科技部補助專題研究計畫成果報告
期末報告

以ECC為基礎的雲端檔案完整保全之綜合性策略研究

計 畫 類 別 ： 個別型計畫
計 畫 編 號 ： MOST 104-2221-E-343-002-
執 行 期 間 ： 104年08月01日至106年01月31日
執 行 單 位 ： 南華大學資訊管理學系

計 畫 主 持 人： 周志賢

計畫參與人員： 碩士班研究生-兼任助理人員：廖一清
碩士班研究生-兼任助理人員：林哲宇
其他-兼任助理人員：屈宜靜

報 告 附 件 ： 出席國際學術會議心得報告

中 華 民 國 106 年 04 月 28 日

中 文 摘 要 ： 隨著雲端儲存體服務日益進步，且經由雲端資料的稽核方案可以讓使用者有效地稽核他們儲存在雲端伺服器裡面的資料有無完整性，以確保雲端伺服器是誠實的，因此客戶才可以完全確定放在雲端伺服器裡的資料沒有被任意竄改或更動，另外資料擁有者也能夠確保資料上傳到雲端伺服器時有進行資料的加密，因此公開稽核的服務方案是非常重要的。在本文中，我們提出一個有效率的公開稽核方案，主要是為了證明出惡意雲端伺服器不能偽造出不存在的消息塊，而且能讓客戶可以完全地相信雲端伺服器裡面的資料是安全的，另外我們還有做出證明計算大量資料的正確完整性，最後我們有實現證明我們的協議是非常有效率與安全的。

中 文 關 鍵 詞 ： 雲端儲存體, 公開稽核, 橢圓曲線密碼系統, 資料完整性

英 文 摘 要 ： The cloud storage service grows rapidly mainly due to that cloud data auditing schemes can enable cloud users to effectively verify the integrity of their outsourced data. For ensuring that their data is faithfully stored by the cloud server, the function correctness of the auditing scheme is extremely important. If the auditing function is correct, the user can completely assure that the stored data is untampered or inadvertently changed. In this paper, we propose such an efficient scheme, where a malicious server can’t forge a nonexistent message block to be successfully verified by a challenger. The proposed scheme allows not only for several blocks which is the case for many research in the literature, but also for batch files stored in the cloud. The proof shows that our protocol is correct and can be fulfilled very efficiently.

英 文 關 鍵 詞 ： cloud storage, public auditing, elliptic curve, data integrity

A new efficient public auditing scheme for cloud storage

Jue-Sam Chou [1] and Zhe-Yu Lin*[2]

[1] Department of Information Management, Nanhua University, Taiwan
*: corresponding author: jschou@mail.nhu.edu.tw
Tel: 886+ (0)5+272-1001 ext.56536

[2] Department of Information Management, Nanhua University, Taiwan
10369017@nhu.edu.tw

Abstract

The cloud storage service grows rapidly mainly due to that cloud data auditing schemes can enable cloud users to effectively verify the integrity of their outsourced data. For ensuring that their data is faithfully stored by the cloud server, the function correctness of the auditing scheme is extremely important. If the auditing function is correct, the user can completely assure that the stored data is untampered or inadvertently changed. In this paper, we propose such an efficient scheme, where a malicious server can't forge a nonexistent message block to be successfully verified by a challenger. The proposed scheme allows not only for several blocks which is the case for many research in the literature, but also for batch files stored in the cloud. The proof shows that our protocol is correct and can be fulfilled very efficiently.

Keywords: cloud storage, public auditing, elliptic curve, data integrity

1. Introduction

Cloud computing provides several benefits such as, broad network access rapid elasticity, measured service, on-demand self-service, and resource pooling as mentioned by NIST [37-39]. It has significantly changed the way of computing resources usage by providing dynamic service model for resource usage via the internet which mainly resorts to the broad network access, and can be configured to share resources in a timely manner thought the network. It also offers the possibility of improving system management efficiency and changes the current hardware and software design type for computer utilization. In a cloud, after the client had stored data, he typically does not retain the original data mainly

due to the data consistency maintenance. This is because the client may cooperate on the stored data with his partners at different places. Therefore, he must assure that any device can get the latest version of the data. In other words, for the data consistency, client always trust the cloud service provider on ce he had stored the data. It's therefore unnecessary for the client to maintain the data file. To reach such a scenario, other than the agreements in the contractor signed by the client and server, designing an efficient and correct auditing mechanism is very important.

A good cloud contractor needs to focus on the following three issues:
1. Better storage management efficiency.
2. To reduce a lot of hardware and software cost.
3. The client is allowed to access the data anywhere anytime.

Recently, the public auditing technology for ensuring data integrity is obtaining more and more attention due to people who may be equipped with different terminals such as phones, tablets, laptops, desktop computers, and some other devices can take advantage of low-cost cloud storage to store their data anytime anywhere. That is, they can access personal information which may be stored in large multinational corporations, independent on their real locations. With cloud computing, you can also build a private cloud, and work with multiple partners to fulfill collaborative design, product development, or order processing. The processing result can be instantly shared. Even, it can be applied to the public sector for public information publishing and many more. Although, cloud computing makes our life more convenient, it brings us new security and privacy challenges due to that it is on the air. For this reason, many persons do not want to use cloud storage due to the serious security problems. They concern about the integrity of the outsourced when facing several factors that may result in data corruption. First, the cloud service providers are usually not fully trusted. They may update data without notifying the data owners. Second, the stored data may be broken because of the cloud server's fails, management errors, or the adversary attack, but to preserve the good service reputation, cloud service provider may hide the data loss event. Therefore, the issues of data leakage and integrity on the cloud storage have become the main concerns of client. How to determine that the data stored in the cloud is complete and safe is a important issue. Without maintaining the data file at the client side, the clients will lose the control of the data. This leads to the untrustness of the client to the cloud storage provider. Therefore, the cloud data integrity checking is necessary. To this end, there has been many researchers [1-3, 5-8, 9, 12,13, 15, 20,

21,31-33] working in this field, attempting to resolve the problem. In this article, we refer to these related technologies as Provable Data Possessing (PDP).

However, this study found that most of the PDP technologies in the literature are implemented with higher computation cost, because they used expensive cryptographic operations such as, RSA, bilinear pairing cryptosystems to fulfill their scheme (Pairing-Based Cryptosystem is referred to as PBC). According to National Institute of Standards and Technology (NIST) recommendations for the same security level, a RSA cryptosystem with key length 1024/2048 bits is equivalent to an elliptic curve cryptosystem with only 163/233 bits. This means that the output length of elliptic curve cryptosystem which directly relates to the communication cost is about 1/6 to 1/8 to the RSA cryptosystem. According to the literature [35-36], the computation cost of RSA cryptosystem is about 3.2 to 6 multiple to the ECC with the same security level. In Table 1, we compare three cryptosystem in the dimensions of communication cost and computation costs. From the table shown, we can easily see that ECC obviously has advantages over the other two.

