User Revocation using Advanced Key Generation in Cloud Architecture

D. Deepika¹ Dr. Antony Selvadoss Thanamani²

¹M.Phil Research Scholar ²Associate Professor & Head of Depatment

^{1,2}Department of Computer Science & Engineering

^{1,2}NGM College, Pollachi, India

Abstract— The data storage and sharing services in the cloud, users can easily modify and share data as a group. In many cases the users inside the organization itself is not trustful for their concern. So many threads are happening due to users inside the organisation. In order to overcome this problem, we propose a new centralized access control scheme for secure data storage in clouds that supports and warn on anonymous authentication. The cloud verifies the authenticity of the series without knowing the user's identity before storing data. These file systems are used according to the users rights, which can be decided by the admin. Users groups will be created, each group will be provided with a group key and each users will be provided with a personal key. All the uploaded data will be stores in the centralized server. Our scheme also has the added feature of access control in which only valid users are able to decrypt the stored information. The scheme prevents data stored in the cloud from anonymous users. These methods can be implemented using KDC (Key Distribution and certification) methods. Through the introduction of user revocation function group integrity will be maintained according the file systems. The revocation process will modify the key if those users must not have the ability to access data, even if they possess matching set of attributes. For this reason, the owners should change the stored data key and send updated information to other users. Also the revocation process will change the unwanted data while the user is reliving from the group.

Keywords— Centralized cloud server, user revocation, Public key, private key, anonymous authentication, Key Distribution and certification

I. INTRODUCTION

In cloud computing Group communication is becoming increasingly popular in Internet applications such as work group, work allotment, work processing and etc. For secure communications, the integrity of messages, member authentication, and confidentiality must be provided among group members [1]. To maintain message integrity, all group members use the Group Key for encrypting and decrypting messages while providing enough security to protect against passive attacks. Tree-based Group is an efficient group key agreement protocol to generate the GK. Tree group assumes all members have an equal computing power. [2] One of the characteristics of distributed computing and grid environments is heterogeneity; the member can be at a workstation, a laptop or even a mobile computer. Member reordering in the tree group protocol could potentially lead to an improved protocol; such reordering should capture the heterogeneity of the network as well as latency [3]. This research investigates dynamic reordering mechanisms to consider not only the overhead involved but also the scalability of the proposed protocol.Group communications are created all over the network in the form of videoconferences, on-line chatting programs, games, and gambling. Security plays an important role in these instances of group communication [4]. According to member authentication processes and key distribution take place at the beginning of a group communication. The group size tends to be less than 100. However, the Group Key computation takes a relatively long time to complete. For achieving a high level of security, the GK should be changed after every member joins and leaves so that a former group member has no access to current communications and a new member has no access to previous communications [5]. The group key agreement protocol focuses on the GK computation, which evaluating a function of modular consists of exponentiations. In order to calculate the GK using modular exponentiations, the adaptation of key trees is needed to reduce the computational overhead. Modular exponentiation is the computationally most expensive operation in tree group [6]. The number of exponentiations for membership events depends on the number of group members. The algorithm efficiency of tree group is O (log2 n), where n is the current number of members, so it is efficient as long as the key tree is perfectly balanced. However, maintaining a perfect key tree balance results in a significant overhead [7]. Maintaining a perfectly balanced tree after a membership change is one problem; another is that tree group assumes an underlying homogeneous network.

II. PROBLEM DEFINITION

We know that nowadays security is less in group oriented based applications. To secure the confidential data we are going to use user revocation and key transfer protocols. User revocation is process of changing or modifying. Key transfer protocol which is fully trusted on Key distribution and certification, where it will generate the key and pass it to all the group members in a safe and secure way. Admin will create a user, whenever user is created a 16 bit Alpha numeric key will generated along with user name and password and the details about these things will send to user's mail. So Admin will act as a KDC (Key Distribution and certification) once that user is created. Admin only had a rights to allot the group for that user. Admin only had a rights to view the group details and to edit or update the user details. A member in one group can send a file to a member in a same group or to another group. A file can be sent from one Group to another group so all the group member belongs to both the group can view the file. A safe group Communication will be done. All the users in every group will be provided by a 16 bit key, by using this key the user can view their received files. To send data from one group to another group they have a group key, by using that key they can send the data to all the group members in a safe and secure manner.Users have rights to change their password. Users can view their files and download it. Whenever group member leave the group, the group key is regenerated dynamically so that member cannot re-join the group. Whenever a new memberenters in to the group he/she can't able to retrieve the previous messages. Whenever a new member leaves the group he/she can't able to retrieve the messages by using key, because the key will regenerate dynamically.

