Journal of Ambient Intelligence and Humanized Computing (2023) 14:7983–8002
https://doi.org/10.1007/s12652-023-04608-7

ORIGINAL RESEARCH

An improved lattice based certificateless data integrity verification techniques for cloud computing

Dilli Babu Salvakkam[1,2] · Rajendra Pamula[1]

Received: 22 March 2021 / Accepted: 4 April 2023 / Published online: 26 April 2023

Abstract
Today, cloud computing security schemes face many challenges, risks, and pitfalls. Digital signatures are vital security measures that must not be overlooked in present advancement. As a result of the difficulty involved in verifying and enforcing digital signatures, classic public-key approaches are vulnerable to security threats. Numerous assaults on cloud-based vital services such as search engines, storage systems, and crucial applications ranging from healthcare to defense have recently been reported in the literature due to advances in quantum computing. A collection of post-quantum cryptography primitives is being created to assist users in deciding which to utilize to overcome the present migration. This paper reviews the work done by Yongqiang Zhang et al. and Yang et al. Crypto Scheme. Using our technique, users may generate a Lattice-based public key without offering a Certificate-based method for Data Integrity Verification, as proposed in this study. It has the additional benefit of lowering the cost of the signature algorithm. This paper presents various Post-Quantum security attacks based on the AVISPA tool and Vulnerability attacks of existing security. We demonstrate that our technique is probably safe for usage in the random oracle model using this way. As an additional benefit, it is more secure than past alternatives. After demonstrating the effectiveness of our approach in the random oracle model, we tested its safety by examining the recent techniques. The various cryptographic protocols from literature surveys do not entirely cover the complete security attacks. This paper cryptanalyzes these cryptographic protocols with Cloud data integrity verification techniques and also provides a Proposed Lattice based Certificateless Data Integrity Verification Model for Cloud Computing (CDIVM), using AVISPA (Automated Validation of Internet Security Protocol and Applications Tool) and BAN logic.

Keywords Cloud computing · Data integrity · Cryptanalysis · Lattice based crypto system · Cloud storage · Privacy preserving · Attribute-based encryption · Identity-based cryptography

1 Introduction

In recent year's applications and data in cloud raise to big volumes, so providing data trust, preserving privacy for data and identifying the various cyber-attacks become significant research issue. According to Sen in 2015 there will be a maximum of 20 to 50 billion devices connected to cloud servers uses various cloud computing platforms by 2025 (Sen et al. 2015). If any security failure of the computing device will not be tolerable, this affects billions of Internet users to face vulnerable to privacy and even the lives. Providing advanced level of novel security is inevitable, although the current solutions still imperfect and inefficient to provide security from various attacks for the larger volumes of data (Zissis et al. 2012). The development of novel security strategies still becomes lag for cloud. The available classical and modern cryptographic recipes do not transfer easily to cloud settings. Here this paper tries to improve the various pitfalls in cryptographic recipes by analysing cryptanalysis using BAN logic and AVISPA tool. Some of the security aspects in cloud are organized in four topics: privacy preserving, data integrity, secure storage and access control mechanisms (Zhang et al. 2015).

As the first step we analyzed the security or lapses in privacy preserving in cloud, data integrity and access control mechanisms in cloud. For privacy preserving, the third party auditor (TPA) techniques are analyzed and enhanced for

✉ Dilli Babu Salvakkam
dillibabusalvakkam@gmail.com

Rajendra Pamula
rajendra@iitism.ac.in

[1] Department of Computer Science and Engineering, Indian Institute of Technology (ISM), Dhanbad, Jharkhand, India

[2] Department of Computer Science and Engineering, Malla Reddy University, Hyderabad, India

better crucial data auditing for cloud data. In data integrity service level agreements (SLA's), integrity checking auditing protocols is simulated for safe storage in cloud. In order to protect from vulnerability of access restrictions such as Distributed Daniel of Service (DDoS) attack are identified and enhanced the protocol such as attribute based encryption techniques, bilinear pairing, order preserving encryption, hash authentication, discrete logarithm problem, message authentication codes, BLS protocol and RSA protocol (Boldyreva et al. 2011).

Many people use cloud technology because cloud computing has become so popular. Some cloud storage services let users store their data for free. Still, these services can lose their stored data for some reasons, such as problems with the hardware or a need to improve performance. A data audit is an important service that makes sure that the data that has been stored is safe. As part of a primary data audit, the user should make sure that the data that has been saved is secure. This operation successfully uses different technologies, such as cryptography with symmetric keys. The third-party uses techniques, such as a signature scheme, to confirm that the data is not modified. A signature scheme is more complicated to use than an integrity check, but it has more benefits. Most of the time, users would rather not know that their signature is linked to the information they gave (Zhang et al. 2015).

After describing the pitfalls in privacy preserving, integrity and access control for cloud storage data, these techniques RSA, DES, 3-DES, AES, Blowfish, ECC, Elgamal, Diffie–Hellman, etc., are cryptanalysis using BAN logic, AVISPA and MIRCL tools (Team 2006).

1.1 Review of Yongqiang Zhang et al. and Yang et al. Crypto Scheme

Yongqiang Zhang et al. proposed a designated verifier signature (DVS) scheme with a simulation algorithm. When the signer chooses a verifier, that verifier can simulate the user signature. Even if a DVS has not been done on a message, the verifier can still figure out who signed it. But because it can do that, it can not tell the public anything about the person who signed the document.

A cloud service provider can do a data audit by using a primitive in a data storage audit service. This method can only be used by someone designated as a verifier. Quantum computing could make traditional security measures like the DVS constructs useless because they depend on how hard some number-theory issues are to solve. In 1997, Shor reported a paper showing how a quantum computer could solve discrete logarithms and integer factoring.

Yang et al., used a quantum computer to show in a paper in 2021 how to factor the number 15. Xu et al., said they had broken the record by factoring in 143 in 2012. Even though 143 is a low number, it is essential to work on making new safety measures to protect against the threat that quantum computers pose.

There is a threat, and one of them is to make safe cryptographic algorithms based on quantum physics. For example, Chuang and Gottesman suggested in 2001 that a digital signature be based on a quantum process. The authors said that a signer needs to pay more attention to how public keys are shared. Doing so could lead to sharing confidential information.

The distribution of public keys is one of the most challenging parts of making secure algorithms. Because of this, it is essential to work on coming up with solutions that can stand up to an attack from a quantum computer. Lattice puzzles are complex problems used to see how well a quantum computer works.

The main contribution of this work is creating a DVS scheme based on Lyubashevsky's signature. To do this, a 1994 study by Cramer et al., described an OR-proof method. Using this technology makes it possible to create many different kinds of DVS.

The method in a system based on a lattice is inefficient. For example, Lyubashevsky created a hashing function in 2012 as H: 0, 1*v:v −1, 0, 1 k. In the OR-proof method, a hashing function can make a random, signed vector.