Table 1：Comparisons of 3 type's systems' communication and computation cost

cryptography system	Communication costs	Computing costs
Elliptic-Curve Cryptosystem	1	1
RSA	6~8	3.2
Pairing-Based Cryptosystem	1	7.5

Up to now, we only see one design [41] using lower cost ECC cryptosystem. Unfortunately, we found that their scheme doesn't possess the whole nine properties mentioned in [48].

2. Literature review

In cloud computing environment, users can get rid of their own storage maintenance burden. They rely this function on the cloud service provider (CSP). CSP thus must have a trustable, manageable, and effective accessible storage infrastructure for clients to create, store, and update data. In the recent research [22,29,32,35,41-47], all assume that the CSP is partially trusted. Under this assumption, it's possible that the data's integrity is defective, because the client lacks the control of the outsourced data. Therefore, it's necessary for the client to perform the integrity checking for his outsourced data. However, often, the client computation capability is limited. This is

especially the case when the mobile devices become popular. To resolve this problem, a third party with better expertise and capabilities is delegated to measure the cloud storage reliability and validity on behalf of the clients when needed. This model is called public auditing.

In this model, there are three entities: Client, Third party auditor (TPA), cloud server. Among them involves the information transfer process. The client delegates the right of data integrity verification to the TPA. TPA will challenge the cloud server for obtaining the integrity proof. Subsequently, after receiving the proof from the server, TPA will return this proof to the client, as illustrated in Fig.1. Other than the integrity checking, there also exists a research area known as privacy protection public auditing (PPPA) for cloud storage system [1-27] which encrypt the stored data e except for the function of public auditing. But this research field is not the focus of our design. We only concern the PDP scheme s.

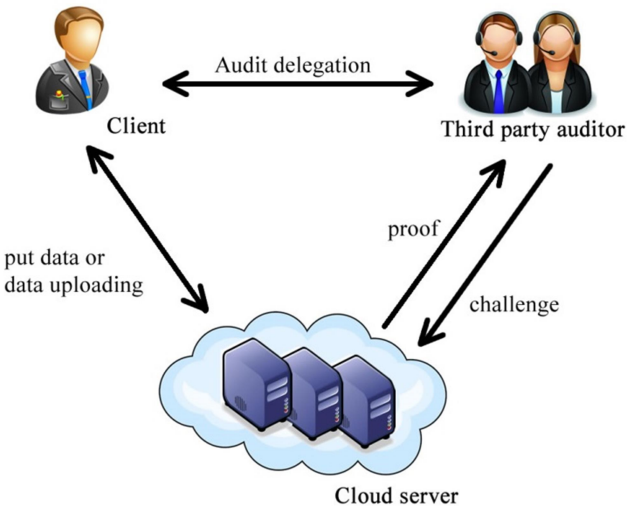


Figure 1：A traditional public auditing system model

Wang et al.[34] proposed a Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in the cloud. He can give third-party auditors (TPA) users the ability to verify the integrity of their data without retrieving the entire data. In addition, Zhu et al.[8]propose an efficient approach based on probabilistic query and periodic verification for improving the performance of

audit services. They claimed that their scheme is can support provable updates to outsourced data, and timely abnormal detection. But also leakage of the user's secret and some of the documents of the dynamic and efficient security audit services, which also allows the attacker has the advantage of easy pass.

As mentioned earlier, for blinding the server chooses a random element $r \in_R Z_p$ by using the same pseudorandom function and let $r = f_{k3}(chal)$, where k_3 is a pseudorandom function key generated by the server for each auditing. It then calculates $R = u^r \in G$ and computes $\mu^* = \sum_{i=s_1}^{s_c} v_i m_i$, $\mu = \mu^* + \mathrm{rh(R)} \in Z_p$, and $\sigma = \prod_{i=s_1}^{s_c} \sigma_i^{v_i}$.

From the received (μ, σ, R), we can see that since $\sigma = \prod_{i=s_1}^{s_c} \sigma_i^{v_i}$ $= \prod_{i=s_1}^{s_c} (H(i)^{v_i} \cdot u^{\sum_{i=s_1}^{s_c} v_i m_i})^x$, a malicious server can regard $v_i s$ as constants and $m_i s$ as variables. He then computes $\mu^* = \sum_{i=s_1}^{s_c} v_i m_i$ using the constants $v_i s$ and the message blocks stored. That is, he can obtain an equation containing multiple variables, the $m_i s$, which in mathematics has more than one solution. This means that other than the original $m_i s$, the malicious server can find out some message blocks satisfying the equation without alerting σ. We take S_c=3 as an example. Suppose the values of $v_i s$ are (6, 8, 9), and the values of $m_i s$ are (1, 4, 2) respectively, then the plane can be defined by $6x + 8y + 9z = 56 (= 6m_1^* + 8m_2^* + 9m_3^*)$, where $m_i^*, i=1 \text{ to } 3$, are the forged message blocks. We know that this plane also passes through the point (5, 1, 2). This implies that the malicious server can forge the message blocks from (1, 4, 2) to (5, 1, 2) without alerting the value σ. Moreover due to the independence between $\mu^* (= \sum_{i=s_1}^{s_c} v_i m_i)$ and R, the malicious server can even set $R' = u^{r'}$ and $\mu' = \mu^* + r' h(R') \in Z_p$ and sends (μ', σ, R') to *TPA*. *TPA* will accept the verification without detection. That is, the proof of the selected blocks is not unique. This might lead the scheme incur more vulnerabilities.

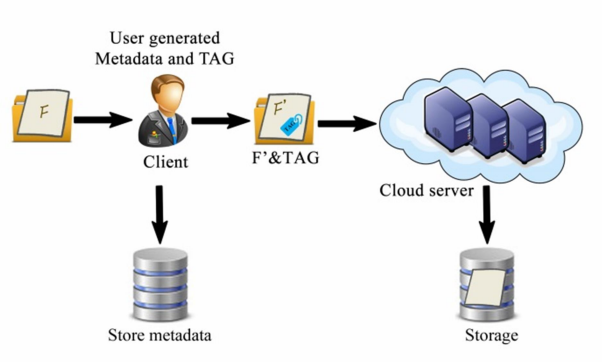
Recently, several articles proposed also have the same problem. For the interested r eaders, please refer to [34,40-45].

3. Security requirements in the system model

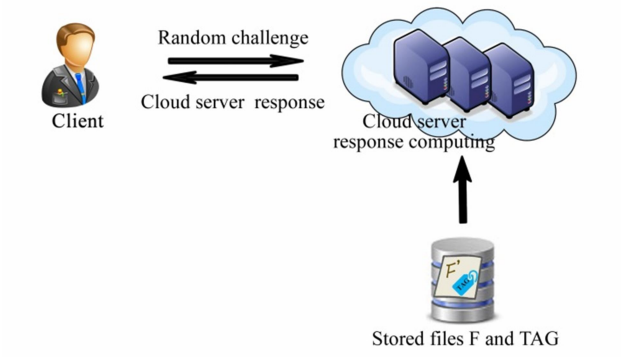
In this section, we describe the cloud storage system model and its security requirements. Then, we illustrate the goal of our design.

