A Solution to Defend Cloud Auditing Protocols from A Potential Attack

Jiaxian Lv¹, Yi Wang² and Jinze Liu³
¹,²,³College of Computer, National Defence of Technology, Changsha, Hunan, 541004, China
*Corresponding author’s e-mail: ljx_nudt@163.com

Abstract. PRISM, a monitoring plan carried out by American government and revealed to the public in 2013, brought subversion attack back to the centre of cryptography research. In this work, we propose a kind of subversion attack on the cloud auditing protocols, where the cloud server plays the role of subversion attacker, manages to recover the secret message stored by the users of the auditing protocol. Then we propose a general frame of defence solutions with experiments to evaluate the practicability of our theme.

1. Introduction
Cloud Auditing Protocols. Cloud Auditing Protocol is an agreement for users to verify the integrity of data stored in the Cloud by storing key information locally. Through this protocol, users can monitor the behavior of cloud service providers while deleting copies of local data, and ensure the availability and integrity of users’ own data is not violated.

With the development of cloud data and cloud platform technology in the past two decades, the research on cloud auditing protocols has developed greatly. The main research results of cloud auditing will be introduced below. In [1], Deswarte et al. pointed out that there are two difficulties in data integrity verification in the cloud: 1) The cloud server is vulnerable to hacker attacks, so it is very unsafe to run a simple data verification protocol on the cloud. 2) it is very inefficient for users to confirm the data integrity of the cloud by downloading data from the cloud to the local, comparing the data copies stored locally and running data verification protocol locally. Deswarte proposed a data verification protocol based on challenge-response to solve these two problems. Depending on that, Deswarte implemented a remote data integrity verification protocol based on Diffie-Hellman. In [2], Filho constructed an auditing protocol based on a homogenous hash function, which based on RSA. This authentication protocol has high security, small footprint on network resources, and can help users to verify the integrity of any part of the data. Chen in [3] wanted to construct a remote data validation scheme based on digital signature, and considered the advantages of low cost, high safety at the same time.

In 2007-2008, two basic themes of auditing protocols are proposed, which have a deep influence in the field. One is PDP (provable data possession), proposed by Atenieses in [4]. Ateniese defined the protocol framework of PDP in the research, gave out the security attributes and the security proof of related attributes, and carried out some experiments of PDP scheme on these bases. However, the PDP protocol of that version can only support the integrity verification of static data. This makes PDP less useful for cloud data, where data often changes. Ateniese has improved the PDP protocol in 2008 [5]. Compared with the efficiency of POR protocol, PDP protocol is modified to support dynamic data.
operation. Erway's in [6] specifically focused on dynamic operations for PDP, defined a system framework for dynamic provable data possession (DPDP), and improved performance by reducing the overhead of dynamically modifying data for PDP-based cloud auditing protocols using ranking information. Zhu constructed the cooperative provable data possession (CPDP) protocol in 2011 in [7]. Zhu's CPDP scheme is a cloud audit protocol based on notification verification response and subscript hash. In the article, Zhu used zero-knowledge-proof to prove the security of the system, and put forward the three security properties of CPDP: 1) integrity (completeness); 2) knowledge robustness (knowledge soundness); 3) zero-knowledge properties. In the paper, CPDP was actually deployed, and the final experimental results show that CPDP is more efficient than previous PDP schemes. Another development is POR (proofs of retrievability), which provides data restoring compared with PDP. Because such functionality is less efficient and expensive than the PDP, POR is often used for integrity verification of key user information. POR earliest from such people as Juels [8] in and complete security definition was put forward. Shacham et. al in [9] used PRFs and BLS signature scheme to carry out the POR protocol. Dodis et.al in [10] worked on POR protocol for the further safety certification, and constructed the first unbounded-use POR protocol scheme.

Subversion Attacks. In a subversion attack, a subversion attacker has the ability to insert a backdoor into a cryptographic protocol. By invading the black box function in the cryptographic protocol, the subversion attacker can steal the key information of the protocol user. The users of the auditing protocol will not find that the auditing protocol has been invaded. Once the subversion attack is successfully carried out, its destructiveness should not be underestimated.