The range of the hashing function for c 1 will likely fit within the scope of its satisfying role. Nevertheless, the proportion of the process that falls inside the domain of these vectors is a relatively tiny part of the total number of potential vectors. A set of parameters like k = 80 and = 28 to ensure that the vector is re-signed many times.

It eliminates the constraint of v 1 1. It defines the hashing function as H:0, 1*− 1, 0, 1 k to bring the value of k down to a more manageable level. Because of this improvement, updated parameter settings are used to implement the DVS scheme.

They also give a lot of proof that the DVS scheme can not be given to someone else and can not be copied. In that case, the strategy is not used, and the characteristics listed in the third column of the table will be the best way to put the scheme into action. This parameter set is used differently to make the system harder to fake (Salvakkam et al. 2022a; b).

The authors compared the size of the DVS signature that was assumed by Wang et al. They provide a data audit service structured according to the DVS scheme. Using this technology protects the right to privacy of the data owner by stopping unauthorized people from getting to their information.

As a result of the existence of lattice problems, it is generally thought to be better to design safe cryptographic systems based on lattice problems. The reason for this is that lattice problem are a type of problem. When making a lattice-based signature, there are two main ways to go about it. One is the hash-and-sign method, written in a paper by Gentry et al. in 2008. In 2012, Wang et al., developed a strong DVS idea based on the hash-and-sign algorithm. In this article, Yongqiang Zhang et al., talk about a variation of Lyubashevsky's 2012a-based signature scheme.

1.2 Motivation for certificateless data integrity verification model (CDIVM)

Using an OR-proof method, Yongqiang Zhang et al. provide a powerful variational approach based on Lyubashevsky's signature. About 37,212 bits is the minor signature that can be made using acceptable iterations. They used the method Ducas et al. developed in 2013 to improve the instantiation process. This scheme is not based on a certificateless approach. CDIVM is a Certificateless signature scheme, one of the most feasible solutions for providing data integrity and identity authentication in the Cloud Computing Environment. It does not need the time-consuming certificate administration and key escrow processes that are otherwise required. At the moment, numerous certificateless signature systems have been proposed. Still, only a handful are safe and acceptable for use in the Cloud Computing Environment with Quantum Secure.

1.3 Paper flow

The remaining part of the paper is structured as follows. In Introduction section, it is addressed the review of Yongqiang Zhang et al. Crypto Scheme and different cryptographic techniques used in cloud storage, whose security pitfalls, improved version of protocol, and further studies required are proposed in each of the sections. The familiar cryptanalysis tool AVISPA and is used simulation for formal verification of all protocols described in preliminaries and then the data integrity verification CDIVM finally proposed. In conclusion, we wrap up the paper with overall remarks and further research directions.

2 Preliminaries

This section discusses the main theme of bilinear pairing, order preserving encryption, hash authentication, discrete logarithm problem, message authentication, etc., and the computational problems required for understanding, to overcome the pitfalls in existing studies and to follow the improved propositions as discussed below (Amin et al. 2015).

To crypt-analyze various cloud security protocols it is assumed some valid assumptions from a range of security protocols. In order to evaluate various threat models in cloud computing it is assumed that the user is honest, but the authentication servers may face different kinds of attacks speculation over data compromise (Amin et al. 2018).

2.1 BAN logic

BAN Logic used to crypt analyse security and privacy protocols, which is well-defined in mathematical model for security logics proposed by Burrows, Abadi and Needham. The BAN logic can help cloud (Internet) clients to validate the protocol what is practically to be believed. It is a set of propositions for analyzing data exchange protocol over the Internet. It facilitates better way to check whether swapping of data is secure and safe against vulnerable eavesdropper attacks. The BAN logic has three steps to verify the protocol. The first step is to make hypothesis and then prepare symbolic notations. In Sect. 2.3 brief discussion of AVISPA tool which uses the BAN logic for logical constructs for various cryptanalysis. The Euations-1 shows few logic constructs for developing the code. The second step is to verify goals by creating secured communication paths between cloud users. In the last step is to verify the performed or experimented hypothesis to gain the goal. The crypt-analyser prepares hypothesis which are taken from first step. To experiment these BAN propositions will get hypothesis to achieve the goals. For instance, if "A believes, that only the objects A and B know the key K", then user A object receives the encrypted message with the key K. Now you have to believe that the message sent from the user B. Here a new consequence or hypothesis is achieved from deferent assumptions. There are deferent logical symbols and notation available in BAN logic to differentiate between several entities. These rules are defined by the following formulae.

$$
\begin{gathered}
P|\equiv \mathrm{M} : \text{It denotes that the object P believes the M message object(i.e. if X is true)}\\
\mathrm{P} < \mathrm{M}\text{:It denotes that the object P has received the M message.}\\
\mathrm{P} \sim \mathrm{M} : \text{It denotes that the object P once said M message which is sent by P object.}\\
\mathrm{P} \Rightarrow \mathrm{M} : \text{It denotes that jurdication of object P over M object(P has authority on M).}\\
\#(\mathrm{M}) : \text{It denotes that the M is new message.}\\
\mathrm{P} \overset{\mathrm{K}}{\longleftrightarrow} \mathrm{Q} : \text{It defines the P and Q shared the common key K}\\
\overset{\mathrm{K}}{\longrightarrow} \mathrm{P} : \text{It defines the public key K shared to P object.}\\
\{\mathrm{M}\}_{\mathrm{k}} : \text{It denotes the M message encrypted by the key K.}
\end{gathered}
\tag{1}
$$

These notations definitions are used to build AVISPA logic programs. Here the rule has the upper-part specifies the conditions and the bottom-part specifies the result of the conditions. For instance, in shared key logic the BAN rule is described as.

$$\frac{P|\equiv Q \overset{K}{\longleftrightarrow} P, Q \triangleleft M_K}{P|\equiv Q \sim M}. \tag{2}$$

The Eq. 2 depicts that if object P believes that among objects P and Qshared the key K, and the object P receives a message encyphered by the key K, then P may be confident in which the message is sent by the object Q.

$$\frac{P|\equiv \#(M)}{P|\equiv \#(M,N)} \tag{3}$$

This assumption is true that if P object believes that any portion is sent just now, so that P would believe that all messages are recently sent.m 'M'

$$\frac{P|\equiv \#(M), P|\equiv Q \sim M}{P|\equiv Q|\equiv M} \tag{4}$$

In order to verify the hypothesis to prove that "the object P believes so that the object Q believes". If object P believes M message which has recently sent the object P believes that the message M was sent by object Q, then object P may believe that the message for the object Q. For third party verification the proposition will be as Eq. 5.

$$\frac{P|\equiv M|\equiv (m,n)\, and\, P|\equiv M \Rightarrow (m)}{A|\equiv S} \tag{5}$$

Here, P believes that message M has an authority over the message in some portion then object P is ready to believe M. Certainly the BAN logic has been verified over many security techniques proposed by the Needham-Schroeder, KERBOROS and CCITT X.509. The BAN identifies the flaws in most of these protocols such as Spinellis, Gritzalis and Georgiadis in 1999. Hence we strongly believed to adopt to verify cryptanalysis for various cloud privacy security protocols to evaluate protocol and propose pitfalls.

