2011 International Conference on Information Management and Engineering (ICIME 2011)
IPCSIT vol. 52 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V52.2

Secure Storage and Retrieval of Data without Original Files in Cloud Architectures Subtitle as needed

Jianhong Zhang and Xue Liu [+]

College of Sciences, North China University of Technology, Beijing, China

Abstract. When a client uses the service which is offered by cloud storage providers, he might worry about how to retrieve their original files securely. Relieving client's storage burden and successfully retrieving data on demand are the ultimate purposes of cloud storage service. To achieve the above two aims, in this paper we propose a specific way to store and retrieve the out stored data in cloud storage. We utilize an efficient publicly verifiable secret sharing (PVSS) scheme to divide the file into multiple parts and store them among a group of servers, then using the recovery algorithm in PVSS, the users can retrieve their stored files successfully from any k qualified servers without the need of holding original file. Thus, the storage burden of the user is relieved much. And we show that the security of the protocol is closely related to the computational Diffie-Hellman problem.

Keywords: component; formatting; style; styling; insert

1. Introduction

In recent years, distributed data storage—cloud storage, has gained increasing popularity for its efficient and robust data management in cloud computing. Cloud storage brings the data owner many appealing benefits: relief of the burden of storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, personnel maintenance, and so on [1]. Although cloud storage provides a new method for information processing and makes these advantages more appealing than ever, there are also many security challenges needed to be solved properly since many service providers are untrusted, and they may make malicious behaviors. For example, the providers may refuse to release the data that client has stored in cloud. This makes that the retrievability of user's data is at the mercy of their service providers. So, enabling secure retrieval of outsourced data becomes the most important problem we must consider.

The researchers have proposed two basic approaches which are called Proofs of Retrievability (POR) [18] and Provable Data Possession (PDP) [19] to check the availability of the stored data in cloud storage. Jeuls et. al. [18] proposed a model of "proof of retrivability" (POR), a POR is a protocol in which a server/archive proves to a client that a target file F is intact, in the sense that the client can retrieve all of F from the server with high probability. Anteniese et al. defined the "provable data possession" (PDP) model in [19] for verifying the possession of file on trustless storage severs. In PDP model, the client preprocesses the data and then sends it to an untrusted server for storage, while keeping a small amount of meta-data. The client later asks the server to prove that the stored data has not been tampered with or deleted (without downloading the actual data) [21]. Based on the two models, many variants [20-22] were proposed under different cryptographic assumptions. These schemes provide probabilistic guarantees of possession, where the client checks a random subset of stored blocks with each challenge. However, theses variant schemes introduced above mainly focus on adopting a series of interactive verification to provide a proof for clients that the data

[+] Corresponding author. Tel.: + (13717565719); fax: + (01088803275)
E-mail address: (jhzhangs@163.com).

they have stored in cloud can be obtained on demand, but they haven't provided a specific method to retrieve out stored data.

To solve the problem, in this paper we put forward a new approach to achieve efficient and secure retrieval of data which the user stored in cloud storage providers. Based on the PVSS (publicly verifiable secret sharing) scheme proposed in [3,4], we divide the file into multiple parts and store them among a group of servers, then we can retrieve the original data from a qualified set of servers correctly. In the process, we use the properties of PVSS: firstly, due to the publicity of PVSS, we needn't private channel between the client and the servers; Secondly, public verifiability can also ensure that the correctness of any server's contribution can be verified by its cooperators, so it prevents the cheat of the cooperative servers in retrieve phase. The prominent advantage in our method is that the client needn't keep the backup copy of their original data in his/her PC, and then he/she can efficiently retrieve the original data from the storage provider. This will make the storage burden of the user relieved much. So, our scheme achieves the two dominating aims of cloud storage service.

Organization: The rest of the paper is organized as follows: in section 2 we recall the related preliminaries of secret sharing mechanism and some number theories. Then our retrieval scheme is proposed in section 3, security analysis and discussion of the scheme is given in section 4. Finally, we conclude in section 5.

2. Preliminaries

2.1 PVSS(Publicly Verifiable Secret Sharing)

A PVSS scheme consists of three parties: a dealer D, participants , and a verifier V (V is not necessarily one of the participants).It includes following four algorithms.

1) *Share*: D runs algorithm *Share* and divides secret s into n shares:

$$Share(s) = (s_1, s_2, \cdots s_n)$$

2) *Encpt:* *D* sends $(E_1(s_1), E_2(s_2), \cdots, E_n(s_n))$(the encryption of s_i) to the verifier V.

