
- Extract
- Decrypt

IV. KEY ENCRYPTION

4.1 Public key Encryption

If a user needs to classify his cipher texts into more than n classes, he can register for additional key pairs Since the new public-key can be essentially treated as a new user, one may have the concern that key aggregation across two independent users is not possible. It seems that we face the problem of hierarchical solution as reviewed but indeed, we still achieve shorter key size and gain flexibility as illustrated in the fig 2.
The fig 2 shows the flexibility of our approach. We achieve "local aggregation," which means the secret keys under the same branch can always be aggregated. We use a quaternary tree for the last level just for better illustration of our distinctive feature.
Our advantage is still preserved when compared with quaternary trees in hierarchical approach, in which the latter either delegates the decryption power for all four classes (if the key for their parent class is delegated) or the number of keys will be the same as the number of classes.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

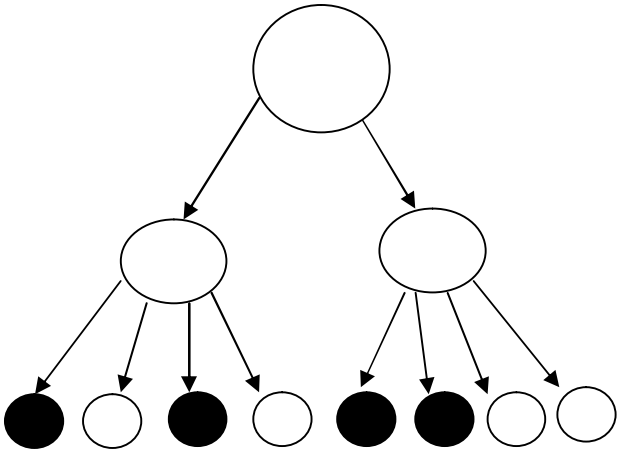

Fig 2. Key Assignment

We suggest that the cipher text classes for different purposes should be corresponded to different public-keys. This is reasonable in practice and does not contradict our criticism on hierarchical methods that an efficient assignment of hierarchy requires a priori knowledge on what to be shared.
This key extension approach can also be seen as a key update process the small aggregate key size minimizes the communication overhead for transferring the new key.

4.2 Private Key Encryption

For a concrete comparison, we investigate the space requirements of the tree-based key assignment approach we described in the fig1. This is used in the complete sub tree scheme, which is a representative solution to the broadcast encryption problem
If we grant the key one by one, the number of granted keys would be equal to the number of the delegated cipher text classes. With the tree-based structure, we can save a number of granted keys according to the delegation ratio. On the contrary, in our proposed approach, the delegation of decryption can be efficiently implemented with the aggregate key, which is only of fixed size, the delegation is randomly chosen. It models the situation that the needs for delegating to different users may not be predictable as time goes by, even after a careful initial planning. This gives empirical evidences to support our thesis that hierarchical key assignment does not save much in all cases.
Again, we confirm empirically that our analysis is true. We implemented the basic KAC system in C with the pairing-based cryptography (PBC). The execution times of Setup, Key Gen, and Encrypt are independent of the delegation where the number of cipher text classes is large but the non confidential storage is limited, one should deploy our schemes using the Type-D pairing bundled with the PBC. But we saved expensive secure storage without the hassle of managing a hierarchy of delegation classes.

V. MODULES

1. Homomorphic encryption Module.
2. Generalization Module.
3. Cryptography Module.
4. User and Admin Module

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

5.1 Homomorphic Encryption Modules

This module to use the first protocol is aimed at suppression-based anonymous databases, and it allows the owner of DB to properly anonymize the tuple t, without gaining any useful knowledge on its contents and without having to send to t's owner newly generated data. To achieve such goal, the parties secure their messages by encrypting them. In order to perform the privacy-preserving verification of the database anonymity upon the insertion, the parties use a commutative and homomorphic encryption scheme.

5.2 Generalization Module

In this module, the second protocol is aimed at generalization-based anonymous databases, and it relies on a secure set intersection protocol, such as the one found in, to support privacy-preserving updates on a generalization based k-anonymous DB.

5.3 Cryptography Module

In this module, the process of converting ordinary information called plaintext into unintelligible gibberish called cipher text. Decryption is the reverse, in other words, moving from the unintelligible cipher text back to plaintext. A *cipher* (or) *cipher* is a pair of algorithms that create the encryption and the reversing decryption. The detailed operation of a cipher is controlled both by the algorithm and in each instance by a *key*. This is a secret parameter (ideally known only to the communicants) for a specific message exchange context.

5.4 User and Admin Module

In this module, to arrange the database based on the patient and doctor details and records. The admin to encrypt the patient reports using encryption techniques using suppression and generalization protocols.