2.2 Brief discussion of AVISPA simulation.

The AVISPA tool is utilized for testing the pitfalls of the various cloud computing security and privacy protocols by using HLSPL language. This tool tests the security levels of passive attacks as well as active attacks confine with the man-in-middle server attacks, Deniel of Service (DOS) attack and replay attacks by impersonation to the server etc. This tool cryptanalyzes the security in the BAN logic. Thia tool also verifies the different assumed security protocol for cloud environment is whether SAFE against certain types of attacks or UNSAFE.

The AVISPA tool is renown technique and broadly used in cloud internet security protocols analysis for formal security verification. This tool analyses whether protocol is SAFE under various types of attacks or UNSAFE for the cloud security protocol and its security levels of protocol using OFMC, CL-ATSE, etc. The BAN logic formed based on the language called High Level Protocol Specification Language (HLPSL). The AVISPA tool structure is presented in Fig. 1 which depicts the flow of tool and verification of the HLPSL code using four different back ends, one is (On the fly Model Checker (OFMC) in which symbolic techniques are countable for this server code to explore the protocol state space. The Constraint-Logic-based Attack Searcher is a back end CL-AtSe translates from any security protocol details into intermediate format file.

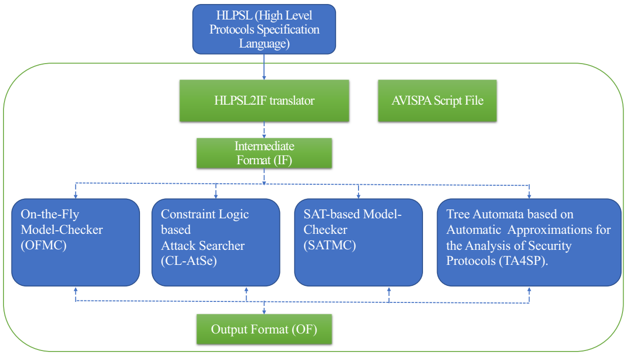

HLPSL (High Level Protocols Specification Language)
HLPSL2IF translator
AVISPA Script File
Intermediate Format (IF)
On-the-Fly Model-Checker (OFMC)
Constraint Logic based Attack Searcher (CL-AtSe)
SAT-based Model-Checker (SATMC)
Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP).
Output Format (OF)

Fig. 1 The *AVISPA* tool Architecture

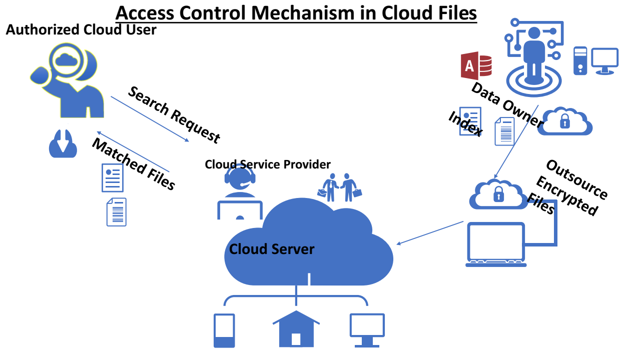


Fig. 2 Access Control mechanism in encrypted cloud

2.3 Cryptanalysis of various security and privacy techniques in cloud computing

Data encryption is a well-established conventional technique for defending important data. If the data enciphered once, then that data might be very typical to access (queried) to match certain query-based match retrieval. We analyse various order-preserving encryption schemes for queering outsourced cloud data. Here the standard cloud encrypted index tables are generated for querying the databases as represented in the Fig. 2 by data owner.

In this type of encryption method, even though the attacker can obtain right to use the encrypted database, who practically cannot known the encrypted index table values so that the attacker can not decrypt or encrypt arbitrary values of his choice. Hence the access can be restricted. The following two schemes Order Preserving Encryption (OPE) and (Attribute Based Encryption) ABE schemes provides better access control for stored data indexing.

The encryption protocols are run over the insecure channel, the intruder may use various privileges during execution of stored encrypted files like normal user. Here we considered few widely accepted assumptions.

i. An attacker get access to server can extract stored encryption file.
ii. An attacker can guess low entropy passwords to decrypt all possible keys to open the stored search encryption files.
iii. The attacker may be a lawful person to access the files or vice versa.

2.4 Order Preserving Encryption (OPE) scheme

This section presents brief reviews for order preserving encryption in cloud storage outsourced data, which executes several phases such as introduction, algorithm description, limitations, and cryptanalysis of the protocol and research issues.

In the Order Preserving Encryption, Rakesh Agarwal et al. Protocol. Presents the encipherment scheme which allows comparison parameter such as MAX, MIN, and COUNT, etc., encrypt the data without doing decryption operation. Also, the visualizing queries such as ORDER BY and GROUP BY operations may be implied to the data. But, when applying AVG or SUM the data values need to be decrypted. In Cao and Cong Wang et al. present ranked keyword finding technique to encipher the outsourcing files in the cloud server. This assumption assesses to clear demands of huge data and user files with efficient, secure search using ranking the key word over encrypted cloud data. The typical OPE model of cloud security attack is described by the characters Alice and Bob as shown in the Fig. 3.

Usually in OPE the comparison operator is used in the numeric columns, the attacker may caught the access to the enciphered database, although not having prior indexed file data, so that intruder cannot have different key values in order to encipher or decipher of his own. Here we considered the OPE of One-to-many mapping with the Setup and Retrieval phases.

The data user calls the method KEY GEN to generate random keys and outputs K. The data user calls the method BUILDINDEX to build the inverted index (K, C) to encrypt scores with index of collection C. The legal user generates the trapdoor word T_w and is sent to search and match the file at the cloud server then uses $f_y^d(w)$ to decrypt the files. The cloud server now identifies the file and associated order preserved encrypted files. The server brings the files and sends to user with chronological order of encrypted relevance scores. In order to address the rage queries, the issue leads to encryption techniques which are not preserves the order and B-tree index database, etc. The well-organized relevance score search dynamics require to authorize the results, and One-to-Many, many-to-one technique in order to preserve the mapping needs.

