Security into Cloud Storage with Adding Integrity and Timed Ephemerizer

Surbhi Sahu
Mtech CSE
Jayoti Vidyapeeth
Womens University
Jaipur, Rajasthan

Amandeep Saini
Mtech CSE
Jayoti Vidyapeeth
Womens University
Jaipur, Rajasthan

Sarvesh Kumar
Assistant Professor
Jayoti Vidyapeeth
Womens University
Jaipur, Rajasthan

Sonali Rathod
Mtech CSE
Jayoti Vidyapeeth
Womens University
Jaipur, Rajasthan

ABSTRACT
Today cloud computing technology is very popular .there are several reason for using cloud computing. Cloud computing used for internet based computing solution. There are many computer work together into cloud computing. With the help if cloud computing, the individual can now run the application from anywhere. Cloud user does not purchase any software or buy software licence. User can take other benefits including scalability, reliability and efficiency. Ephemerizer is concept for developing security into cloud storage at any storage device. It is ensure that data completely destroyed. If any sensitive data not deleted so it will be create difficulty. This paper is concern about security of sensitive data into cloud storage. Data into cloud storage is absolute deletion become difficult because if we deleted data, it copies may be remain on backup media. We show that ephemerizer protocol work with finite amount of time with integrity property.

Data into cloud storage is absolute deletion become difficult because if we deleted data, it copies may be remain on backup media. We show that ephemerizer protocol work with finite amount of time with integrity property.

Keywords-
Cloud storage, integrity, time server, data consumer, data generator.

1. INTRODUCTION
Rapid growth of information technology has greatly facilitated individuals to generate and store information. Cloud storage provides an abstraction of unlimited storage space for client to host data. For example any social for social networking sits or your e-mail account. Client shares their photos, videos, or any other sensitive data. Individuals can benefits from cloud storage because device such as mobile or any moving device have limited Storage space so client stored their data on cloud. Confidentiality prevents sensitive data so therefore Integrity is very important to when transmit the data over internet. Integrity involves accuracy and consistency. When any data send to receiver then intruder change or corrupt this data. To protect data from legitimate leakage, ephemerizer has defined some solution. One solution is that to only store the data in encrypted form and delete its key. Timed ephemerizer will provide assured deletion for sensitive data. In this protocol, data is encrypted through public key from data consumer and ephemerizer. Encrypted data reside in persistent storage device of data consumer. If data consumer wants to recover data, it can decrypt text with the help of ephemerizer.

Fig.1 cloud storage where user can store their sensitive data

2. THE TIMED EPHEMERIZER'S PROTOCOL
The ephemerizer protocol is a communication protocol that allows any user to send one message protected by an expiration time. Receiver shall be able to access the message before expiration time. This is not possible after time reached. So a trusted third party is needed to provide $K_{eph\ i.e.}$ ephemeral key. If receiver wants a decryption key then he must ask ephemerizer. The protocol is displayed in figure1 where K key is a receiver's public key. A timed ephemerizer protocol define the data will be available during pre define time interval after this time data will be not recover. A timed ephemerizer explicitly after pre define initially disclosure.

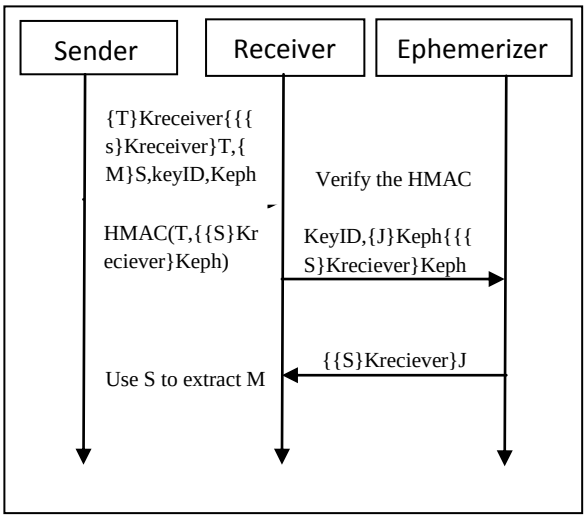

Fig.2. Ephemerizer scheme and T, S J is symmetric keys.

3. THE ALGORITHM DEFINITION
Adding integrity into timed ephemerizer protocol involves four type of entity.

1. Ephemerizer- which is trusted to revoke and publish ephemeral private and public key pair and try to decrypt the cipher text.

2. Data consumer-a data consumer try to access data before initial disclosure time.

3. Data generator- it define lifecycle of data in which will make its data available to data consumer.

4. Time server- it is most important entity of this protocol time server will publish time stamp periodically and concerning about the privacy of data. It may try to decrypt the cipher text.

Compare with ephemerizer protocol, an integrity with timed ephemerizer has additional property is time server and integrity. The time server needs to define time stamp without any additional interaction with other entities.

A integrity with timed ephemerizer protocol have some polynomial time algorithm. In this algorithm L is security parameter.