3.1 System model

The following will describe the cloud storage integrity checking model. The scenario is shown in Fig.2. We divide it into two parts: (a) data file preprocessing (b) data file verification.



(a) Data file preprocessing



(b) Verification the data file

Figure 2：Data has to prove technology architecture

(a) Data file preprocessing

When the client wants to upload the file F into the cloud, he first needs to generate the metadata, for instance cutting the file into blocks of the same size, then computing

the corresponding metadata called tags. After this, the client will store F and all its tags to the cloud server and then delete them from its own storage.

(b) Data file Verification

When auditing, according to the metadata stored in the server's storage, the cloud server read out the corresponding file blocks and their tags to compute the proof. If the proof is correct, the cloud server is considered honesty.

3.2 Security requirement

In the model, three important security threats are defined [41, 49]. We show and explain them as follows.

(1)Impersonation attack: Although, the attacker hasn't the blocks and the corresponding tags, he impersonates the cloud server trying to respond the correct proof.

(2)Replay attack: The malicious cloud server or attacker replays the previously record proof to fool the client, but indeed has modified the corresponding blocks secretly.

(3)Forgery attack: The malicious cloud server may forge a block m_i with its tag to satisfy the verification challenged from the client.

3.3 Our design goal

In this section, we present a new ECC-based auditing scheme and prove its security. Our design goal is mainly to reduce the verification cost at the client side, except for meeting the security requirements of a storage auditing scheme. In addition, we also have the followings as our design goal:

1. Our preliminary construction is to design only for a single file block to satisfy proof verification. Then, it can be applied to the whole file's integrity checking.
2. It has minimum communication and computation cost.
3. The proof can be certified effectively. That is, once a client had retrieved damaged data, he can instantly identify it.
4. The role of auditor is moved from TPA to the user.

4. The proposed scheme

To ensure the integrity of stored data, we propose a new efficient auditing scheme for cloud storage. We first define and show a list of used notations in Table2.

Table 2：The definitions of used notations

Notations table	
G_1	an additive cycle group on an Elliptic Curve with order q
$H_1(\cdot)$	a hash function mapping from $G_1 \rightarrow \{0,1\}^*$
$H_2(\cdot)$	a hash function mapping from $\{0,1\}^* \rightarrow Z_q$
s	client secret key
Y	client public key
m_i	the ith block of the stored cloud data file, that is $F=\{m_1,\ldots,m_n\}$
r_i	a random integer in Z_q^*
r_{i1}	a nonce chosen by the client
r_{i2}	a nonce chosen by the server
T_2	a timestamp
$Proof_i$	the proof a message sent to the client by the cloud server
P	a generator of group G_1
i,k_i	the challenge message sent to the cloud server by the client

Then, we illustrate our scheme as follows, It consists of four phases:

1. Setup phase: This phase includes the client private key and client public key generations.
2. TagBlock phase: In this phase, the client performs the data file pre-processing to generate block-tags. The client will then store the file with all its tags to the cloud server. He then deletes the stored file.
3. Challenge phase: In this phase, the client sends a challenge to the server, The server generates an integrity proof corresponding to the set of challenged blocks, and returns it to the client.
4. ChallengeVerify phase: In this phase, when the client receives the proof from the server, he will perform calculations to see whether it is correct.

We now describe the four phases in our two protocols: (1) for a single block and (2) for a file, in details in the following and also show them in Figure3 and Figure 4, respectively.

4.1 Setup phase

Let G_1 be an additive cyclic group of prime order q and P be a generator of G_1 where q is a large prime which is at least 160 bits or more. Define two cryptographic hash functions with the mapping: H_1: $G_1 \rightarrow \{0,1\}^*$ and H_2: $\{0,1\}^* \rightarrow Z_q$. The system initialization generates the client secret key $s \in Z_q^*$, and its public key is $Y = sP$.

Afterwards, the system publishes the system parameters $\{G_1, P, q, Y, H_1, H_2\}$.

4.2 Tagblock generation phase

The main purpose of this phase is to verify whether the stored file block or blocks had been stored honestly. The preliminary step is to generate the corresponding tag for a single blocks m_i in a file F. The client first divides file F into n blocks of equal size.

We denote it as $F=\{ m_i \| 1 \le i \le n \}$. After that, for each block m_i the client generates its tag by choosing a random number $r_i \in Z_q$ and computes its $Tag_i\ (d_i, A_i, B_i, c_i, D_i)$, where as $d_i=H_1(sr_im_i)$, $A_i=r_i\ m_iP$, $B_i=r_iY$, $c_i = H_2(s\|H_1(A_i))m_i$ and $D_i=c_i\ B_i$. The client then calculates

$$v_i = r_i\, m_i + sH_1(B_i) + r_i\, d_i\, s \qquad \text{....(1)}$$

He then sends v_i, m_i, Tag_i to the server for its verifications. After receiving, the server checks to see whether $v_iP =? A_i + H_2(B_i)Y + d_iB_i$ holds. If it does, the cloud server stores m_i together with its tag Tag_i into the storage. We demonstrate the correctness of equation **(1)** as follows.

$$v_iP=(r_i\, m_i + sH_1(B_i) + r_i\, d_i\, s)P = A_i + H_1(B_i) + d_i\, B_i$$

4.3 Challenge phase

After the storing of m_i, and Tag_i for $1 \le i \le n$, the client can examine whether the cloud server honestly store his delegated file blocks by randomly selects $i \in 1 \le i \le n$ and k_i, r_{i1} to challenge the server. Subsequently, the cloud server randomly selects r_{i2} and computes $l_i=r_{i1}r_{i2}\ d_i$.and $R_i = r_{i1}\ k_i\ d_i\ m_i\ B_i$. It then uses the system timestamp T_2 to calculate $t_i = H_2(T_2)$, $Q=t_i\ r_{i1}\ r_{i2}Y$, $w_i=H_1(t_i\ r_{i1}r_{i2}P)$, and $a_ip = r_{i1}r_{i2}(d_i\, p + t_i\, H_2(k_i\ d_i\ A_i))P$

The server then generates $Proof_i = \{ l_iP, R_i, a_iP, Q, w_i, T_2, t_i, t_i^{-1}, Q\}$ and sends to the client for the client's checking.

4.4 Proof verification phase

In this phase, we demonstrate two kinds of verifications: (1) a single block verification and (2) batch verifications for a file.