The cryptanalysis of an OPE scheme by the help of AVISPA tool for access control mechanism for testing pitfalls in OPE protocol. It is assumed that the role for authorized user as Alice and the data owner as Bob. The

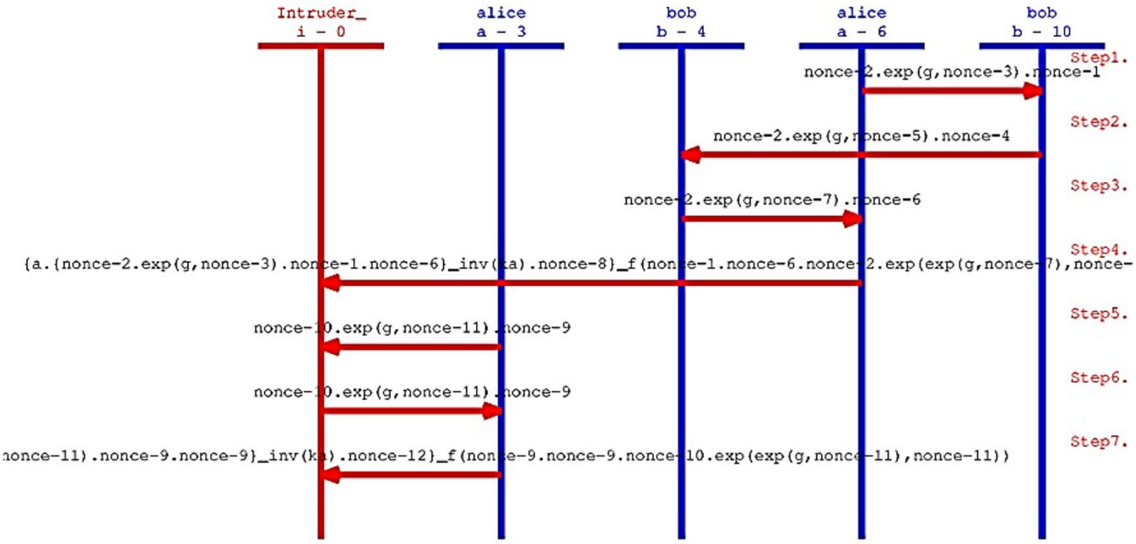

Intruder_
i - 0
alice
a - 3
bob
b - 4
alice
a - 6
bob
b - 10
Step1.
nonce-2.exp(g,nonce-3).nonce-1
Step2.
nonce-2.exp(g,nonce-5).nonce-4
Step3.
nonce-2.exp(g,nonce-7).nonce-6
Step4.
{a.{nonce-2.exp(g,nonce-3).nonce-1.nonce-6}_inv(ka).nonce-8}_f(nonce-1.nonce-6.nonce-2.exp(exp(g,nonce-7),nonce-
Step5.
nonce-10.exp(g,nonce-11).nonce-9
Step6.
nonce-10.exp(g,nonce-11).nonce-9
Step7.
nonce-11).nonce-9.nonce-9}_inv(ka).nonce-12}_f(nonce-9.nonce-9.nonce-10.exp(exp(g,nonce-11),nonce-11))

Fig. 3 An intruder attack model over encrypted data during unsecured channel

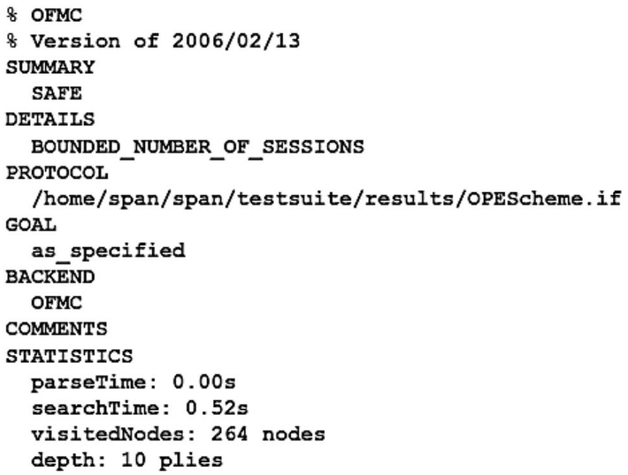
result obtained in Figs. 4 and 5 are based on OPE scheme with respect to widely-accepted cryptanalysis backend codes OFMC and CL-AtSe, based on the tool SPAN-AVISPA. Since it is to conclude OPE method is SAFE in OFMC back end and CL-AtSe code backends under BAN logic. The variable length strings study in OPE is the further research required for data files encryption in clouds, also the key index maintenance for fast encryption, the query plans and query optimization in clouds yet to be addressed.

2.5 Attribute based encryption scheme

The access control mechanism in cloud environment with help of cryptographic primitives like attribute-based encryption scheme (ABE) is major research issue for multi level security. This section describes brief reviews of attribute-based encryption in cloud storage outsourced data, which executes several phases such as introduction, algorithm description, limitations, and cryptanalysis of the protocol and research issues as described in Fig. 6.

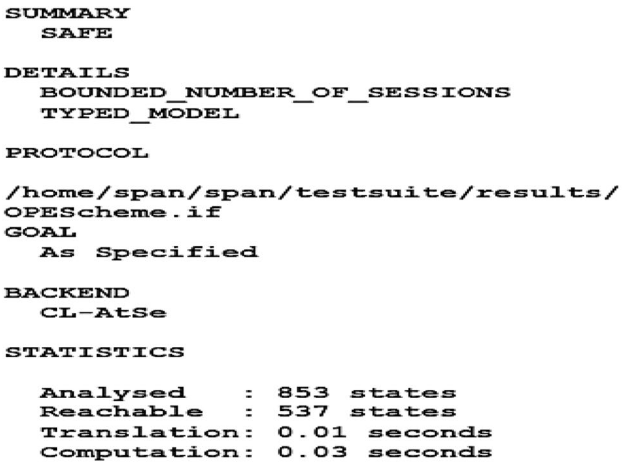
An attribute-based encryption system works based on the sender and receiver's private keys, also cipher text related to policy or set of user attributes. Here, the encrypted data is not necessarily done by using one particular user public key. In Xu et al. (2015) protocol presents the encryption technique called ABES "Attribute based Encryption Scheme", which uses AND, OR gates to access the encrypted data like OPES but, in APES it takes cipher text labels as attributes for group of users and private keys are used to access the structures.

In the study of cryptanalysis for OPE scheme the following security goals are verified using OFMC.



```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/OPEScheme.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.52s
  visitedNodes: 264 nodes
  depth: 10 plies
```

Fig. 4 AVISPA output of OPE scheme in the OFMC code in cloud



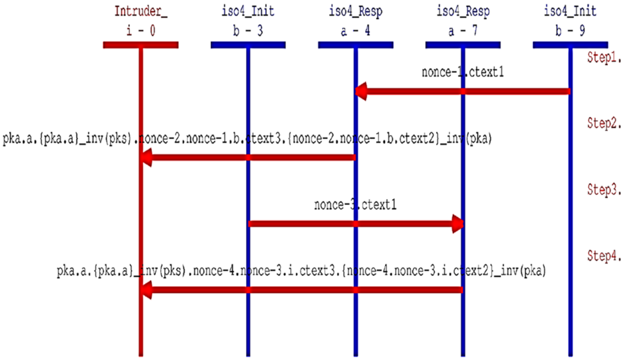
```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL

/home/span/span/testsuite/results/
OPEScheme.if
GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed    : 853 states
  Reachable   : 537 states
  Translation: 0.01 seconds
  Computation: 0.03 seconds
```

Fig. 5 AVISPA output of OPE in CL-AtSe code in cloud

2.5.1 Cryptanalysis *of the protocol*

The ABES which allows the stored data in cloud with minimum amount of information laws even though the server is compromised. If the intruder gets access to the encrypted trusted database server then confidentiality and integrity of the data can also compromise but in ABE scheme which has complex access control to restrict the partial data loss on the cloud storage.