1. $Setup_T(L)$- this algorithm generate private or public key pair(PK_T,SK_T). That is run by time server.
2. TimeExt(t,SK_T)-It generate time stamp TS_T i.e. also run by time server. In this paper used notation t<t'.
3. SetupE(L)- It generate set of tuples ($PK_{t\ ephj}$, $SK_{t\ ephj}$,t_{ephj}) and j>=1, where ($PK_{t\ ephj}$, $SK_{t\ ephj}$) is private and public key pair where t_{ephj} is expiration time. Let $t_{ephj}<t_{ephk}$ where j<k. This is run by ephemerizer.
4. Setupu(L)-It generate public and private key pair (PK_u, SK_u) which is run by data consumer.
5. Generate (M,t_{int},PK_u,$PK_{t\ ephj}$ PK_T)- This algorithm provide output cipher text C.message M and t_{int} is initial disclosure time. Let we know (t_{int}, t_{ephj}) and C should be send to data consumer. This algorithm runs by data generator.
6. Retrive(C,TS_{int}, SK_u; SKt_{ephj}) – this algorithm provide output a plain text M otherwise error symbol to data consumer. This algorithm rum between and ephemerizer and data consumer.
The workflow is similar to that of ephemerizer protocol with timed ephemerizer protocol.
1. For encrypt data, the data generate runs the algorithm **Generate.** This algorithm involve public key of time server.
2. For decrypt cipher text, data consumer run algorithm **Retrieve** through ephemerizer and this algorithm needs timestamp from time server.

4. DEFINE SECURITY

Adding integrity into timed ephemerizer is designed to provide guaranty both confidentiality of M (only sender and receiver can retrieve M) and ephemeral property on M (receiver cannot obtain M after expiration time) according to analysis of protocol that we be performed for two session, these properties are satisfied with CL-Atse. However, this protocol does not guaranty of M, it appears immediately: intruder can send own message to receiver. However user wants to maintain integrity. So in this paper add two properties.

1. Integrity of message – during transfer, it is never possible for an intruder to replace, corrupt or change M.

2. Integrity of protocol run- it is never possible for an intruder to replace, corrupt or change any of key protocol that is K_{eph} , T, S.

Property first above can be maintain if user would certainly sign M with sender's private key and he is assume that receiver knows her public key but this work for single session not for multiple once: if user play sender twice than intruder can exchange message of two session therefore user include session id of sender or official recipient name. so we defined two options.

1.Weak integrity- a data like M is corrupted between sender and receiver when receiver a receive a value for M that has been never sent by sender in any of session to ensure this use $\{M\}_{inv(k\ sender)}$.receiver, in place of M that is sender's signature on M add with receiver's name.

2. Strong entity- if M accepted is in other session so it is also corrupted. For ensure this we use additional information with receiver's. We use $\{M\}_{inv(k\ sender)}$. SID, SID is unique, public number that is identifying sender's session playing with receiver.

The security of protocol is calculated by performs some experiment between challenger and attacker. This is called standard practice. Challenger provides answer of oracle queries of attacker. Timed ephemerizer protocol gives guaranty that data will available at either after initial disclosure and before expiration time. In this protocol all public key were verified all participant.

We define some types of adversaries for considering threats against confidentially.

I adversary- This want to access data before it's initially disclosure time.

II adversary- This want to access data before its expiration time.

III adversary- It represent a curious ephemerizer and time server.

First and second adversary used for a timed ephemerizer protocol and third adversary make sense in presence of this two types of adversary.III adversary does not have direct access to private key of data consumer.

Definition 1.- A timed ephemerizer protocol achieve semantic security if only negligible advantage of polynomial time adversary in this semantic security game were advantage is | Pr[b0 =b] – 1/2|.
The attack between adversary A and challenger problem in this way. In this game challenger use functionality of time server.

1. The challenger runs $setup_T$ to generate (PK_u, SK_u) and run $setup_E$ to generate (PK_{tephj}, SK_{tephj})
Where j>=1.
2. Adversary provides a time t gets T time stamp from challenger for adversary can adaptively query TimeExt oracle.
3. The challenger picks a random bit b €{0,1}.
4. The adversary can continue to query.

5. SUCCESSFUL ANALYSIS

All analyzed scenarios, there were no attacks found on ephemerizer protocol with signature on $\{S\}_{k\ receiver}$. Define all paper in this paper such as secrecy of M and $K_{eph,\ T,S}$ and J and integrity and weak integrity and strong integrity property. Signing S give same result and correction of patch n°1 with signing M provide guaranty of integrity of M. this protocol we

analyzed some S scenarios with three roles per honest user and many scenarios as we would include all relevant scenarios where honest play at most two roles. There is some concept show that proposed protocol is secure against I, II, III adversaries.