(1)Single block verification

After receiving $Proof_i = \{ l_iP, R_i, a_iP, Q, w_i, T_2, t_i, t_i^{-1}, Q\}$, the client computes and

Client (File)	**P: generator**	**Server**
Step phase		
Privkey = s	H_1: $G_1 \rightarrow \{0,1\}^*$	
Pubkey = $Y = sP$	H_2: $\{0,1\}^* \rightarrow Z_q$	
TagBlock generation phase		
$d_i = H_2(r_i m_i s)$, $A_i = r_i m_i P$, $B_i = r_i Y$,		
$c_i = H_2(s \| H_1(A_i)) m_i$		
$D_i = c_i B_i$		
$v_i = r_i m_i + s H_1(B_i) + r_i d_i s$		
m_i, d_i, A_i, B_i, v_i, c_i, D_i		
	m_i, d_i, A_i, B_i, v_i, c_i, D_i →	verify $v_i P =? A_i + H_2(B_i) Y + d_i B_i$
		store m_i, $Tag_i = \{d_i, A_i, B_i, c_i, D_i\}$
Challenge phase		
For each i, $1 \le i \le n$		
choose r_{i1} and k_i	i, k_i, r_{i1} →	choose r_{i2}
		compute
		$l_i = r_{i1} r_{i2} d_i$
		$t_i = H_2(T_2)$
		compute t_i^{-1}
		$R_i = k_i d_i m_i B_i$
		$a_i P = r_{i1} r_{i2} (d_i + t_i H_2(k_i d_i A_i)) P$
		$Q = t_i r_{i1} r_{i2} Y$
		$w_i = H_1(t_i r_{i1} r_{i2} P)$
		$Proof_i = \{ l_i P, R_i, a_i P, Q, w_i, T_2, t_i, t_i^{-1} \}$
	← $Proof_i = \{ l_i P, R_i, a_i P, Q, w_i, T_2, t_i, t_i^{-1}, Q \}$	
Obtain $Proof_i$		
Checks to see the freshness of T_2		
Compute $t_i ? = H_2(T_2)$		
$H_1(s^{-1} t_i^{-1} Q) ? = w_i$		
ProofVerify phase		
$l_i P =? a_i P - t_i r_{i1} r_{i2} H_2(s^{-1} R_i) P$		

Figure 3：Data block m_i integrity checking.

Client (File)	P: generator	Server
Step phase		
Privkey = s	H_1: $G_1 \rightarrow \{0,1\}^*$	
Pubkey = $Y=sP$	H_2: $\{0,1\}^* \rightarrow Z_q$	
file verification		
Chose r_1 and k_i	$i, k_i, r_1, 1 \le i \le n$ →	
		choose r_2
		$t = H_2(T_2)$
		compute t^{-1}
		$R_i = r_i k_i d_i m_i B_i$
		$l_i = r_1 r_2 d_i$
		$Q = t r_1 r_2 Y$
		$w = H_1(t r_1 r_2 P)$
		$lP = \sum_{i=1}^{i} l_i P = r_1 r_2 \sum_{i=1}^{i} d_i$
		$R = \sum_{i=1}^{i} R_i$
		$ap = lP + t r_1 r_2 H_2(\sum_{i=1}^{i} k_i d_i A_i)P$
	$Proof_i = \{ lP, R, aP, Q, w, T_2, t, t^{-1}Q \}$ ←	
Obtain *Proof*		
Check to see the freshness of T_2		
Compute $t ?= H_2(T_2)$		
$H_1(s^{-1} t^{-1} Q) ?= w$		
ProofVerify phase		
$lP ?= ap - H_2(s^{-1}R)Q$		

Figure 4：Batch verification for a file F

checks to see whether $t_i=H_2(T_2)$. If so, he computes to see whether $H_1(s^{-1}t_i^{-1}Q) = w_i$ holds. If it holds, this implies that the client's challenge is realtime. The client them checks weather the following equation holds

$$l_i P =? a_i P - t_i r_{i1} r_{i2} H_2(s^{-1} R_i) P$$

$$= a_i P - t_i r_{i1} r_{i2} H_2(s^{-1} k_i d_i m_i B_i) P$$

$$= a_i P - t_i r_{i1} r_{i2} H_2(s^{-1} k_i d_i m_i r_i s P) \text{P}$$

$$= a_i P - t_i r_{i1} r_{i2} H_2(k_i d_i m_i r_i P) P$$

$$= r_{i1} r_{i2} d_i \mathrm{P} + t_i r_{i1} r_{i2} H_2(k_i d_i A_i) \mathrm{P} - t_i r_{i1} r_{i2} H_2(k_i d_i m_i r_i P) P$$

$$= l_i \mathrm{P} + t_i r_{i1} r_{i2} H_2(k_i d_i r_i m_i \mathrm{P}) \mathrm{P} - t_i r_{i1} r_{i2} H_2(k_i d_i m_i r_i P) P$$

$$= l_i \mathrm{P} \qquad \textbf{(2)}$$

If it does, the client believe that his message block m_i is honestly stored by the server.

(2) Bath verification for a file

The client can also perform batch auditing for the whole file F by launching the challenge for each blocks. The server computes $t = H_2(T_2)$, t^{-1}, $Q = tr_1 r_2 Y$,
$w = H_1(tr_1 r_2 P)$, $l = r_1 r_2 d_i$,
$l_i = r_1 r_2 d_i$
$lP = \sum_{i=1}^{i} l_i P = r_1 r_2 \sum_{i=1}^{i} d_i P$,
$R = \sum_{i=1}^{i} R_i = \sum_{i=1}^{i} l_i \ d_i \ m_i \ B_i$,
$aP = r_1 r_2 \sum_{i=1}^{i} d_i \ P + t \ r_1 r_2 \ H_2(\sum_{i=1}^{i} k_i \ d_i \ A_i) P$.

The server then transfers these parameters lP, R_i, aP, Q, w, T_2, t, t^{-1}, Q, to the client for verification. The client verifies whether T_2 is in time. If so, he computes to see whether $H_1\left(s^{-1} t^{-1} Q\right) = w$ holds. If it holds, this implies that the client's challenge is realtime. The client then checks whether the following equation holds.

$$lP \stackrel{?}{=} aP - tr_1 r_2 H_2(s^{-1} R) P$$

$$\stackrel{?}{=} aP - H_2(s^{-1} R) Q \qquad \textbf{....(3)}$$

If the equation holds, the client believe that the data is well maintained by the server. The protocol is illustrated in Figure 4.

5. Security analyses

In this section, we will show that our two auditing protocols (1) for a single block and (2) for a file, are secure by inspecting them with three security features: (1) Impersonate attack, (2) Reply attack, and (3) Forgery attack.

5.1 For a single block

(1) Impersonate attack

Assume that an attacker E can successfully forge $proof_i$ without block m_i. That is, E can pretend the server to respond with correct $proof_i$ without m_i and its tag, once he had received the challenge from the client. However, without m_i and its tag, E cannot compute l_ip, R_i, a_iP to pass the client's verification $l_iP =? a_iP - r_{i1}r_{i2}H_2(s^{-1}R_i)P$. Since c_i, d_i, v_i in the tags are embedded with the client's secret s.