In encryption the algorithm uses the sender and receivers public key to encrypt message by doing XOR with mapping. The encryption takes as system public parameters to access structure and as universal attribute. The encrypted cipher text only accessible if the access structure a possessed by the users. The decryption takes PK public parameter and multiple of receiver's secret key to decrypt cipher text C with PK's hash value. The receiver takes all this information to decipher all the encrypted data. Another decryption step takes as input as public keys and cipher text with access policy AP and secret key for S attributes. In order to address



Intruder_
i - 0
iso4_Init
b - 3
iso4_Resp
a - 4
iso4_Resp
a - 7
iso4_Init
b - 9
Step1.
nonce-1.ctext1
Step2.
pka.a.{pka.a}_inv(pks).nonce-2.nonce-1.b.ctext3.{nonce-2.nonce-1.b.ctext2}_inv(pka)
Step3.
nonce-3.ctext1
Step4.
pka.a.{pka.a}_inv(pks).nonce-4.nonce-3.i.ctext3.{nonce-4.nonce-3.i.ctext2}_inv(pka)

Fig. 6 An intruder attack model over stored attribute based encrypted data during unsecured channel

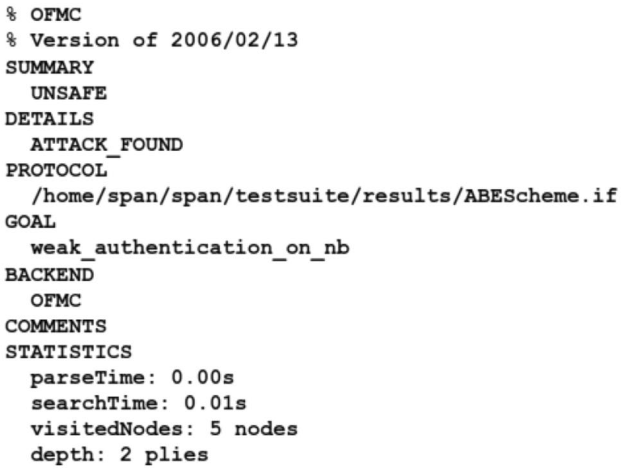
the encrypted key word search queries, the problem is that which does not support multi-level access control for cloud access. This algorithm only follows blind signature and pairing technique for encryption.

The cryptanalysis of ABE scheme discuss here are to access control mechanism of cloud users in certain parameters by the help of AVISPA tool to test the pitfalls of the protocol. In this we have implemented the role for group users as Alice and the data owner as Bob. In our cryptanalysis the below mentioned security goals are verified.

The result obtained in Figs. 7 and 8 are based on ABE scheme with respect to widely-accepted cryptanalysis codes of OFMC and CL-AtSebackends tools based on SPAN-AVISPA. Hence it is confirmed that ABE methodology is UNSAFE in OFMC, CL-AtSebackends under logic of BAN.The multi user authentication of variable length string attributes for group policy study in ABE is the further research required for multi data files encryption in clouds, also the complex attributes management needs to be addressed.

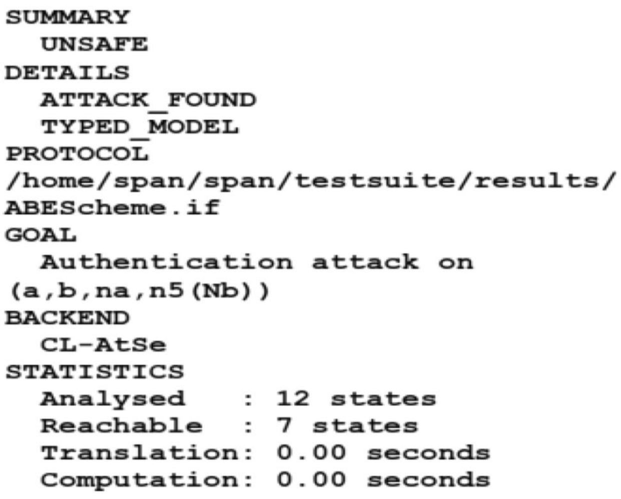
2.6 Third party auditing techniques

The cloud computing visualizes as next-generation public private infrastructures which moves application software and huge data bases into the centralized large data centers, Many public cloud data centers and levels of its services most of the time not completely trustworthy (Zhang et al. 2016). One of the well-versed techniques to check trustworthiness of a cloud server is third party auditor (TPA). The TPA provides data dynamic such as data operations, block modification, insertion and deletion in the cloud server as shown in the Fig. 9 (Genise et al. 2020). In this type of stored cloud data, the TPA checks the data integrity whether the data is accessed or modified by the intruder. The TPA provides certificate regarding the client data stored in the server (Sasikala et al. 2019).



```
% OFMC
% Version of 2006/02/13
SUMMARY
  UNSAFE
DETAILS
  ATTACK_FOUND
PROTOCOL
  /home/span/span/testsuite/results/ABEScheme.if
GOAL
  weak_authentication_on_nb
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.01s
  visitedNodes: 5 nodes
  depth: 2 plies
```

Fig. 7 Simulation result of ABE scheme under OFMC back-end for cloud



```
SUMMARY
  UNSAFE
DETAILS
  ATTACK_FOUND
  TYPED_MODEL
PROTOCOL
/home/span/span/testsuite/results/
ABEScheme.if
GOAL
  Authentication attack on
(a,b,na,n5(Nb))
BACKEND
  CL-AtSe
STATISTICS
  Analysed    : 12 states
  Reachable   : 7 states
  Translation: 0.00 seconds
  Computation: 0.00 seconds
```

Fig. 8 Simulation result of ABE under CL-AtSe back end for cloud

A client, who stores huge files in the cloud storage and believes that these files are maintained and computationally secure to access its client consumers and organizations. The cloud service provider manages cloud storage in terms of space and resource maintenance of client data. The third-party auditor gives assurance to the client and tests the security features of the cloud storage and report accordingly to the client (Fig. 10).