3) *PubVerify*: V checks the correctness of $(E_1(s_1), E_2(s_2), \cdots, E_n(s_n))$and outputs 1 if it is correct.

$$PubVerify(E_1(s_1), E_2(s_2), \cdots, E(s_n)) = 1$$

4) *Recover*: Any set of more than k participants of P_i can recover s by decrypting the cipher:

$$Recover\left(\left\{D_{a_i}(s_{a_i}) \mid P_{a_i}, a_i \in [1,n], 1 \le i \le k\right\}\right) = s$$

2.2 Computational Diffie-Hellman Assumption

The security of our scheme is based on the following CDH assumption. The CDH problem in a cyclic group G_p with order p is described as follows:

Definition 1: Given a triple $(g, g^a, g^b) \in {G_p}^3$, its goal is to compute the $g^{ab} \in G_p$ .We say that algorithm A has the advantage ε in solving the CDH in group G_p if

$$\Pr\left[A(g, g^a, g^b) = g^{ab}\right] \ge \varepsilon$$

where the probability is over the random choice of generator $g \in G_p$, the random choice of $a, b \in Z_p$.

Assumption 1: The (t, ε)−CDH assumption holds in group G_p if no t−time adversary has advantage at least ε in solving CDH in G_p.

2.3 Signatures of Knowledge Proof

So-called zero-knowledge proofs of knowledge allow a prover to demonstrate the knowledge of a secret w.r.t. some public information such that no other information is revealed in the process. The protocols we use in the following are all 3-move protocols and can be proven zero-knowledge in an honest-verifier model. Such protocols can be performed non-interactively with the help of an ideal hash function H. We refer to the resulting constructs as signatures of knowledge.

In this section, we consider the building blocks: signature of knowledge of equality of two discrete logarithms of, say y_1, y_2 w.r.t. base g and h, i.e., knowledge of an integer α satisfying $y_1 = g^\alpha$ and $y_2 = h^\alpha$.

Definition 3: Let G_p be a finite cyclic group of order p and α be a generator of G_p , random $\beta \in G_p$. The discrete logarithm of β to the base α, denote $\log_\alpha \beta$, is the unique integer x, $0 \le x \le p-1$, such that $\beta = \alpha^x$.

Definition 4: Let $y_1, y_2, g, h \in G_p$, verifying $c = H\left(y_1 \| y_2 \| g \| h \| g^s y_1^c \| h^s y_2^c \| m\right)$is a signature of knowledge of the discrete logarithm of both $y_1 = g^\alpha$ w.r.t. base g and $y_2 = h^\alpha$ w.r.t base h on a message m .The party in possession of the secret α is able to compute the signature, provided that $\alpha = \log_g y_1 = \log_h y_2$, by selecting a random t and then computing c and s as:

$$c = H\left(y_1 \| y_2 \| g \| h \| g^t \| h^t \| m\right) \text{ and } s = t - c\alpha$$

3. Our Storage and Retrieve Scheme

In our protocol, we can share the important file among a group of servers and everyone besides the servers can verify the correctness of the shares. Thus the shared data are secure and could be retrieved even if some shares stored on the servers were attacked. We utilize the protocol based on a novel PVSS proposed in [3] which adopts (k, n) secret sharing and the signatures of knowledge [16] technique to complete the retrieval and publicly verifiable functions. By this mechanism, we are able to achieve the secure storage and retrieval of files in the cloud. Just as the PVSS, it includes following three phases:

1). Initialization We selected two generators g, h in group G_p and published them. We denote the user in the cloud storage by U and severs by S_i acted respectively as the D and P_i in the PVSS. Each severs choose a private key $x_i \in Z_p$ and generate a public key $y_i = h^{x_i}$.

2). Distribution ***Distribution of shares:*** The user U who wants to share a random file in the form of h^s among a set of severs $S_1, S_2, \cdots, S_n$ will choose a random polynomial $f(x)$ of degree k-1 with coefficients in Z_p .

$$f(x) = \sum_{j=0}^{k-1} a_j x^j$$

Let $a_0 = f_0 = s$, then U keeps the polynomial secret but opens $C_j = g^{a_j} \left(0 \le j \le k-1\right)$.U encrypts the shares of file h^s using the public key of S_i to be $Y_i = y_i^{f_i}, \left(f_i = f(i), 0 \le i \le n\right)$ and produces $X_i = g^{f_i}$. Then, publishing the master public parameters in this system $\left(p, g, h, y_i, X_i, Y_i, C_j\right)$.