(2)Replay attack

Assume that attacker E had record the two communication messages transferred between the client and server, and launches a replay attack. That is, when a client challenged the server with the same message 1, he can replay message 2 to fool the client that he is the cloud server. However, in our protocol, the timestamp T_2 and its hash $H_2(T_2)$ can thwart such an attack. Because the client will examine the freshness of T_2 which is binded into Q to be checked by the client by using the equation $H_1(s^{-1}t^{-1}Q) ?= w_i=H_1(t_ir_{i1}r_{i2}P)$. If E tries to modify Q by first multiplying it with t_i^{-1} then using another timestamp T_3 to compute $t_i' = H_2(T_3)$ and computes $Q = t_i'r_{i1}r_{i2}Y$. However, T_3 cannot pass the client's freshness checking. Since $H_1(s^{-1}t_i^{-1}Q)$ is not equal to w_i.

(3) Forgery attack

Forgery attack means that an attacker wants to forge block m_i's tag without the block. That is, he can choose a random forged block m_i', and compute its tag $d_i=H_1(r_i'm_i' s)$, $A_i=r_i'm_i'P$, $B_i=r_i'Y$ successfully. However, without the knowledge of client secret s, E cannot compute m_i's tag c_i, d_i, v_i for generating the proof to pass the client's verification.

5.2 For a file

(1) Impersonate attack

Assume that an attacker E can successfully forge *proof* without having all the blocks m_is of file F. That is, E can pretend the server to respond with correct *proof* without F and all its tags once he had received challenge from the client. However, without F and its tags, E cannot compute lp, R, aP, Q to pass the clients verification $lP =? aP - H_2(s^{-1}R)Q$.

Since c_i, d_i, v_i in all the tags are embedded with the client's secret s.

(2)Replay attack

Assume that attacker E had record the two messages transferred between the client and server and launches a replay attack. That is when a client challenged the server with the same message 1, he can replay the record message 2 to fool the client that he is the cloud server, However, in our protocol, the timestamp T_2 can thwart such an attack. Because the client will examine the freshness of T_2 which is binded into Q to be checked by the client by using the equation $H_1(s^{-1}t^{-1}Q) =? w=H_1(tr_1r_2P)$. If E tries to modify Q by first multiplying it with t^{-1} then using another timestamp T_3 to compute $t' = H_2(T_3)$ and $Q= t'r_1r_2Y$. However, T_3 cannot pass the freshness checking of the client. Not to mention that $H_1(s^{-1} t^{-1} Q)$ does not equal to w.

(3) Forgery attack

Forgery attack means that an attacker wants to forge block *F's* tags without the blocks. That is, he can choose a random forged block m_i', and compute its tag $d_i=H_1(r_i'm_i' s)$, $A_i=r_i'm_i'P$, $B_i=r_i'Y$, v_i, c_i successfully. However, without the knowledge of *s*, E cannot compute $m_i's$ tag d_i, r_i and c_i for generating the proof to pass the client's verification.

6. Comparisons and Discussion

In this section, we first compare our proposed scheme with some others in the literature, in the aspects of nine properties which are insisted by Garg et al. [48]. Then, discuss several issues about the advantages of employing ECC cryptosystem in the secure cloud storage public auditing design. Finally, we make a discuss to our proposed scheme.

6.1 Comparisons

We describe the reason why our scheme can satisfy the following nine properties which an ideal data auditing protocol should own as described in [48]. We compare then our scheme with the other related works in these properties and show the comparison result in Table 3. From Table 3, we can see that our scheme outperforms the others in the compliance to the nine properties schemes.

1. Batch auditing: Our scheme provides batch auditing, as shown in equation (3).
2. Public auditing: As long as we substitute the role of user to third party, our scheme

also provide public auditing. That is, the secret s which originally possessed by the user is now set to be the third party's. The other operations are kept unchanged.

3. Blockless verification: In our scheme, no actual data blocks are sent by the server to the client for verification. The proof comprises of only the points on an elliptic curve and the hash values. Thus, our scheme is blockless verification in the consideration of communication overhead.
4. Privacy Preservation: The secret of the client who is now actually the auditor is unknown to the others.
5. Data Dynamics: Data dynamics means when the data owner stores data into the cloud, the data owner can access the data to perform certain operations such as, insertion or deletions anytime anywhere. Clearly, our scheme possess this property as long as the client recomputes the corresponding tags. Therefore, our scheme supports the data owner operating on the data dynamically.
6. Unbounded number of challenges: In our proposal, the client can launch unlimited number of challenges to the server. Therefore, our scheme meets this requirement.
7. Non reproducibility: In our scheme, the server cannot pass the verification without the owner's actual data m_i and its tag. If the server modifies a data block without alerting its tag, then the proof cannot pass the client's checking. To alert its tag, it must have the client's secret s assistance.

From Table 3, we can see that our scheme outperforms the others in the above nine properties which an ideal auditing scheme should possess, insisted by Garg et al. [48].

Table 3：A performance comparison

Protocols	Batch auditing	Public auditability	Blockless verification	Privacy preservation	Data dynamics	Unbounded number of challenges	ECC
Ateniese et al. (2007) [2]	No	Yes	Yes	No	No	No	No
Ateniese et al. (2008) [3]	No	Yes	Yes	No	No	No	No
Wang et al. (2013) [6]	No	No	No	No	No	No	No
Erway et al. (2009) [12]	No	No	No	No	Yes	Yes	No
Zheng and Xu (2012) [18]	No	Yes	No	No	No	No	No
Wang et al. (2009) [20]	No	No	No	No	No	No	No
Wang et al., (2011) [23]	Yes	Yes	Yes	No	Yes	Yes	No

Zhu et al. (2011) [27]	No	Yes	No	No	No	No	No
Yang and Jia (2013) [28]	Yes	Yes	No	Yes	Yes	No	No
Hao et al. (2011) [30]	No	Yes	No	Yes	Yes	No	No
Zhang et al. (2015) [41]	No	Yes	No	No	No	No	Yes
Ours	Yes	Yes	Yes	Yes	Yes	Yes	Yes

8. Recoverability: This means there should exist mechanism which can spring back the original data by using available data. In our method, as the long as the server gives the block m_i's $Tag_i = \{d_i, A_i, B_i, v_i, c_i, D_i\}$, the client can recover mi by computing $m_i = c_i H_2^{-1}(s \,\|\, H_1(A_i))$. Of course, firstly the client should check whether the equation $D_i = c_i B_i$ holds. If it does, that means ci is kept unchanged.
9. Adaptability: Our protocol complies with virtual machines dynamic mutability, due to the secrets *s* is only known to the client.