Research on similar technologies began in the 1980s. We will introduce the research results of subversion attack below. Subliminal channel is a covert channel established by cryptographic mechanisms such as authentication or digital signature. The use of subliminal channels is not known to anyone except the sender and the recipient designated by the sender. The concept of subliminal channels was first proposed by Simmons et al in [11]. Simmons presented a model of secret communication between prisoners Alice and Bob. Alice and Bob are held in separate cells. The only way they could communicate with each other was through their prison guard Carol. Carol claims that must be visible to him. Moreover, Carol may pretend to be one of Alice and Bob from time to time to communicate with another person, so as to judge whether they have a plan to escape from prison. Therefore, in order to pass the really useful information without being noticed by the jailer Carol, Alice and Bob establish subliminal channel in the passing content \( M \). Even though \( M \) itself may be a bit of an innocent-sounding statement, Alice and Bob can still decrypt the key messages contained in \( M \) using their agreed algorithm. For example, supposed that each time Alice and Bob use parity of \( M \) as a secret message, the data Alice sends to Bob, \( M \), is parsed into a string of 01 strings. However, the limitation of this model is that the security of subliminal channels established between Alice and Bob is completely based on the confidentiality of subliminal information encryption algorithm and decryption algorithm, which is not consistent with the traditional cryptographic security. Therefore, Simmons constructs a subliminal channel based on digital signature in [12], changing the security of subliminal channel from algorithmic confidentiality to privacy based on the private key of digital signature. Simmons further demonstrated in [13] the possibility that Carol and Bob might conspire in the "cuckoo channel" model. Subliminal channel has a series of research work so far [11][14-17].

Young and Yung et al. first proposed the concept of the dark side of black-box cryptography in 1996 in [18]. Similar to subversion attack, cryptography is an attack against the cryptography mechanism itself. Cryptography extends the concept of subliminal channels to reveal key information from digital signature to more general aspects. Young proposed a technology called Secretly Embedded trapdoor with Universal protection (SETUP). Through SETUP, attackers can embed the backdoor into the black box function of cryptographic protocols, so as to secretly steal the user's privacy. Young actually deployed SETUP on RSA, El-Gamal, DSA and key exchange protocol to verify the practicality of SETUP technology. Young and Yung adopted a more complete security
definition of SETUP in [19], and proposed definitions of strong SETUP, conventional SETUP and weak SETUP.

Contributions. In this paper, we make contributions as follows.
- We propose a general frame of auditing protocol themes against subversion attacks. In the description, we give out the working flow of the theme, and prove the security of it.
- We carry out the theme on an auditing protocol to check out the practicality. We deploy the experiments based on the auditing proposal proposed in [compact of retrievability], which is a classic POR theme.

Organizations. In section 2, We give the preliminaries including the definitions of psedorandom public key encryption, auditing protocol with index-coefficient, and the frame of subversion attack auditing protocol with index-coefficient. In section 3, we show a general frame of auditing protocols against subversion attacks. In section 4, we show the details and results of the experiments. Section 5 is about conclusions.

2. Preliminaries
If $\mathcal{D}$ is a sample space, then $x \leftarrow \mathcal{D}$ means selecting a random element $x$ from $\mathcal{D}$.

2.1. Auditing Protocol with Index-coefficient Challenge
Auditing protocol with index-coefficient challenge is a special type of auditing protocol which uses sets of indexes and coefficients in its challenge. It consists of five algorithms (KeyGen,TagGen,ChallGen,ProofGen,Vrfy) which are defined as follows.