III. EXISTING METHOD – PUBLIC KEY CRYPTOGRAPHY

The idea of Public Key Cryptography is to send messages in such a way that only the person who receives them can understand them even if the method of encryption is discovered by 'an enemy' who intercepts the messages. The person who sends the message encodes it; the person who receives the message decodes it (puts it back into a readable form). Public Key Cryptography was discovered (or invented) by R. Rivest, A. Shamir and L.Adleman about 2001. This method has been widely used to ensure security and secrecy in electronic communication and particularly where huge transactions are involved. The method depends on the fact that while it is easy to calculate the product of two large prime numbers (particularly with the help of a computer) it is, for all practical purposes, impossible to find the factors of a large number if it has only very large prime factors. This is because all methods of finding such factors would take many thousands of years by even the fastest modern computers.

IV. IMPLEMENTATION

A. Revocation and Key generation

Advanced Encryption Revocation Standard (AERS) is based on a design principle known as a substitution-permutation network. It is fast in both software and hardware. Unlike its predecessor, DES, AES does not use a Feistel network. AERS has a fixed block size of 128 bit and a key size of 128, 192, or 256 bit, whereas Rijndael has specified with block and key sizes in multiples of 32 bit, with a minimum of 128 bit. The block size has a maximum of 256 bit but the key size has no theoretical maximum AERS operates on a 4×4 column-major order matrix of bytes, termed the state (versions of Rijndael with a larger block size have additional columns in the state). Most AERS calculations are down in a special finite field.

- Key length is variable: the key length can be varied from 16 up to any larger value depending on the security level required.
- Word length is variable: the block size can be varied between 1 to 16 bit or 1 to 32 and so on. That is, encryption can be performed on 16 or 32 or 64 bit blocks. This, in turn, can be used on different processor architectures employing 16, 32, or 64 bit registers.
- The algorithm, therefore, provides variable degrees of security. However, this increased security level will be at the cost of increased size of the cipher-text.
- The number of rounds is variable: the whole process can be repeated r times using the same key.

B. Algorithm

- The adversary first outputs the challenge identity and time, and also some information state it wants to preserve. Later it is given access to three oracles that correspond to the algorithms of the scheme. The oracles share state.2 since we use the simplified notation for the oracles, we define them now:
- (S, SK, KU, DK, E, D, R) be a Revocable scheme
- The private key generation oracle $SK(\cdot)$ takes input identity ω and runs $SK(pk, mk, \omega, st)$ to return private key sk ω .
- The key update generation oracle $KU(\cdot)$ takes input time t and runs KU(pk, mk, t, rl, st) to return key update kut .
- The revocation R (\cdot, \cdot) takes input identity ω and time t and runs R (ω, t, rl, st) to update rl. For adversary A and number of users n define the following experiments:

C. Experiment

Expsrid-cpaRIBE,A,n(1 κ) b \$ \leftarrow {0, 1} (ω *, t*, state) \$ \leftarrow A(1 κ) (pk, mk, rl, st) \$ \leftarrow S(1 κ , n) (m0, m1, state) \$ \leftarrow ASK(·),KU(·),R(·,·) (pk, state) c* \$ \leftarrow E(pk, ω *, t*, mb) d \$ \leftarrow ASK(·), KU(·),R(·,·) (pk, c*, state) If b = d return 1 Else return 0.

RESULT AND DISCUSSION

Key Length

Now we will show the number of possible keys, i.e., the key space when the key length is 16. The probability of replacing a string of bits whose length ranges from 1 to 8 bit in an octet is 1/64. Consequently, if the key length is 16 there are 64(16) possible keys. So we can say that if the attacker has a cipher text and he knows that the key length is 16, there are $7.9 \times 10(28)$ attempts to find the correct key, i.e., there are $7.9 \times 10(28)$ attempts to find the correct plaintext or secret message. This eliminates brute force attack; however other types of attacks will be discussed in future work.

Encryption type = 64 bit key

Key Length = 23 char

Key type = Alpha numerical with special characters.