6.2 Discussions

In this paper, we use ECC to design a flexible, more practical data storage integrity auditing protocol for mobile devices. This mainly resorts to that ECC cryptography key length is far more less than the other public key cryptosystems such as RSA, Elgamal, Bilinear pairing, and thus is far more fast to process with the same security level. Hence, it's more suitable for applications such as, smart cards, mobile phones, wireless memory devices, and NFC resource limited devices. It can operate efficiently with minimum computation and communication overhead, because it uses only two passes in the communication. As for the computation cost of proof verification for a file, it uses only one point multiplication and two hash operations in the freshness confirmation, and two point multiplications, one hash, and one subtraction operations in the proof verification phase. Totally, it needs only one subtraction, 3 hashes, and 3 point multiplications, in the proof verification. Therefore, our scheme not only can save the client's cost in delegating the auditing to the third party, but also can be implemented very efficiently. This allows the client to do any audit conveniently. Although, we do not adopt the public auditing architecture, it is quite easy to transfer our scheme to public audit, We can simply adapt the client's secret s to be the third party auditor's secret.

6.3 Future work interoperability

Near Field Communication (NFC) is a short-range high-frequency wireless communication technology that allows non-contact, peer-to-peer data transmission between electronic devices, and can operate fast and smoothly. With attractive

properties, such as interoperability and portability smart phones embedded with NFC become very popular. It brings mobile device users dynamic usage experience which lets them operate in an interactive wireless way. At present, many researchers of NFC technology have gradually developed it with smart card (SC) to form the mobile payment scheme on the market. In the mobile payment life cycle, the data is from the mobile device through the wireless network to reach the payment platform. Then, the payment instructions on the card are implemented to complete the payment action. Due to the resource limited environments of mobile devices, in the future work, we will use ECC to design an efficient e-cash transaction scheme using the card emulation mode of NEC. We with the same security level, which otherwise needs more complex computations when using the other cryptosystems such as, RSA, Bilinear pairing, and so on.

7. Conclusions

In this paper, we propose an efficient auditing scheme in cloud computing using ECC cryptosystem, which allows the client to audit the outsourced data efficiently. Our scheme not only has computation and communication advantages, but also satisfies the nine desirable properties for an ideal data integrity auditing protocol as insisted in Gary et al, which are not yet fully achievable at present, as shown in Table 3. In addition, we also show that our scheme meet the security requirements an auditing protocol needs in Section 5.

Besides, our technology is unnecessary to be assisted by a third party when auditing. It can be adopted to NFC smart phone to enforce wireless payment transaction on the market, because it only needs one subtraction, two hashes, and three point multiplications in the proof verification. Totally speaking, our proposal is competitive in modern cloud data auditing scheme.

8. References

[1] Ateniese, G., Burns, R., Curtmola, R., Herring, J., Khan, O., Kissner, & Song, D. (2011). Remote data checking using provable data possession. ACM Transactions on Information and System Security (TISSEC), 14(1), 12.

[2] Ateniese G, Burns R, Curtmola R, Herring J, Kissner L, Peterson Z, et al., 2007. Provable data possession at untrusted stores. In: Proceedings of the 14th ACM.

[3] Ateniese, G., Di Pietro, R., Mancini, L. V., & Tsudik, G. (2008, September). Scalable and efficient provable data possession. ACM. conference on computer

and communications security, CCS 2007. p. 598–609.

[4] Husain, M. I., Ko, S. Y., Uurtamo, S., Rudra, A., & Sridhar, R. (2014). Bi directional data verification for cloud storage. Journal of Network and Computer Applications, 45, 96-107.

[5] Wang, C., Wang, Q., Ren, K., & Lou, W. (2010, March). Privacy-preserving public auditing for data storage security in cloud computing. In INFOCOM, 2010 proceedings IEEE (pp. 1-9). .

[6] Wang, C., Chow, S. S., Wang, Q., Ren, K., & Lou, W. (2013). Privacy-preserving public auditing for secure cloud storage. Computers, IEEE Transactions on, 62(2),362-375.

[7] Worku, S. G., Xu, C., Zhao, J., & He, X. (2013). Secure and efficient privacy-preserving public auditing scheme for cloud storage. Computers & Electrical Engineering40, (2014), 1703-1713. .

[8] Zhu, Y., Hu, H., Ahn, G. J., & Yau, S. S. (2012). Efficient audit service outsourcing for data integrity in clouds. Journal of Systems and Software, 85(5), 1083-1095

[9] Zhu, Y., Hu, H., Ahn, G. J., & Yu, M. (2012). Cooperative provable data possession for integrity verification in multi-cloud storage. Parallel and Distributed Systems, IEEETransactions on, 23(12), 2231-2244.

[10] Zhu, Y., Ahn, G. J., Hu, H., Yau, S. S., An, H. G., & Hu, C. J. (2013). Dynamic audit services for outsourced storages in clouds. Services Computing, IEEE Transactions on, 6(2), 227-238.

[11] Xu, C., He, X., & Abraha-Weldemariam, D. (2012). Cryptanalysis of Wang's auditing protocol for data storage security in cloud computing. In Information Computing and Applications (pp. 422-428). Springer Berlin Heidelberg.

[12] Erway, C., Küpçü, A., Papamanthou, C., & Tamassia, R. (2009, November). Dynamic provable data possession. In Proceedings of the 16th ACM conference on Computer and communications security (pp. 213-222). ACM.

[13] Di Pietro, R., & Sorniotti, A. (2012, May). Boosting efficiency and security in proof of ownership for deduplication. In Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security (pp. 81-82). ACM.

[14] Bowers KD, Juels A, Oprea A. HAIL: a high-availability and integrity layer for cloud storage. In: Proceedings of the 6th CM conference on computer and communications security, CCS 2009. p. 187–98.

[15] Deswarte Y, Quisquater J-J, Saïdane A. In: Jajodia S, Strous L, editors. Remote integrity checking: integrity and internal control in information systems VI, vol. 140. Boston: Springer; 2004. p. 1–11.

[16] Dodis Y, Vadhan S, Wichs D. Proofs of retrievability via hardness amplification. In:

Reingold O, editor. Theory of cryptography, vol. 5444. Berlin/Heidelberg: Springer; 2009. p. 109–27.

[17] Shacham H, Waters B. Compact proofs of retrievability. In: Pieprzyk J, editor. Advances in cryptology – ASIACRYPT 2008, vol. 5350. Berlin/Heidelberg: Springer; 2008. p. 90–107.

[18] Zheng Q, Xu S. Secure and efficient proof of storage with deduplication. In: Proceedings of the second ACM conference on data and application security and privacy, CODASPY 2012. p. 1–12.

[19] Zheng Q, Xu S. Fair and dynamic proofs of retrievability. In: Proceedings of the first ACM conference on data and application security and privacy, CODASPY 2011. p. 237–48.

[20] Wang Q, Wang, Li J, Ren K, Lou W. Enabling public verifiability and data dynamics for storage security in cloud computing. In: Backes M, Ning P, editors. Computer security – ESORICS 2009, vol. 5789. Berlin/Heidelberg: Springer; 2009. p. 355–70.