- **KeyGen** takes as input a security parameter $\lambda$. It returns a key pair $(pk,sk)$, where $pk$ is the public key and $sk$ is the private key.
- **TagGen** takes as inputs a file $M$ consists of file blocks $m_1, \ldots, m_n$, private key $sk$. For each file block $m_i$, it computes a file tag $t_i$ based on $sk$. It returns a set of file tags $T = \{t_i\}_{i \in [1,n]}$.
- **ChallGen** takes as input the abstract information of file $M_{info}$ (e.g., file identity, total number of blocks and timestamp etc.). It returns a challenge $Q$ that consists of set of indexes $I = \{i_1, \ldots, i_c\} \subseteq \mathbb{Z}_n^+$ and set of coefficients $V = \{v_i\}_{i \in I} \subseteq \mathcal{V}$ where $1 \leq i_j \leq n, 1 \leq j \leq c, 1 \leq c \leq n$.
- **ProofGen** takes as inputs challenge $Q$, file $M = \{m_i\}_{i \in [1,n]}$, file tag set $T$ and public key $pk$. It returns a proof $\sigma$.
- **Vrfy** takes as inputs the proof $\sigma$, challenge $Q$ and public key $pk$. It returns either 1 or 0. If $\sigma$ is a valid proof, it outputs 1. Otherwise, it outputs 0.

An auditing protocol should satisfy correctness which is defined as follow.

**Definition 1. (Correctness)**
Let $\Pi = (\text{KeyGen}, \text{TagGen}, \text{ChallGen}, \text{ProofGen}, \text{Vrfy})$ be an auditing protocol. We say that $\Pi$ satisfies correctness if for any file $M$, the following inequality holds.

$$
Pr[\text{Vrfy}(\sigma, Q, pk) \neq 1] \leq \varepsilon(\lambda),
$$

where $\varepsilon(\lambda)$ is negligible.

The soundness of auditing protocol is defined as follow.

**Definition 2. (Soundness)**
Let $\Pi = (\text{KeyGen}, \text{TagGen}, \text{ChallGen}, \text{ProofGen}, \text{Vrfy})$ be an auditing protocol. We say that $\Pi$ is $\varepsilon$-soundness if there exists an extraction algorithm $\text{Extr}$, that for any adversary $A$, whenever an $\varepsilon$-admissible cheating prover $\mathcal{P}'$ for a file $M$, the following holds:

$$
Pr[\text{Extr}(pk, sk, T, \mathcal{P}') = M : T \leftarrow \text{TagGen}(M, sk)] \leq \varepsilon(\lambda),
$$

where $(pk, sk)$ is the key pair of client, and $\varepsilon(\lambda)$ is negligible.
2.2. Subversion attack on auditing protocols

Let $\Pi = (\text{KeyGen}, \text{TagGen}, \text{ChallGen}, \text{ProofGen}, \text{Vrfy})$ be an auditing protocol with index-coefficient challenge and $\tilde{\Pi} = (\text{KeyGen}, \tilde{\text{Enc}}, \tilde{\text{Dec}})$ be a public key encryption scheme.

- **Setup**: KeyGen is run by client $C$ to obtain a key pair $(pk_C, sk_C)$ for $C$. KeyGen is run by cloud server (subversion attacker) to obtain a key pair $(pk_s, sk_s)$ for subversion attack. Then $C$ runs TagGen to generate tags $T = \{t_1, \ldots, t_n\}$ for file $M = \{m_1, \ldots, m_n\}$, send $\{T, M\}$ to cloud server and delete their local copies.

- **Challenge**: Given abstract information of file $M_{\text{info}}$, secret message $x$ (generally it will be $sk_C$) and public subversion key $pk_s$, challenge $Q$ is computed as follows.
  - Randomly pick $l$ different indexes $i_j \leftarrow \mathbb{Z}_n^+$ (1 ≤ $j$ ≤ $l$) and index set $I = \{i_1, \ldots, i_l\}$;
  - Choose a random bit string $r \in \{0,1\}^*$ and compute $c \leftarrow \tilde{\text{Enc}}(pk_s, x \mid r)$.
  - Choose a random bit string $r \in \{0,1\}^*$ and compute $c \leftarrow \text{Enc}(pk_s, x \mid r)$.