Key character = Encryption and decryption

Key Limitation = Caps alphabets = 26, Small alphabets = 26, 0 -9 numbers = 10, Special Characters = 10: result = 3.848329407410064e+135 of key combinations can be produced. It is equivalent to 30000 trillion and above combination.

		GE	NERATE	GROUP KI	EY							
		Select Group			GROUP A	•						
		Generate Key		ley	Cancel							
			GROUP	NAME		G	ROUP	KEY				
		GROUP A			RwTM97ArId0wtjN1RtYvLg==							
	GROUP B GROUP C			bUQs+7rnaep2OTpnDCN+HA== 1D8Aj9z8LDQeLW160UXXTQ==								
		GR	OUP D		rutGwlkxxIEtJ	W101	/SIC01	v==				
		1			Fig. 1: Genera	te gro	up key					
	Nai	ne Ag	ge Gender	Address	s Phone nu	m		Email		Group		
	banu	29	Male	ram nagar	9944228168	ba	nu@gm	ail.com		GROUP A	Select	
	siva	23	Male	R.S.Puram	8768886832	43 siv	a@gma	il.com		GROUP B	Select	S.,
	giri	22	Male	sdfsdfsd	23435	sd	fsf			GROUP C	Select	1
	mano) 44 22	Male	fghgfhg	65546456	fg	dg	.		GROUP D	Select	
	abina	iya 22	Female	coimbatore	9988770507	ab	ishree0	5@gmail.c	om	GROUPD	Select	
		Allot		GROUP B GROUP C GROUP D								
	I				Fig. 2: User de	etail se	lection					
	72.0		GROUI	'A					GRO	UP B	_	
u	banu	Passwar 1	d AZnx5IWYe	Ker 2GDyCHfmqC3W	Www= Edit Delete	siva chithra	erinam siva chithra	Password 1 1 1 8	8N2qhGo 3MoADU	Ker jeHEqxk4swMo oN8/o5kqzhX5v	uZHg== 1	dit Delete dit Delete
			GROUT	PC .		_			GRO	UPD		
-	Username	Passwa	đ	Key		Name	Usernar	ne Password		Key:		
	giri	1	i7mYyod5W	hD3yumCC4UVB	g== Edil Delete	mano abinay	mano a abinaya	1 12345	BEi1sJzF L38WkO	UINKz]9JD7go 00LdvNNhQt0	Q== DUEEow==	Edit Delete
	195				Fig. 3: Group	distril	oution				6	7
nam pna	e kaila me GRC	sh DUP A										
in al	Date 10/15	/2017 Unloaded	0	alayer tooke			-Edit Co	ontents				
er	Date	Title		Uploaded Co	ontent		Upload	ded Title	About	C#		
in	2/21/2016	Visual Basic Content Content 15/2017 About C# C# is a simple, modern, general- programming language develope teach you basic C# programming through various advanced concep programming language		et is a type of .net .modern, general-m	numnee object oriented		Uploaded Contents Uploaded Contents Update Contents Ca		C# pro take) advanc C# pro	C# programming and will also take you through various advanced concepts related to C# programming language, it is object oriented language Cancel		
				he Microsoft within ite		is ob						

Fig. 4: Upload files

Select Group	GROUP A 🔻	View Modifie	ed Data			
Original User	Modified User	Uploaded Date	Uploaded Title	Uploaded Content		
Admin		2/21/2016	Visual Basic .Net Content	visual basic .net is a type of .net		
Admin		10/15/2017	About C#	C# is a simple, modern, general-purpose, object-oriented programming language developed by Microsoft within its .NET initiative led by Anders Hejlsberg. This tutorial will teach you basic C# programming and will also take you through various advanced concepts related to C# programming language.		

Fig. 5: View modified data

V. CONCLUSION

In this article, we proposed efficient user revocation in the cloud. When a user in the group is revoked, we allow the semi-trusted cloud to re-sign blocks that were signed by the revoked user with proxy re-signatures. Various testing has been done to verify the accuracy of the key generation and revocation process. Experimental results show that the cloud can improve the efficiency of user revocation, and existing users in the group can save a significant amount of computation and communication resources during user revocation. This research will play a major impact in group work based applications. All the result has been verified successfully

VI. FUTURE WORK

There are some alternative approaches to Tor for providing non link ability and non-traceability of network communications, such as JAP, Ultrasurf, and FreeGate. Among these, Tor is a more mature implementation and one of the most popular. More importantly, Tor can be easily integrated with any TCP based protocol that can be SOCKSified. This was one of our requirements and why we evaluated the performance over Tor. It remains to be seen whether the other anonym zing protocols can be so adapted and, if so, how they may perform well in user revocation modeling.