Verification of the encrypted share: The verifier (not necessary to be server S_i) can compute $X_i = \prod_{j=0}^{k-1} C_j^{i^j}$ from C_j .

Then the prover (who will be the user) and the verifier execute the following steps:

Prover		Verifier
$t_i \in Z_p$		
$\phi_i = g^{t_i}, \varphi_i = y_i^{t_i}$		
	$\xrightarrow{\phi_i, \varphi_i}$	
	$\xleftarrow{c_i}$	$c_i \in Z_P$
$s_i = t_i - c_i\alpha$	$\xrightarrow{s_i}$	$\phi_i \stackrel{?}{=} g^{s_i} X_i^{c_i}$
		$\varphi_i \stackrel{?}{=} y_i^{s_i} Y_i^{c_i}$

Then prover computes

$$c = H\left(X_i \| Y_i \| g \| y_i \| g^{s_i} X_i^{c_i} \| y_i^{s_i} Y_i^{c_i}\right).$$

The process executes *k* times and the proof consists of $R = \left(s_1, \cdots, s_n\right)$and c .So, the verifier calculates $c' = H\left(X_i \| Y_i \| g \| y_i \| \phi_i \| \varphi_i\right)$ and then checks weather $c' = c$.

3). Retrieve of shares ***Decryption of shares.*** Each sever employ its private key x_i to find the shares σ_i of the random value s by computing $\sigma_i = Y_i^{1/x_i} = h^{f_i}$.

Extract the shares. Without loss of generality, suppose that a set of severs S_i decrypt correct shares $\sigma_i \ (i = 1,2,\cdots,k)$.Then, they can reconstruct the function $f(x)$ by Lagrange base function

$$l_i(x) = \prod_{\substack{j \neq i \\ i=0}}^{k} \frac{x-j}{i-j}$$

and compute f_0 to recover the shared file h^s :

$$\prod_{i=1}^{k} \sigma_i^{\ l_i} = \prod_{i=1}^{k} \left(h^{f_i}\right)^{l_i} = h^{\sum_{i=1}^{k} f_i l_i} = h^{f_o} = h^s$$

where $l_i = \prod_{\substack{j \neq i \\ i=0}}^{k} \frac{j}{j-i}$ is the Lagrange coefficient.

4. Security Analysis and Discussion

4.1 The Security Analysis

There are two points we must consider when discussing the security of the scheme as [3]. That is ⅰ) the retrieval protocol results in the file distributed by the user U for any qualified set of severs, ⅱ) any non-qualified set of severs is not able to recover the file. Also, we need consider the security of the encrypted shares. So, we have the following propositions and the corresponding proofs.

Proposition1. Under the Diffie-Hellman assumption, it is infeasible to break the encryption of the shares.

Proofs: As discussed above, breaking the encryption of the random shares is to find h^{f_i} from given g, h, X_i, Y_i, y_i .Because g, h are both the generator in the group G_p , so we can set $h = g^{\alpha}, X_i = g^{\beta}, y_i = g^{\gamma}$, which implies that $f_0 = \beta$ .Suppose we have obtained h^{f_i} , then from the constructed parameters above we have following computational process to gain $g^{\alpha\beta}$:

$$h^{f_i} = \left(g^{\alpha}\right)^{f_i} = \left(g^{f_i}\right)^{\alpha} = X_i^{\ \alpha} = g^{\alpha\beta}$$

$$Y_i = y_i^{\ f_i} = \left(g^{\gamma}\right)^{f_i} = g^{\beta\gamma}$$

That is to say we can deserve $g^{\alpha\beta}$ and $g^{\beta\gamma}$ from given $g^{\alpha}, g^{\beta}, g^{\gamma}$ .Obviously, this contradicts the Diffie-Hellman assumption. So, we get the result in proposition 1 under the Diffie-Hellman assumption: the encryption of the shares can't be break without the secret key.

