Abstract—In this paper, we focus on the security of outsourced database. A verification scheme for outsourced database is proposed based on the verifiable polynomial technique. In this scheme, we consider the encrypted database. The outsourced high degree polynomial will enhance the authenticity of the data. If the cloud server returns fake data, it will be detected by the clients easily.

Keywords—Verifiable, Polynomial, Outsourced Database

1 Introduction

With the rapid development of Internet and network technology, more and more data will be used. The fast growth of data brings much trouble to people. Cloud computing is powerful, it helps us to solve the big data problem. Cloud computing is the development and application of distributed computing, parallel computing and grid computing. With the help of the powerful cloud server, the client can easily accomplish the complex work easily.

Outsource service allows resource constrained clients to outsource the complex tasks to the powerful cloud server with the manner of pay-per-use. In order to save the own storage space, people will outsource the huge database to a remote cloud server. With the development of cloud computing, outsourcing is becoming an important part in modern business. By utilizing outsourcing, clients can concentrate on their own work and operate their business applications via the Internet, rather than maintaining the substantial hardware, software, and applications.

However, the powerful cloud server is untrusted. When data is outsourced, clients loose the controllability of the outsourced data. It is a hard work to verify the authenticity of the data. The untrusted cloud server may tamper the data. When clients query the data, they will get the fake data. Thus, it is important to study the verifiable outsourced database.

Our Contributions. This paper focus on the correctness of the outsourced data. A new and simple scheme to verify the encrypted data is proposed. In the scheme, the secure outsourcing computation of high degree polynomials is used to help the clients accomplish the verification. Clients can verify the required parts of the outsourced database by checking the proof information. The outsourced data is encrypted, and it will not be revealed.

1.1 Related Work

In 1980s, Ben-Or et al. proposed a outsource computation scheme with an honest-but-curious oracle [3, 4]. Then some outsourcing computation
schemes come out [7, 11]. In 2002, Atallah, Panta-
zoomopoulos and Rice [1] proposed secure outsourcing
scheme for scientific computing and numerical cal-
culations. However, the verification phase was not
considered. In 2008 Benjamin et al. [6] proposed a
verifiable outsourcing computation scheme for lin-
ear algebraic calculation. And In 2010, Gentry et
al. [10] expanded verifiable outsourcing computa-
tion to arbitrary function F. However, the efficien-
cy is low. In 2016, Ye et al.[15] proposed a ver-
ifiable delegation scheme for polynomials, which
improved the efficiency.

There are a lot of work on the verification of
outsourced database, such as, [8, 2]. The homo-
 morphic encryption was used in some schemes,
however, this reduced the computation efficiency.
Then some schemes without homomorphic encryp-
tion comes out. Some schemes are based on based
on Message Authentication Code [5, 9], and some
are based on Merkle hash trees [13, 12]. An in-
dex tree for the database is generated by using hash
functions, by which the authenticity auditing can
be achieved. However, lots of information for ver-
ification has to be stored. In 2009, Pang et al. [14]
proposed an outsourced database scheme based on
the signature chaining technique, in which the com-
putation can be used for verification. In 2013 Cata-
lano and Fiore [9] used the vector commitment to
generate a verifiable database with efficient update.

1.2 Organization

The organization of this paper is as follows. Some
preliminaries are given in Section 2. The proposed
scheme are given in Section 3. Finally, the conclu-
sion is made in Section 4.

2 Preliminaries

2.1 Hash Function

A hash function can take an arbitrary input and out-
put a fixed-size string, which satisfies the following
properties.

- It is easy to compute the hash value for any
given input.
- It is infeasible to compute an input such that
the hash value equals a given hash value.
- It is infeasible to find two different inputs that
can get the same hash value.

2.2 Verifiable Outsourced Database

The outsourced database is an example of client-
server model. In the outsourced database model,
the service provider is powerful, who has the in-
frastucture for outsourced databases, and can pro-
vide efficient data processing, such as, store, update
and query the database.

Verifiable outsourced database allows clients to
authenticate database operations. The clients can
query the outsourced database, when get the re-
turned results, the clients can verify the correctness
of the outsourced data with the help of the proof.

3 The Proposed Scheme

The are three parts in the system, client, cloud serv-
er 1 and cloud server 2.

Initialization. The client generates a high de-
gree polynomial, f(x).

\[ f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_d x^d \]

where, \( a_i \in \mathbb{Z}_p \), \( i = 0, 1, \cdots, d \), \( P \) is a large prime,
and \( g \) is a generator.