1. Based on Bilinear Diffie-Hellman (BDH) assumption in random oracle model in proposed scheme achieve semantic security against I adversary.
2. Based on Bilinear Diffie-Hellman (BDH) and KE assumption in random oracle model in proposed scheme achieve semantic security against I adversary given that the public key encryption ε_1 ,ε_2 are one way permutation.
3. Public key encryption scheme ε_1 ,ε_2 are one way encryption given in proposed scheme achieve semantic security against III adversary in random oracle model.

6. ANALYSIS WITH CL-ATSE

An integrity attack on M- when we start analysis of this protocol with CL-Atse, many attacks were found internal data (M, Keph, S, T). However central data is M only and choice to protect attack on M. This extension present for strong integrity and same for weak integrity.

Patch $n^0$1- to prevent previous attack we can include SID. We can use $\{M, SID\}_{inv(Ksender)}$. We also need to protect SID or receiver name with signature. This modification gives guaranty of integrity of M. There may many attacks on integrity of local key keph, S, J, and T.

Patch $n^0$2-. Problem that there may many attacks on integrity of local key keph, S, J,T so we cannot sign everything. But it would not be affordable to add more than one signature. Moreover signing with HMAC can be use. Since it contain data. This data depend on Keph, T, S but some attacks remain. So only way to guaranty the integrity of all local keys is sign to S. signing of S will prevent modification of M and nobody can access M. Only actual receiver can access data.
These proof are given in technical report 15.

7. CONCLUSION

In this paper use the concept of integrity and timed ephemerizer aimed to provide confidentiality and ephemeral property for sensitive data. In this protocol define proposed security model. The analysis give three result, first is that the original ephemerizer protocol provide secrecy of M and provide integrity of M so prevent the modification of M, at least patch $n^0$1 is required and tired it is more secure if we use sign S For this concept of adding integrity with time ephemerizer. There are many interesting research question will open. Future work for this protocol that is it would be interesting to have security proof for another extension for unbounded number of session. The interesting research question is to use integrity and timed ephemerizer as a tool to solve security problem.

8. REFERENCES

[1] M. Bellare and A. Palacio. The knowledge-of-exponent assumptions and 3-round Zero-knowledge protocols. In M. K. Franklin, editor, Advances in Cryptology —CRYPTO 2004, volume 3152 of LNCS, pages 273–289. Springer, 2004.

[2] D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. In J.Kilian, editor,Advances in Cryptology—CRYPTO2001, volume 2139 of LNCS, pages 213–229. Springer, 2001.

[3] J. Cathalo, B. Libert, and J.-J. Quisquater. Efficient and non-interactive timed-release Encryption. In S. Qing, W. Mao, J. Lopez, and G.Wang, editors, Proceedings of the 7th International Conference on Information and Communications Security, volume 3783 of LNCS, pages 291–303. Springer-Verlag, 2005.

[4] Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In J. Feigenbaum, editor, Advances in Cryptology — CRYPTO 1991, volume 576 of LNCS, pages 445–456. Springer, 1991.

[5] A.W. Dent and Q. Tang. Revisiting the security model for timed-release encryption with pre-open capability. In J. A. Garay, A. K. Lenstra, M. Mambo, and R. Peralta, editors, Information Security, 10th International Conference, ISC 2007, volume 4779 of LNCS, pages 158–174. Springer, 2007.

[6] J. A. Halderman, S. D. Schoen, N. Heninger,W. Clarkson,W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten. Lest We Remember: Cold Boot Attacks on Encryption Keys. In P. C. van Oorschot, editor, Proceedings of the 17th USENIX Security Symposium, pages 45–60. USENIX Association, 2008.

[7]. Y. Hwang, D. Yum, and P. Lee. Timed-release encryption with pre-open capability and its application to certified e-mail system. In J. Zhou, J. Lopez, R. Deng, and F. Bao, editors, Proceedings of the 8th International Information Security Conference (ISC 2005), volume 3650 of LNCS, pages 344–358. Springer, 2005.

[8] Charu Arora. The ephemerizer's models and analysis tools. "http://www.loria.fr/_turuani/Ephemerizer models.zip".

[9] Y. Boichut, P.-C. H´eam, and 0. Kouchnarenko. Automatic verification of security protocols using approximations. Research report, INRIA, 2005.

[10] Danny Dolev and Andrew Chi-Chih Yao. On the security of public key protocols. IEEE Transactions on Information Theory, 29(2):198–207, 1983.

[11] Catherine Meadows. Open issues in formal methods for cryptographic protocol analysis. In MMM-ACNS, page 21, 2001.

[12] Radia Perlman. The ephemerizer: Making data disappear. Technical report, Sun Labs, 16 Network Circle, Menlo Park, CA 94025, USA, 2005.

[13] The AVISPA team. The avispa tool for the automated validation of internet security protocols and applications. In CAV, pages 281–285, 2005.

[14] Mathieu Turuani. The cl-atse protocol analyser. In RTA, 2006. to be published.

[15] Q. Tang. Timed-ephemerizer: Make assured data appear and disappear. Technical report, Centre for Telematics and Information Technology, University of Twente, 2009. http://eprints.eemcs.utwente.nl/15802/.