The client data stored in anonymous insecure remote server, the attacker may get the server access remotely. Here we considered few widely accepted assumptions. An attacker get access to server can modify the stored data. An attacker can guess vulnerable web access files in the server. The attacker may be a lawful person to access the files or vice versa. The pairing based cryptography is widely used for TPA in cloud data. Few of the well-known methods are

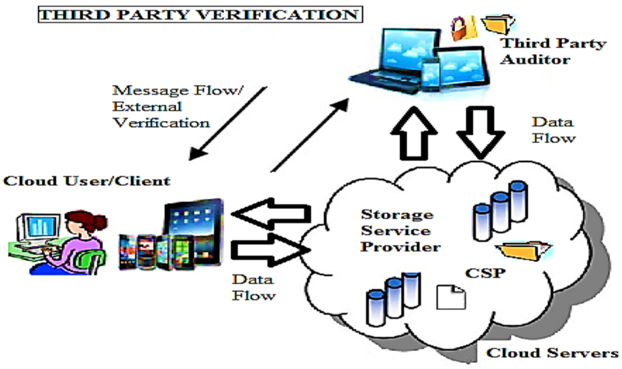


Fig. 9 Third party verification mechanism in cloud

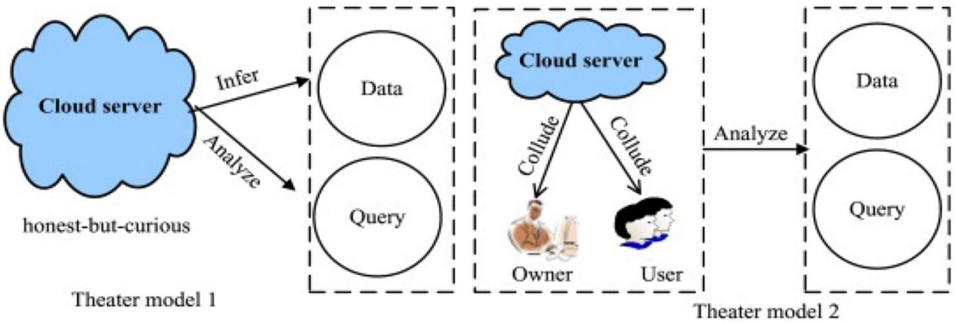


Fig. 10 An intruder attack model in pairing based DLP model cryptography

pairing based cryptography is discrete logarithm problem and elliptic curve cryptosystem. Bilinear pairings were initially express to the cryptographic group by Menezes, Okamoto and Vanstone with their assault. For instance, an unknown pairings can be utilized to for the discrete logarithm issue on a specific class of elliptic curve over a limited field to the discrete logarithm issue on a littler finite field. But a sub-exponential record math assault can be utilized to assault the issue.

For instance, let $P_A = <x>$ be the n order additive group with ∞ as the identity, and P_M be the n order multiplicative group with 1 as the identity element (Yu et al. 2016).

The bilinear pairing on (G_A, G_M) is given by $b\hat{}{:}P_A \times P_A \rightarrow P_M$ based on the following conditions:

1. Bi-linearity: For all M, N, U P_A, $b^{(M + N, U)} = b^{(M, U)e\hat{}(N, U)}$ and $b^{(M, N + U)} = b^{(M, N)}e^{(M, U)}$.
2. Non-degeneracy: $b^{(x,xP)} \neq 1$.
3. Computability: b^ can be efficiently computed.

2.6.1 Discrete logarithm and Diffie–Hellman problem

In discrete logarithm and Diffie–Hellman methods we considered the brief reviews of Zhang et al. for pairing based key authentication in cloud storage data, which executes several phases such as introduction, DLP and DHP description, limitations and cryptanalysis of the protocol and research issues. The DLP in presents cloud data authentication operations using bilinear mapping as follows. The DLP is based on the finite group of multiplicative field and finite points defined over elliptic curves. So, it is closely related to DHP. The DLP and DHP follows the following properties of bilinear pairing for authentication and authorization services in the cloud.Let P1 and P2 be cyclic of order m,n prime multiplicative groups of generator P_A. The Bilinear map b: P1 × P1 → P2 based the conditions.

1. Bilinearity: For all x, y ∈ p1and e, f ∈ Zp, b(xe, yf) = b(x, y)ef.
2. Non-degeneracy: b(p, p) = 1.
3. Compatibility: There exists an algorithm for computing b. (Bilinear pairing).

This proposal assesses to solve the problem of the demands by the cloud users for the trust worthiness of cloud storage.

The DHP works based on multiplicative groups of finite fields and group of points defined over finite field whichallowsTPA to validate the data integrity in different client data. Here we considered the DHP of bilinear pairing with the definition of DHP.

DH-Problem. In Diffie–Hellman problem, it is taken that f, f^x, f^y ϵ G for x, y ∈ Z^p, to calculate f^{xy}.

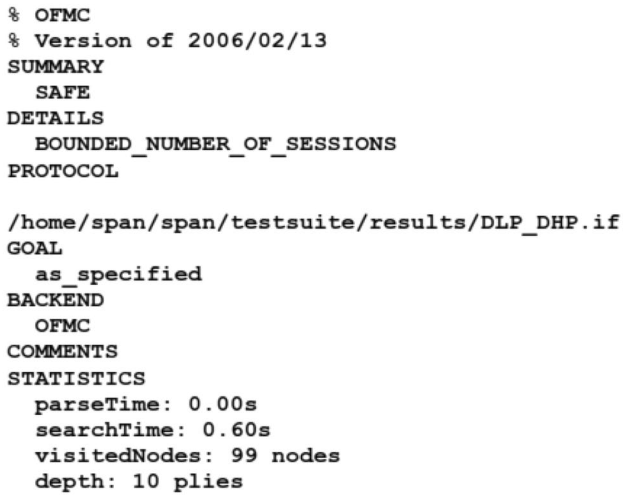
Here we assume that the (t, ∈) holds the time method which must have the probability to solve the DH-problem. The TPA needs to evaluate critical issues perspectives of cloud servers using group signatures, widely used DLP and DHP methods. However this technique only supports single user settings to validate for cloud servers.

To cryptanalyze the DLP and DHP scheme this section describes for public auditing mechanism using AVISPA tool. This widely accepted tool used for testing the drawback of the DLP and DHP protocol. In this we have implemented the role for cloud client as Alice and the TPA as Bob. In our cryptanalysis the following security goals are verified.

The Figs. 11 and 12 depicts the result of DHP scheme using the OFMC and CL-Atse code back ends using SPAN-AVISPA tool also DHP is SAFE under these back ends, respectively. The TPA verifiability of multiple audit cloud service sessions for different users and services of various outsourced files and also a multi-user setting with multiple service auditing tasks is the further research issue required.