[21] Zhuo H, Sheng Z, Nenghai Y. A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability. IEEE Trans Knowl Data Eng 2011;23:1432–7.

[22] Chunxiang X, Xiaohu H, Daniel-Abraha W. Cryptanalysis of Wang's auditing protocol for data storage security in cloud computing. In: Liu C et al., editors. Information computing and applications, vol. 308. Berlin Heidelberg: Springer; 2012. p. 422–8.

[23] Wang Q, Wang C, Ren K, LouW, Li J. Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Trans Parallel Distrib yst 2011;22:847–59

[24] Juels, A., & Kaliski Jr, B. S. (2007, October). PORs: Proofs of retrievability for large files. In Proceedings of the 14th ACM conference on Computer and communications security (pp. 584-597). ACM.

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

[26] Yu, Y., Ni, J., Ren, J., Wu, W., Chen, L., & Xia, Q. (2014). Improvement of a Remote Data Possession Checking Protocol from Algebraic Signatures. In Information Security Practice and Experience (pp. 359-372). Springer International Publishing.

[27] Zhu, Y., Wang, H., Hu, Z., Ahn, G. J., Hu, H., & Yau, S. S. (2011, March). Dynamic audit services for integrity verification of outsourced storages in clouds. In Proceedings of the 2011 ACM Symposium on Applied Computing (pp.

1550-1557). ACM

[28] Yang, K., & Jia, X. (2013). An efficient and secure dynamic auditing protocol for data storage in cloud computing. Parallel and Distributed Systems, IEEE Transactions on, 24(9), 1717-1726., K., & Jia, X. (2012). Data storage auditing service in cloud computing: challenges, methods and opportunities. World Wide Web, 15(4), 409-428.

[29] Yang, K., & Jia, X. (2013). An efficient and secure dynamic auditing protocol for data storage in cloud computing. Parallel and Distributed Systems, IEEE Transactions on, 24(9), 1717-1726.

[30] Hao, Z., Zhong, S., & Yu, N. (2011). A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability. Knowledge and Data Engineering, IEEE transactions on, 23(9), 1432-1437.

[31] Itani, W., Kayssi, A., & Chehab, A. (2009, December). Privacy as a service: Privacy-aware data storage and processing in cloud computing architectures. In Dependable, Autonomic and Secure Computing, 2009. DASC'09. Eighth IEEE International Conference on (pp. 711-716). IEEE.

[32] Kamara, S., & Lauter, K. (2010). Cryptographic cloud storage. In Financial Cryptography and Data Security (pp. 136-149). Springer Berlin Heidelberg.

[33] Yang, J., Wang, H., Wang, J., Tan, C., & Yu, D. (2011). Provable data possession of resource-constrained mobile devices in cloud computing. Journal of networks, 6(7), 1033-1040.

[34] Wang, B., Li, B., & Li, H. (2012, January). Knox: privacy-preserving auditing for shared data with large groups in the cloud. In Applied Cryptography and Network Security (pp. 507-525). Springer Berlin Heidelberg.

[35] D. Johnson and A. Menezes, “The Elliptic Curve Digital Signature

[36] Algorithm (ECDSA),” Centre for Applied Cryptographic Research, University of Waterloo, August 1999.

[37] NIST, Digital Signature Standard (DSS), FIPS 186-3, June 2009,

[38] http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf

[39] NIST, Recommendation for Key Management - Part 1: General(Revised), Special Publication 800-57, March 2007, http://csrc.nist.gov/groups/ST/toolkit/documents/SP800-57Part1_3-8-07.pdf

[40] Liu C, Yang C, Zhang X, Chen J. External integrity verification for outsourced big data in cloud and IoT: A big picture. Future Generation Computer Systems 49, (2015), 58-67.

[41] Zhang J, Dong Q. Efficient ID-based public auditing for the outsourced data in

cloud storage. Information Sciences 343-344, (2016), 1-14.
[42] Yang G, Yu J, Shen W, Su Q, Fu Z, Hao R. Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability. The Journal of Systems and Software 113, (2016), 130-139.
[43] Yu Y, Niu L, Yang G, Mu Y, Susilo W. On the security of auditing mechanisms for secure cloud storage. Future Generation Computer Systems 30, (2014), 127–132.
[44] Yu Y, Au M-H, Susilo W, Ni J, Zhang Y, Vasilakos A-V, Shen J. Cloud Data Integrity Checking with an Identity-based Auditing Mechanism from RSA. Future Generation Computer Systems, (2016), 1-16.
[45] Yu Y, Ni J, Au M-H, Liu H, Wang H, Xu C. Improved security of a dynamic remote data possession checking protocol for cloud storage. Expert Systems with Applications 41, (2014), 7789–7796.
[46] Malina L, Hajny J, Dzurenda P, Zeman V. Privacy-preserving security solution for cloud services. Journal of Applied Research and Technology 13. (2015), 20-31.
[47] Tan, S., Jia, Y., 2015. NaEPASC: a novel and efficient public auditing scheme for cloud data. J. Zhejiang Univ. Sci. C 15 (9), 794–804.
[48] Garg, N., & Bawa, S. 2016. Comparative analysis of cloud data integrity auditing protocols. Journal of Network and Computer Applications, 66, 17-32.
[49] Alizadeh, M., Abolfazli, S., Zamani, M., Baharun, S., & Sakurai, K. (2016). Authentication in mobile cloud computing: A survey. Journal of Network and Computer Applications, 61, 59-80.

7/26-28 日美國行

Las Vegas 學術研討會與旅遊記

1. 入住 mente Carlo 後的隔天整個早上準備下午學術研討會(springer 出版公司舉辦的年度電腦會議）的英文講稿，而且當 SESSION 2-SAM: CRYPTOGRAPHIC TECHNOLOGIES I + ASSESSMENT 的 session chair 要將各演講人的資歷看過講過一遍免得介紹演講人時 舌頭打結