- **Verification**: Server $S$ recovers secret message $x$ from subverted challenge $Q$. For subverted challenge $Q = \{I, V\}$, concatenate all the coefficients in $V$, remove random bit string $r'$ and decrypt ciphertext $c$ with private subversion key $sk_s$. That is,

$$x \mid r = \text{Dec}(sk_s, c),$$

After removing random bit string $r$, server $S$ recovers secret message $x$ with overwhelming probability.

3. General frame of auditing protocols against subversion attacks

In this section, we propose a general frame of auditing protocols against subversion attacks. Besides, this kind of auditing protocol must satisfy the security attributes of auditing protocol with index-coefficient challenge. In subversion attack (SA), SA attackers would not be noticed by auditing protocols’ users. That means, the subverted version of the auditing protocol satisfies the security demands of correctness and soundness, which cannot prevent the secret information of users from being stolen by servers. In order to protect the secret information of users, we introduce a third party (TP) into our themes, who is honest but curious. In the theme, TP plays a role of transmitting challenge between users and cloud server. We define client as $C$, third party as $\mathcal{F}$ and server as $S$, then give detailed descriptions as follows.

1. \text{Client } C: (pk, sk) \leftarrow \text{KeyGen}(\{i\})
2. \text{Client } C: T \leftarrow \text{TagGen}(M, sk)
3. \text{Client } C \rightarrow \text{Server } S: \{M, T\}
4. \text{Client } C: Q_0 \leftarrow \text{CGen}(M_{\text{info}})
5. \text{Client } C: H_0 := H(V_0)
6. \text{Client } C \rightarrow \text{Third Party } \mathcal{F}: \{H_0, V_0\}
7. \text{Server } S: Q_1 \leftarrow \text{CGen}(M_{\text{info}})
8. \text{Server } S: H_1 := H(V_1)
9. \text{Server } S \rightarrow \text{Third Party } \mathcal{F}: \{H_1, V_1\}
10. \text{Third Party } \mathcal{F}: \text{if } H_0 = H(V_0) \land H_1 = H(V_1) \text{ then } V' := F(V_0, V_1)
11. \text{Third Party } \mathcal{F} \rightarrow \text{Client } C: \{V'\}
12. \text{Client } C: Q' := \{l_0, V'\}

Fig. 2 A general frames of auditing protocols against subversion attacks

In the theme, challenge generation must involve third party $\mathcal{F}$. In that case, client cannot make tricks in coefficient $V$. 


4. Implementation

In this section, we present the implementation of the frame of auditing protocol against subversion attack. Our implementation is based on the cryptography (2.0), a python package developed for some basic cryptographic functions.

Our implementation was performed on Mac os10.13.6 with 4 CPU cores, in which 16GB memory was allocated. CPU was Intel Core i7, and the version of Python is 3.7.3.

In our experiment, we generate a random file as 6MB, concluding 100000 groups of random strings, whose length would not beyond 8. Then we query 10000 in challenge phase. Depending on the MAC theme of POR in [9], we carried out the protocols for 10 times and get an average result of time cost.

| Subverted Sign | Subverted Challenge | Our Sign | Our Challenge |
|----------------|---------------------|----------|---------------|
| 60.31          | 68.85               | 61.00    | 71.00         |
| 60.24          | 67.15               | 60.95    | 54.90         |
| 61.04          | 55.90               | 61.13    | 62.00         |
| 61.00          | 67.23               | 60.93    | 64.39         |
| 61.64          | 55.77               | 65.54    | 55.49         |
| 62.01          | 63.04               | 62.81    | 69.81         |
| 62.60          | 53.44               | 62.06    | 69.55         |
| 61.86          | 59.52               | 61.30    | 49.67         |
| 64.18          | 61.37               | 59.38    | 64.31         |
| 64.01          | 57.05               | 62.40    | 64.60         |

It is easy to see that time cost in challenge phase is unstable, for the reason that in challenge phase, we randomly choose an index set $I$ for query. As the size of $I$ is random, the time cost of verification is random. We calculated the average time cost of subverted protocol and our protocol, and found that influence of time cost of communication between client and third party is far less important than the time cost of verification. So our frame of auditing protocol against cloud auditing works well.