Proposition2. *Under the Diffie-Hellman assumption, the retrieval scheme is correct. That is to say, ⅰ) for any a qualified set of servers, the retrieval protocol can recover the original file which l is stored in the server by the user, ⅱ) any a non-qualified set of servers is not able to recover the original file.*

Proof: ⅰ) this verdict comes from the previous verify process and the facts that X_i is derived from the $C_j = g^{a_j}$ as $X_i = \prod_{j=0}^{k-1} C_j^{i^j} = g^{f_i}$. The signatures of Knowledge [10] we use in previous verification setup has been proven to be zero-knowledge in an honest-verifier model. Thus the knowledge extractor is able to recover the file h^s we shared among severs once it decrypt correct shares σ_i .

ⅱ) This part implies that any set of less than k servers can't recover the shares of the value we distributed. Now we suppose w.l.o.g that severs $S_1, S_2 \cdots, S_{k-1}$ can break the scheme to recover the value. Then we will set up the system to fulfill the recover but this ultimately violates Diffie-Hellman assumption.

We set $h = g^{\alpha}, C_0 = g^{\beta}$, which implies that $f_0 = \beta$ because of $C_0 = g^{f_0}$ defined in part 3. We choose $f_1, f_2, \cdots, f_{k-1}$ randomly from Z_p which will fix polynomial $f(x)$ .So we can compute $X_i = g^{f_i}, Y_i = y_i^{\ f_i} (i = 1, \cdots, k-1)$

Because f_0 is implicit, we are not able to compute $f_k, \cdots, f_n$ .However, we can calculate $X_i = g^{f_i}$ by Lagrange interpolation, then calculate the remaining C_j .For $i = k, \cdots, n$, we compute the public keys y_i of severs S_i as $y_i = g^{\varepsilon_i}$ (random $\varepsilon_i \in Z_p$) and set $Y_i = X_i^{\ \varepsilon_i}$ such that $Y_i = y_i^{\ f_i}$ as required. So far, we defined

the system completely from the right distribution. Now, suppose that severs $S_1, S_2 \cdots, S_{k-1}$ can recover the value h^{f_0} .Then we can compute $g^{\alpha\beta}$ from $h = g^{\alpha}$ and the fact $f_o = \beta$:

$$h^{f_0} = \left(g^{\alpha}\right)^{f_0} = \left(g^{\alpha}\right)^{\beta} = g^{\alpha\beta}$$

That is to say, we can compute $g^{\alpha\beta}$ from g^{α} and g^{β} . Obviously, it is in contradiction to the Diffie-Hellman assumption. Thus, we can conclude that any non-qualified set of servers is not able to recover the stored file.

4.2 Performance Evaluation

In this scheme, the dealer (client) only needs to post k+n elements of G_p (the number of C_j and Y_i) plus n+1 number (n responses and the challenge c) of size $|p|$. And also, all of these exponentiations are with relatively small exponents from Z_p . Compared to previous Stadler's $O(k^2 n)$ secret sharing scheme, where k is a security parameter, n denotes the number of participants, the scheme we utilize only need $O(kn)$ complexity. Other PVSS such as [6] achieve the same asymptotic complexity as our scheme. But that scheme [6] uses a rather complicated proof to show that a share is correctly encrypted. For example, its simplest form is to use RSA with public exponent 3, the scheme requires at least 17 secure commitments per participant, where each commitment requires a two-way or three-way exponentiation. Hence we can conclude that the scheme we utilize is essentially optimal and this will be very suitable for the storage and retrieval of scale data in cloud storage.

5. Conclusion

In this paper, we propose a secure and efficient scheme by making use of PVSS to ensure the storage and retrieval of user's file in cloud storage. Our scheme has realized two important purposes: 1) by our method, the user can efficiently retrieve the original out stored data from the storage provider.2) the most important advantage is that the user needn't keep the backup copy of their original data in his PC. Thus, it greatly relieves the user's storage burden. Furthermore, our scheme has a useful property: due to the public verifiability of PVSS, it can ensure that the correctness of any server's contribution can be verified by its cooperators, so this prevents the cheat of the cooperative servers in retrieve phase; The security of this scheme is based on the computational Diffie-Hellman assumption and we give the relevant proof.

6. Acknowledgment

This work is supported by the National Natural Science Foundation of China (No: 60703044), and the Nova Programma (No:2007B-001) and Beijing Natural Science Foundation Programm and Scientific Research Key Program of Beijing Municipal Commission of Education (NO:KZ2008 10009005).

7. References

[1] M. Armbrust et al., "Above the Clouds: A Berkeley View of Cloud Computing,"Univ. California, Berkeley, Tech. Rep. UCBEECS-2009-28, Feb. 2009.