VI. RELATED WORK

There have been many proposals in the area of privacy preserving data mining recently. It proposed an algorithm to securely compute the weighted average problem (WAP) for the 2-party privacy-preserving K-means algorithm [7] extended WAP to multi-party case. In Mert's method, if some parties collude, they can easily deduce the private information of other parties. It proposed another solution of computation of the ratio of (two) summations of data spreading around *m* parties to solve the problem of privacy-preserving naive Bayes. The result can be securely computed without revealing any private information if there is no collusion. Unfortunately, if some of parties collude, privacy may be revealed.

VII. SECRET SHARING AND SECRECY STRUCTURES

The extract threshold conditions for mixed modules under which secure MPC is possible including fail corruption as a third corruption type. Here we state the result without considering fail corruption. Let be the number of players that have been actively and passively corrupted. This is the special case of Theorem. This strictly improves on non-mixed threshold result [3].

A secret Sharing scheme is usually specified by the so-called access structures. If secret is shared according to a certain scheme then the secrecy structure remain unchanged.

Because of the simplicity of the presented protocols, it is easy to verify that their complexity is polynomial. Although for a threshold adversary the complexity is exponential, the very small number of player s the protocol is very efficient and can possibility lead to the preferred protocol from a practical view point.

The general interest in secure MPC is to design protocols that are efficient in the size of the description of the secrecy and the adversary structures. The task depends on which type of description one uses i.e. on the adversary specification language.

VIII. CONCLUSION

To build privacy into mobile health systems with the help of the private network. We provided a solution for privacy-preserving data storage by integrating a PRF based key management for unlink ability, a search and access pattern hiding scheme based on redundancy, and a secure indexing method for privacy-preserving keyword search. We also

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

investigated techniques that provide access control and audit ability of the authorized parties to prevent misbehaviour, by combining ABE-controlled threshold signing with role based encryption.

REFERENCES

[1] Cheng-Kang Chu, Sherman S.M. Chow et al, "Key-Aggregate Cryptosystem for Scalable Data Sharing in Network Storage", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25, NO. 2, February 2014.
[2] T. Ballardie , J. Crowcroft, "Multicast-specific security threats and counter-measures", Proceedings of the 1995 Symposium on Network and Distributed System Security (SNDSS'95), p.2, February, 1995
[3] Ueli Maurer, "Secure Multi-Party Computation Made Simple", Discrete Applied Mathematics, Volume 154, Issue 2, Pp 370–381, February 2006.
[4] Sandro Rafaeli and David Hutchison, "A Survey of Key Management for Secure Group Communication" ACM Computing Surveys (CSUR), Volume 35, Issue 3, Pp 309-329, September 2003.
[5] D. Boneh and M. Franklin. "Identity-based encryption from the Weil pairing", SIAM J. Computing, Vol.32, Pp.586-615, 2003.
[6] C. Gentry and A. Silverberg, "Hierarchical ID-based cryptography", In Proceedings of Asiacrypt 2002, volume 2501 of LNCS, pages 548-66. Springer-Verlag, 2002.
[7] Mert Ozarar and Attila Ozgit, "Secure multiparty over-all mean computation via oblivious polynomial evaluation", International Conference on Security of Information and Networks, 2007..
[8] J. Birget, X. Zou, G. Noubir, and B. Ramamurthy, "Hierarchy-based access control in distributed environments". In ICC Conference 2001,
[9] H. Liaw, S. Wang, and C. Lei. "A dynamic cryptographic key assignment scheme in a tree structure", Computers and Mathematics with Applications, Vol.25(6), Pp.109–114, 1993.
[10] S.S.M. Chow, Y.J. He, L.C.K. Hui, and S.-M. Yiu, "SPICE – Simple Privacy-Preserving Identity-Management for Network Environment", Proc. 10th Int'l Conf. Applied Cryptography and Network Security (ACNS), vol. 7341, pp. 526-543, 2012.
[11] S.S.M. Chow, J. Weng, Y. Yang, and R.H. Deng, "Efficient Unidirectional Proxy Re-Encryption," Proc. Progress in Cryptology (AFRICACRYPT '10), vol. 6055, pp. 316-332, 2010
[12] Y. Sun and K.J.R. Liu, "Scalable Hierarchical Access Control in Secure Group Communications," Proc. IEEE INFOCOM '04, 2004.
[13] D. Boneh, C. Gentry, and B. Waters, "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys," Proc. Advances in Cryptology Conf. (CRYPTO '05), vol. 3621, pp. 258-275, 2005.
[14] J. Horwitz and B. Lynn. Toward hierarchical identity-based encryption. In Proceedings of Eurocrypt 2002, Vol 2332 of LNCS, pages 466-81. Springer-Verlag, 2002.