Chair: Dr. Jue-Sam Chou, Nanhua University, Taiwan

July 25, 2016 (Monday); 03:20pm - 06:00pm

(LOCATION: Ballroom 2)

http://worldcomp.org/events/2016/program/sam_25

2. Las Vegas 附近大峽谷一日遊

5:10 在飯店的門口坐上中國人舉辦的大峽谷一日遊 車廂內空氣混濁 沿路景觀草是枯黃的少數地方是綠油油的農作物 很明顯是人灌溉出來的 愈接近大峽谷的地方 一種類似仙人掌的植物愈多 當地人稱它為（摩門教經典中的人名），聽說幾百年前 摩門教先知帶領一群摩門教徒逃難到此地 又飢又渴 就是這種樹救了他們 因為樹幹內含有大量的水 也可以吃 到了大峽谷後 共有三種遊園方式 1.基本款 含入門票 園內車費 中餐 2.基本款外加天空步道 3.第二款外加直升機遊繞大峽谷（聽導遊說大峽谷的長度有 800 公里 直升機只是走一小段）選擇第三方案的人會很趕 天空步道可能就無法去 我們是第二方案 隨著遊園公車（one way)的停靠站 玩出來 第一站門口擺一輛破馬車 構造跟台灣早期鐵輪牛車幾乎相同 第二站是天空步道 進入前 要將所有的東西放入鎖櫃內 以減輕橋的負荷 及蓄意破壞 最重要的是你不能照相 要請他們照 出來時 按張數計費 所有入橋分兩個入口 要照相的排很長 我們沒有耐心 選擇沒照相的入口進入 一位中國凌波來的年輕媽媽走了幾步後 不敢走 又折回去 透過玻璃 看到岸邊無法看到的谷底 是有點嚇人 每一步都戰戰兢兢 深怕玻璃有問題摔下去 快點終點時 剛好太陽很大 可能是橋墩與橋身的材質不同 熱脹係數不同 發出嘎嘎的聲音 出了天空步道後 觀賞印地安人小朋友跳傳統舞蹈 雖有特色卻顯得單調其圖騰很類似中國老者過生日穿的福壽服上的圖騰(相片）純種印地安人小孩的輪廓 臉型寬寬 個子矮矮的 跟蒙古人很像 他們的房子是由木頭及泥土建造而成 外型類似蒙古包 蒙古包旁有一種只由幾根木頭撐在一起 外圍裹上枝葉 雖是簡陋 但相較於洛杉磯市中心的 homeless 在路邊搭的帳篷 沒有汽油味 沒有噪音 通風涼爽又有木頭散發出來的香味 現代經濟體制資本主義社會貧富差距越來越大 自然資源愈來愈被少數人所壟斷 導致多數人無法生存下去 政府若沒有透過政策弭平 社會會愈來愈不穩定 最終可能導致資本主義的奔潰

3. 感想

還是生活在台灣比世界各國都還方便，風景秀麗食衣住行育樂樣樣齊備便利,不過對於英文還是要勤加練習常用英語與人對話,溝通才不至於產生問題

科技部補助計畫衍生研發成果推廣資料表

日期:2017/04/24

科技部補助計畫	計畫名稱: 以ECC為基礎的雲端檔案完整保全之綜合性策略研究
	計畫主持人: 周志賢
	計畫編號: 104-2221-E-343-002- 學門領域: 資訊安全實務應用專案

無研發成果推廣資料

104年度專題研究計畫成果彙整表

計畫主持人：周志賢	**計畫編號：**104-2221-E-343-002-
計畫名稱：以ECC為基礎的雲端檔案完整保全之綜合性策略研究	

成果項目					量化	單位	質化 （說明：各成果項目請附佐證資料或細項說明，如期刊名稱、年份、卷期、起訖頁數、證號...等）
國內	學術性論文	期刊論文			0	篇	
		研討會論文			0		
		專書			0	本	
		專書論文			0	章	
		技術報告			0	篇	
		其他			0	篇	
	智慧財產權及成果	專利權	發明專利	申請中	0	件	
				已獲得	0		
			新型/設計專利		0		
		商標權			0		
		營業秘密			0		
		積體電路電路布局權			0		
		著作權			0		
		品種權			0		
		其他			0		
	技術移轉	件數			0	件	
		收入			0	千元	
國外	學術性論文	期刊論文			0	篇	
		研討會論文			1		Proceedings of the 2016 international conference on security & management, PP61-67, ISBN:1-60132-445-6, CSREA Press
		專書			0	本	
		專書論文			0	章	
		技術報告			0	篇	
		其他			0	篇	
	智慧財產權及成果	專利權	發明專利	申請中	0	件	
				已獲得	0		
			新型/設計專利		0		
		商標權			0		
		營業秘密			0		
		積體電路電路布局權			0		

		著作權	0		
		品種權	0		
		其他	0		
	技術移轉	件數	0	件	
		收入	0	千元	
參與計畫人力	本國籍	大專生	1	人次	兼任助理
		碩士生	2		兼任研究助理
		博士生	0		
		博士後研究員	0		
		專任助理	0		
	非本國籍	大專生	0		
		碩士生	0		
		博士生	0		
		博士後研究員	0		
		專任助理	0		
其他成果 （無法以量化表達之成果如辦理學術活動、獲得獎項、重要國際合作、研究成果國際影響力及其他協助產業技術發展之具體效益事項等，請以文字敘述填列。）			擔任 SAM'16 Session:Cryptographic Technologhies I 之 session Chair		

科技部補助專題研究計畫成果自評表

請就研究內容與原計畫相符程度、達成預期目標情況、研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）、是否適合在學術期刊發表或申請專利、主要發現（簡要敘述成果是否具有政策應用參考價值及具影響公共利益之重大發現）或其他有關價值等，作一綜合評估。

1. 請就研究內容與原計畫相符程度、達成預期目標情況作一綜合評估
   ■達成目標
   □未達成目標（請說明，以100字為限）
   　□實驗失敗
   　□因故實驗中斷
   　□其他原因
   說明：

2. 研究成果在學術期刊發表或申請專利等情形（請於其他欄註明專利及技轉之證號、合約、申請及洽談等詳細資訊）
   論文：□已發表　■未發表之文稿　□撰寫中　□無
   專利：□已獲得　□申請中　■無
   技轉：□已技轉　□洽談中　■無
   其他：（以200字為限）
   已投稿至 SAM'16 國際學術研討會, 正修改投稿學術期刊中

3. 請依學術成就、技術創新、社會影響等方面，評估研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性，以500字為限）
   隨著雲端儲存體服務日益進步，且經由雲端資料的稽核方案可以讓使用者有效地稽核他們儲存在雲端伺服器裡面的資料有無完整性，以確保雲端伺服器是誠實的，因此客戶才可以完全確定放在雲端伺服器裡的資料沒有被任意竄改或更動，另外資料擁有者也能夠確保資料上傳到雲端伺服器時有進行資料的加密，因此公開稽核的服務方案是非常重要的。在本文中，我們提出一個有效率的公開稽核方案，主要是為了證明出惡意雲端伺服器不能偽造出不存在的消息塊，而且能讓客戶可以完全地相信雲端伺服器裡面的資料是安全的，另外我們還有做出證明計算大量資料的正確完整性，最後我們有實現證明我們的協議是非常有效率與安全的。

4. 主要發現
   本研究具有政策應用參考價值：□否　■是，建議提供機關電算中心提供給客戶來稽核客戶本身儲存在雲端的資料
   （勾選「是」者，請列舉建議可提供施政參考之業務主管機關）
   本研究具影響公共利益之重大發現：□否　■是
   說明：（以150字為限）
   能讓客戶可以完全地相信雲端伺服器裡面的資料塊是完整的，另外我們還有

做出證明計算一個完整檔案大量資料的正確完整性，最後我們有實現證明我們的協
議是非常有效率與安全的。