[2] Tharam Dillon,Chen Wu and Elizabeth Chang "Cloud Computing: Issues and Challenges" Advanced Information Networking and Applications (AINA), AINA.2010.187. Page(s): 27 – 33

[3] Berry Schoenmakers. A simple publicly verifiable secret sharing scheme and its application to electronic voting.CRYPTO'99, LNCS 1666,pp.148-164, Springer-Verlag, 1999.

[4] M. Stadler. Public verifiable secret sharing, Advances in Cryptology- EUROCRYPT'96, LNCS 070, U. Maurer ed., pp.190-199, Springer-Verlag, 1996.

[5] Wei Ren, Yi Ren and Hui Zhang, Secure, dependable and publicly verifiable distributed data storage in unattended wireless sensor networks. SCIENCE CHINA Information Sciences Vol. 53, Number 5, pp: 964-979, Springer-Verlag, 2010.

[6] Eiichiro FUJISAKI and Tatsuaki OKAMOTO,A Practical and Provably Secure Scheme for Publicly Verifiable Secret Sharing and Its Applications. In K. Nyberg, editor, Advances in Cryptology – EUROCRYPT'98, volume 1403 of LNCS, pages 32–46.Springer-Verlag, 1998.

[7] A. Fiat and A. Shamir. How to prove yourself: practical solutions to identification and signature problems. | CRYPTO'86, vol. 263 of LNCS, pp. 186-194, Springer-Verlag, 1987.

[8] D.Chaum and T.P.Pedersen.Wallet databases with observers. In Advances in Cryptology—CRYPTO ’92, vol740 of LNCS, pages 89–105, Berlin, 1993. Springer-Verlag.

[9] M. Ballare and P. Rogaway, “Random oracle are practical: a paradiam for designing efficient protocols” ACM Conference on Computer and Communications Security: pp62-73

[10] D. Pointcheval and J. Stern. ”Security Proofs for Signature Schemes”, Advanced in Cryptology - Eurocrypt 1996, LNCS 1070, pp387-398, Springer-Verlag, 1996.

[11] Giuseppe Ateniese,et. ,A Practical and Provably Secure Coalition-Resistant Group Signature Scheme,CRYPTO 2000,vol.1880 of LNCS,pp:255-270, Springer-Verlag, 2000

[12] R.Cramer, I.Damgard, and B.Schoenmakers, “Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocol”, In Crypto’94 LNCS 839, pp 174-187, Springer-Verlag, 1994

[13] E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. CRYPTO'97, vol.1297 of LNCS, pp. 16-30, Springer-Verlag, 1997.

[14] J Camenisch, Markus Michels, Proving in zero-knowledge that a number is the product of two safe primes. In Advances in Cryptology | EUROCRYPT'99, vol1592 of LNCS, pp. 107-122, Springer-Verlag, 1999.

[15] E. F. Brickell. “Some ideal secret sharing schemes”. Journal of Combinatorial Mathematics and Combinatorial Computing, 9:105–113, 1989.

[16] C.-P. Schnorr. “Efficient identification and signatures for smart cards”. CRYPTO ’89, vol435 of LNCS, pages 239–252. Springer-Verlag, 1989

[17] C.P. Schnorr. “Efficient signature generation by smart cards”. Journal of Cryptology, 4(3):161-174，1991

[18] A. Juels and J. Burton S. Kaliski, “PORs: Proofs of Retrievability for Large Files,” Proc. of CCS ’07, pp. 584–597, 2007.

[19] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable Data Possession at Untrusted Stores,” Proc. of CCS ’07, pp. 598–609, 2007.

[20] D. L. G. Filho and P. S. L. M. Barreto, “Demonstrating Data Possession and Uncheatable Data Transfer,” Cryptology ePrint Archive, Report 2006/150, 2006, http://eprint.iacr.org/.

[21] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, “Cooperative provable data possession.” Cryptology ePrint Archive, Report 2010/234, 2010. http://eprint.iacr.org/

[22] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou. “Enabling public verifiability and data dynamics for storage security in cloud computing”. In ESORICS, pages 355–370, 2009.

[23] T. S. J. Schwarz and E. L. Miller, “Store, Forget, and Check: UsingAlgebraic Signatures to Check Remotely Administered Storage,” Proc. of ICDCS ’06, pp. 12–12, 2006.