Then, client selects a prime \( q \), such that \( |q| = |p| \),
and computes \( n = pq \). Client chooses \( q_i \in \mathbb{Z}_n \), and
computes

\[ r_i = (1 + n)^{a_i} q_i^n \mod n^2 \]

where, \( i = 0, 1, \cdots, d \).

The polynomial is transformed as

\[ F(x) = r_0 + r_1 x + r_2 x^2 + \cdots + r_d x^d \]

Client randomly selects \( a, k_0, k_1 \in \mathbb{Z}_p \), and com-
putes

\[ t_i = g^{k_0 k_1} g^{a r_i} \]

where, \( i = 0, 1, \cdots, d \).

At last client sends

\( (r_0, r_1, \cdots, r_d) \)
and 
\[(t_0, t_1, \cdots, t_d)\]
to cloud server 1.

**Ciphertext Generation.** The client selects an encryption algorithm \(En(\cdot)\) and encrypts the original data \(x\) as \(\sigma_x = En(x)\), and sends it to the cloud server 1.

When receiving the data \(\sigma_x\), the cloud server 1 computes 
\[\sigma_y = F(\sigma_x) = \prod_{i=0}^{d} r_i^{\sigma_i^t}\]
and
\[T = \prod_{i=0}^{d} t_i^{\sigma_i^t}.\]

Then, cloud server 1 sends \(\sigma_y\) and \(T\) to the client. 

Client computes 
\[Z = \prod_{i=0}^{d} (g^{k_0k_i})^{\sigma_i^t},\]
and 
\[y = \frac{\Phi(n) - 1}{n} \Phi^{-1}(n).\]

Then, client verifies whether the following equation holds 
\[T = Z \cdot g^{ay}.\]
If not, client outputs \(\perp\). Otherwise, client computes 
\[proof_i = H(i||\sigma_y||y)\]
where, \(H(\cdot)\) is a non-collision hash function. 

At last, client sends 
\[(i, \sigma_y, proof)\]
to cloud server 2.

**Query.** The client queries for the data in the position \(i\). Then the cloud server returns the ciphertext 
\[C_i = (i, \sigma_y, H(i||\sigma_y||y_i))\]
to the client.

**Verify.** Client sends \(\sigma_x\) to the cloud server 1. Cloud server 1 computes 
\[\sigma_{y_i} = F(\sigma_{x_i}) = \prod_{i=0}^{d} r_i^{\sigma_i^t}\]
and 
\[T_i = \prod_{i=0}^{d} t_i^{\sigma_i^t}.\]

Then, cloud server 1 sends \(\sigma_{y_i}\) and \(T_i\) to the client. 

Client computes 
\[Z_i = \prod_{i=0}^{d} (g^{k_0k_i})^{\sigma_i^t}\]
\[= g^{k_0^n - t_i^{\sigma_i^t}},\]
and 
\[y_i = \frac{(\sigma_{y_i})^n - 1}{n} \Phi^{-1}(n).\]

And then verifies whether the following equation holds 
\[T_i = Z_i g^{ay_i}.\]
If not, client outputs \(\perp\). Otherwise, client computes 
\[proof_i = H(i||\sigma_y||y_i)\]
and verifies 
\[proof_i \overset{?}{=} proof.\]
If the equation holds, the data \(\sigma_i\) is correct, and client can get the original data \(x = Dec(\sigma_i)\). Otherwise, client outputs \(\perp\).

### 4 Conclusion

The rapid increasing of data brings much trouble to people. More storage space is needed. Cloud computing gathers a lot of resource together, and provides huge storage space for users. In order to save the local resource, clients often outsource their database to the remote cloud server. And for the security, the outsourced data should be encrypted. As large amounts of data is outsourced to the cloud server, the users have to keep a little information for verification. In this paper, a new algorithm for outsourced database verification is proposed. The actual data is encrypted, and it will not be revealed. The computation results can be easily verified by the client. The clients can easily check the authenticity of outsourced data.
ACKNOWLEDGMENT

This work was supported by Guangxi Key Laboratory of Cryptography and Information Security (No. GCIS201607); the Fund of Lab of Security Insurance of Cyberspace, Sichuan Province (szjj2016-091); the Talent Project of Sichuan University of Science & Engineering (2017RCL23).

References