REFERENCE

- M. Rabin, "Efficient dispersal of information for security," Journal of the ACM (JACM), vol. 36(2), pp. 335–348, Apr. 1989.
- [2] J. G. et al. (2006) The expanding digital universe: A forecast of worldwide information growth through 2010. IDC. [Online]. Available: Whitepaper
- [3] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proc. of ACM CCS, Virginia, USA, Oct. 2007, pp. 598–609.
- [4] A. Juels and B. S. Kaliski, "Pors: Proofs of retrievability for large files," in Proc. of ACM CCS, Virginia, USA, Oct. 2007, pp. 584–597.
- [5] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: theory and implementation," in Proc. of CCSW 2009, Ilinois, USA, Nov. 2009, pp. 43–54.
- [6] Y. Dodis, S. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplification," in Proc. of TCC 2009, CA, USA, Mar. 2009, pp. 109–127.

- [7] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Proofs of retrievability via hardness amplification," in Proc. of ESORICS 2009, Saint-Malo, France, Sep. 2009, pp. 355–370.
- [8] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proc. of ACM CCS, Illinois, USA, Nov. 2009, pp. 213– 222.
- [9] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacypreserving public auditing for data storage security in cloud computing," in Proc. of IEEE INFOCOM 2010, CA, USA, Mar. 2010, pp. 525–533.
- [10] J. Yuan and S. Yu, "Proofs of retrievability with public verifiability and constant communication cost in cloud," in Proc. of International Workshop on Security in Cloud Computing, Hangzhou, China, May 2013, pp. 19–26.
- [11] E. Shi, E. Stefanov, and C. Papamanthou, "Practical dynamic proofs of retrievability," in Proc. of ACM CCS 2013, Berlin, Germany, Nov. 2013, pp. 325–336.
- [12] Cloud9. (2011) Your development environment, in the cloud. Cloud9. [Online]. Available: https://c9.io/
- [13] Codeanywhere. (2011) Online code editor. Codeanywhere. [Online]. Available: https://codeanywhere.net/
- [14]B. Wang, B. Li, and H. Li, "Oruta: Privacy-preserving public auditing for shared data in the cloud," in Proc. of IEEE CLOUD 2012, Hawaii, USA, Jun. 2012, pp. 295– 302.
- [15] B. Wang, L. Baochun, and L. Hui, "Public auditing for shared data with efficient user revocation in the cloud," in Proc. of IEEE INFOCOM 2013, Turin, Italy, Apr. 2013, pp. 2904–2912.
- [16] J. Yuan and S. Yu, "Efficient public integrity checking for cloud data sharing with multi-user modification," in Proc. of IEEE INFOCOM 2014, Toronto, Canada, Apr. 2014, pp. 2121–2129.
- [17] D. Catalano and D. Fiore, "Vector commitments and their applications," in Public-Key Cryptography - PKC 2013, Nara, Japan, Mar. 2013, pp. 55–72.
- [18] Q. Wu, Y. Mu, W. Susilo, B. Qin, and J. Domingo-Ferrer, "Asymmetric group key agreement," in Proc. of EUROCRYPT 2009, Cologne, Germany, Apr. 2009, pp. 153–170.
- [19] D. Boneh and H. Shacham, "Group signatures with verifierlocal revocation," in Proc. of ACM CCS, DC, USA, Oct. 2004, pp. 168–177.
- [20] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in Proc. of Asiacrypt 2001, Gold Coast, Australia, Dec. 2001, pp. 514–532.

- [21] D. Boneh and X. Boyen, "Collision-free accumulators and failstop signature schemes without trees," in Proc. of EUROCRYPT 2004, Interlaken, Switzerland, May 2004, pp. 56–73.
- [22] N. Baric and B. Pfitzman, "Collision-free accumulators and fail-stop signature schemes without trees," in Proc. of EUROCRYPT 1997, Konstanz, Germany, May 1997, pp. 480–494.
- [23] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in Proc. of CRYPTO 2004, CA, USA, Aug. 2004, pp. 41–55.