5. Conclusion

In this paper, we explored how to defend subversion attacks in auditing protocols with coefficient index challenge. We first introduced background of auditing protocol and subversion attack. Then we show how to defend subversion attack in auditing protocols. Finally, we carried out both subversion attack and defending measures on an auditing protocol and show the time cost of them. Our purpose of the paper is to reveal a potential attack in cloud auditing protocol and to encourage more research on it.

References

[1] Deswarte Y, Quisquater J J, Saida Ne A. Remote integrity checking[C]//Working Conference on Integrity and Internal Control in Information Systems. Springer, Boston, MA, 2003: 1-11.

[2] Gazzoni Filho D L, Barreto P S L M. Demonstrating data possession and uncheatable data transfer[J]. IACR Cryptology ePrint Archive, 2006, 2006: 150.

[3] Chen L. Using algebraic signatures to check data possession in cloud storage[J]. Future Generation Computer Systems, 2013, 29(7): 1709-1715.

[4] Ateniese G, Burns R, Curtmola R, et al. Provable data possession at untrusted stores[C]//Proceedings of the 14th ACM conference on Computer and communications security. Acm, 2007: 598-609.

[5] Ateniese G, Di Pietro R, Mancini L V, et al. Scalable and efficient provable data possession[C]//Proceedings of the 4th international conference on Security and privacy in communication netowrks. ACM, 2008: 9.

[6] Erway C C, Kupcu A, Papamanthou C, et al. Dynamic provable data possession[J]. ACM Transactions on Information and System Security (TISSEC), 2015, 17(4): 15.
[7] Zhu Y, Hu H, Ahn G J, et al. Cooperative provable data possession for integrity verification in multicloud storage[J]. IEEE transactions on parallel and distributed systems, 2012, 23(12): 2231-2244.

[8] Juels A, Kaliski Jr B S. PORs: Proofs of retrievability for large files[C]//Proceedings of the 14th ACM conference on Computer and communications security. ACM, 2007: 584-597.

[9] Shacham H, Waters B. Compact proofs of retrievability[C]//International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2008: 90-107.

[10] Dodis Y, Vadhan S, Wichs D. Proofs of retrievability via hardness amplification[C]//Theory of Cryptography Conference. Springer, Berlin, Heidelberg, 2009: 109-127.

[11] Simmons G J. The prisoners’ problem and the subliminal channel[C]//Advances in Cryptology. Springer, Boston, MA, 1984: 51-67.

[12] Simmons G J. The subliminal channel and digital signatures[C]//Workshop on the Theory and Application of Cryptographic Techniques. Springer, Berlin, Heidelberg, 1984: 364-378.

[13] Simmons G J. Cryptanalysis and protocol failures[C]//Proceedings of the 1st ACM conference on Computer and communications security. ACM, 1993: 213-214.

[14] Burmester M V D, Desmedt Y. All languages in NP have divertible zero-knowledge proofs and arguments under cryptographic assumptions[C]//Workshop on the Theory and Application of Cryptographic Techniques. Springer, Berlin, Heidelberg, 1990: 1-10.

[15] Burmester M, Desmedt Y G, Itoh T, et al. Divertible and subliminal-free zero-knowledge proofs for languages[J]. Journal of Cryptology, 1999, 12(3): 197-223.

[16] Burmester M, Desmedt Y G, Itoh T, et al. A progress report on subliminal-free channels[C]//International Workshop on Information Hiding. Springer, Berlin, Heidelberg, 1996: 157-168.

[17] Desmedt Y. Subliminal-free sharing schemes[C]//Proceedings of 1994 IEEE International Symposium on Information Theory. IEEE, 1994: 490.

[18] Young A, Yung M. The dark side of “black-box” cryptography or: Should we trust capstone?[C]//Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 1996: 89-103.

[19] Young A, Yung M. Kleptography: Using cryptography against cryptography[C]//International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 1997: 62-74.