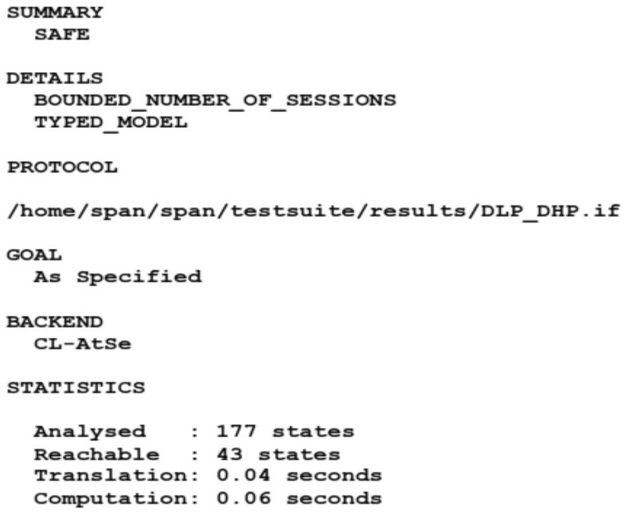
2.7 Key agreement and management techniques

Key agreement technique is a fine-grained key authentication scheme in multi user data sharing environment. The main challenge in key management scheme is its computational cost. The key agreement protocols use asymmetric key cryptosystem to manage multiple key which utilizes different key agreements. The cloud client uses



```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL

/home/span/span/testsuite/results/DLP_DHP.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.60s
  visitedNodes: 99 nodes
  depth: 10 plies
```

Fig. 11 Simulation result of DHP scheme in cloud using OFMC output back end



```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL

/home/span/span/testsuite/results/DLP_DHP.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed    : 177 states
  Reachable   : 43 states
  Translation: 0.04 seconds
  Computation: 0.06 seconds
```

Fig. 12 Simulation result of DHP cloud using CL-AtSe output back end

shared keys such as public keys for its users and the private keys to encipher and decipher various files in the cloud. In public cloud services the key agreement algorithms are run under in secure channel, the intruder may guess weak private keys with respect to already known public keys. The typical key management scheme model of cloud security attack is described by the characters initiator and target as shown in the Fig. 13. An attacker gets access to the public key of encrypted files. An attacker can guess low entropy passwords to decrypt all possible keys to open the stored search encryption files. The attacker may be a lawful person to access the files or vice versa.

2.8 RSA algorithm

In this section we present the brief reviews for key management scheme in cloud storage outsourced data, which executes several phases such as introduction, algorithm description, limitations, and cryptanalysis of the protocol and research issues. In RSA, to compute n = p*q, large prime numbers are considered as keys for the cloud server S. Here p,q are private keys and n is the public ke. Now en*dec = 1 mod (p – 1) (q – 1) is the one-way hash cryptographic function and the mapping as Hash h(.):{0,1}L. Here dec is the private key and en is the public key of the cloud server. In order to share keys from remote server, the cloud user generates both private key and public key pairs (Singh et al. 2013).

Step 1: The U_i cloud user initially chooses his/her desired identity ID_i, a random number b_i and password pw_i.

Then computes $pw_i$and sends through secure channel to the cloud server S.

Step 2: Using received message < ID_i, pwb_i >, the cloud server computes R_i, $B_i = pwb_i$ mod n, L_i and sends to the cloud client.

Step 3: After receiving the private key the cloud server securely shares to the authorized users.

In order to address the key sharing in the cloud network which prone to more vulnerable to insider attack. It leads to security draw back and cannot get user anonymity property.

2.8.1 Cryptanalysis of the protocol

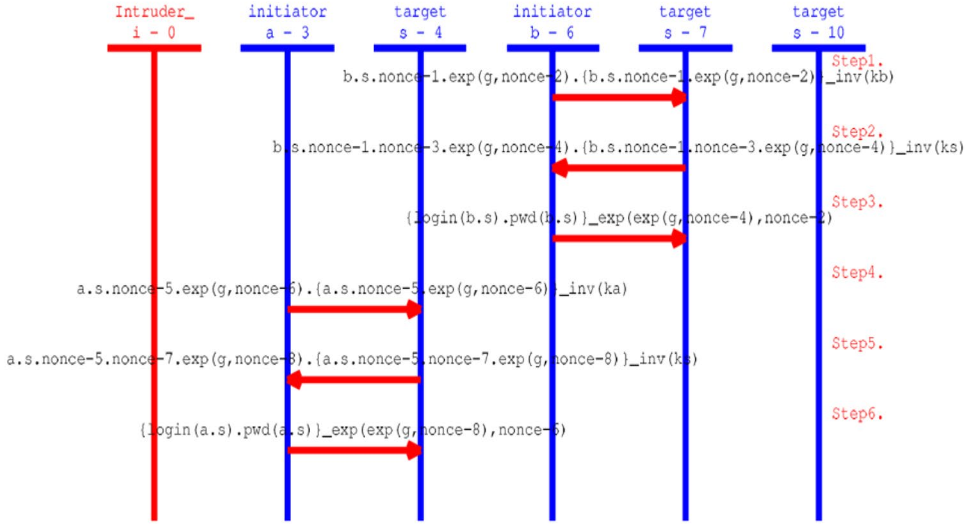
This section discusses regarding the cryptanalysis of RSA method for key sharing mechanism using AVISPA tool used for testing the pitfalls of the RSA protocol. In this we have implemented the role for cloud client user as Alice and the server as Bob. In our cryptanalysis the following security goals are verified.

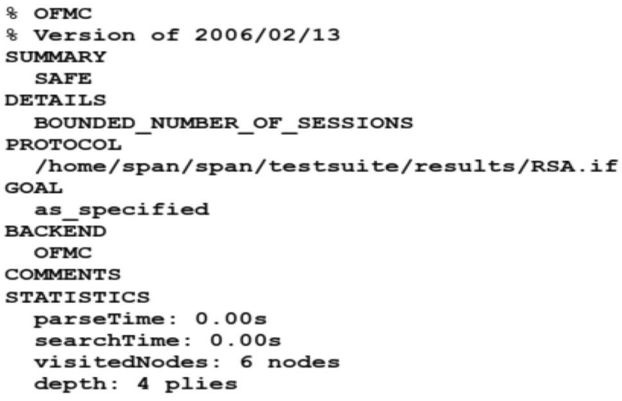
The Figs. 14 and 15 depicts the RSA method is SAFE in OFMC and CL-AtSe code are back-ends using SPAN-AVISPA. The insider attack and offline password guessing attack in RSA is the major pitfall which should be overcome in further research study.

2.9 Cryptographic one-way hash

This section discusses the brief reviews of key management scheme in cloud environment. The cryptographic one-way hash protocol is applied for one of key management schemes using digital signature, random sequence and mutual authentication methods. The one-way hash cryptographic method points from random length of string to a constant length strings called hashed value. This can be denoted as.

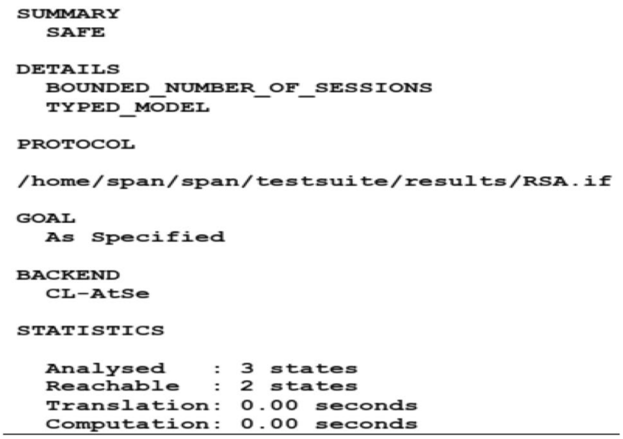
Fig. 13 An intruder attack model over encrypted data during unsecured channel


Intruder_
i - 0
initiator
a - 3
target
s - 4
initiator
b - 6
target
s - 7
target
s - 10
Step1.
b.s.nonce-1.exp(g,nonce-2).{b.s.nonce-1.exp(g,nonce-2)}_inv(kb)
Step2.
b.s.nonce-1.nonce-3.exp(g,nonce-4).{b.s.nonce-1.nonce-3.exp(g,nonce-4)}_inv(ks)
Step3.
{login(b.s).pwd(b.s)}_exp(exp(g,nonce-4),nonce-2)
Step4.
a.s.nonce-5.exp(g,nonce-6).{a.s.nonce-5.exp(g,nonce-6)}_inv(ka)
Step5.
a.s.nonce-5.nonce-7.exp(g,nonce-8).{a.s.nonce-5.nonce-7.exp(g,nonce-8)}_inv(ks)
Step6.
{login(a.s).pwd(a.s)}_exp(exp(g,nonce-8),nonce-6)



```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/RSA.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.00s
  visitedNodes: 6 nodes
  depth: 4 plies
```

Fig. 14 Simulation result of RSA scheme in cloud using OFMC code back end



```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL

/home/span/span/testsuite/results/RSA.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed    : 3 states
  Reachable   : 2 states
  Translation: 0.00 seconds
  Computation: 0.00 seconds
```

Fig. 15 Simulation result of RSA cloud on CL-AtSecode back end

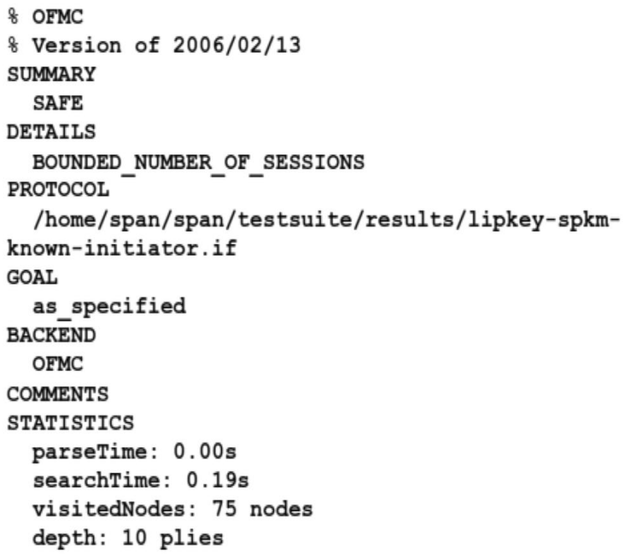
The hash h: U → V, where $\mathrm{U} = \{0, 1\}^*$, and $\mathrm{V} = \{0, 1\}$ n. Here the Uis the binary valued with random length and V is the constant string with n length *by satisfying the following conditions*:

i. *Easiness:* $X \in \mathrm{U}$, $y = h(\mathrm{x})$.
ii. *Pre-image resistant: Difficult* to estimate x from y, where $h(\mathrm{x}) = y$.
iii. *Second-pre-image resistant: Difficult* to estimate input $\mathrm{X} \in U$ so that $h(x) = h(\mathrm{x}^1)$ for $\mathrm{X} \in U$ and $\mathrm{x}^1 = \mathrm{x}$.
iv. *Collision resistant: Hard* to estimate a pair $(\mathrm{x}, \mathrm{x}^1) \in \mathrm{U} \times \mathrm{U}$such that $h(\mathrm{x}) = h(\mathrm{x}^1)$, where $\mathrm{x} = \mathrm{x}^1$.
v. T*ransformation:* $X \in U$, the hash $y = h(\mathrm{x})$ is impossible from string $\{0, 2n\}$,where Kis the output with hash $h(\cdot)$.

In order to address the key management problem the one-way hash method generates keys for different cloud users, keeping the set off for entire users may get this problem.

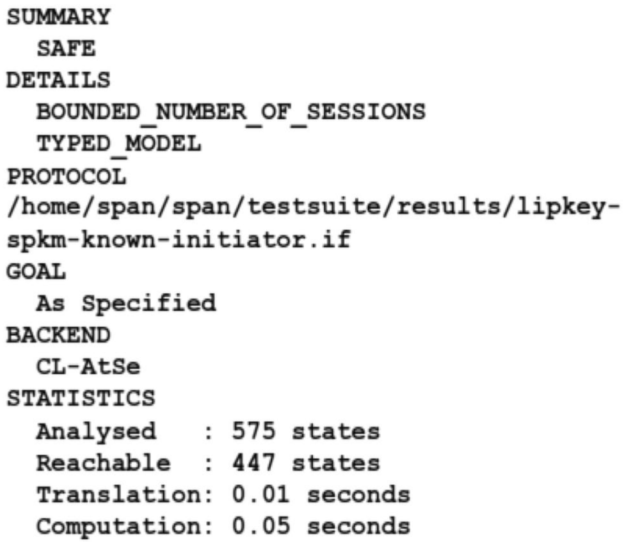
This section discusses about the cryptanalysis of cryptographic one-way hash protocol for key management scheme using AVISPA tool. It is used for testing the drawbacks of the one-way hash protocol. In this we have implemented the roles for cloud initiator and the role of cloud server as target. The Figs. 16 and 17 describes that the key management scheme which is SAFE in OFMC, CL-AtSe codes using SPAN-AVISPA in cloud environment. In our cryptanalysis the following security goals are verified.

The improvement of fixed length hash values in one-way hash for group of cloud users is the further research required for key management in clouds.



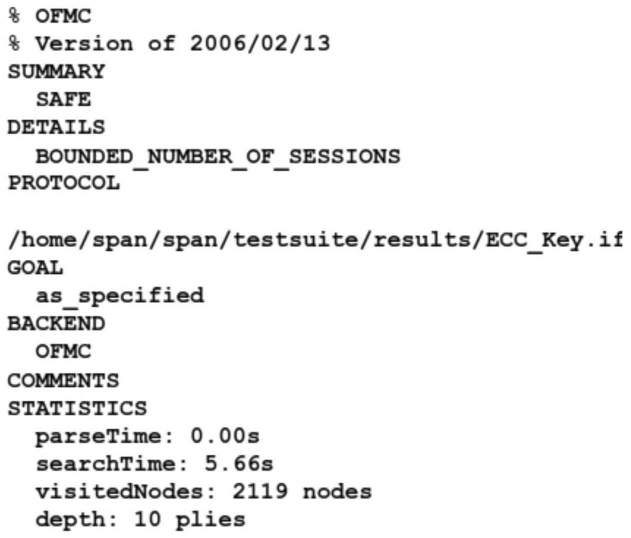
```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/lipkey-spkm-
known-initiator.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.19s
  visitedNodes: 75 nodes
  depth: 10 plies
```

Fig. 16 Simulation result of OPE scheme in cloud using OFMC code



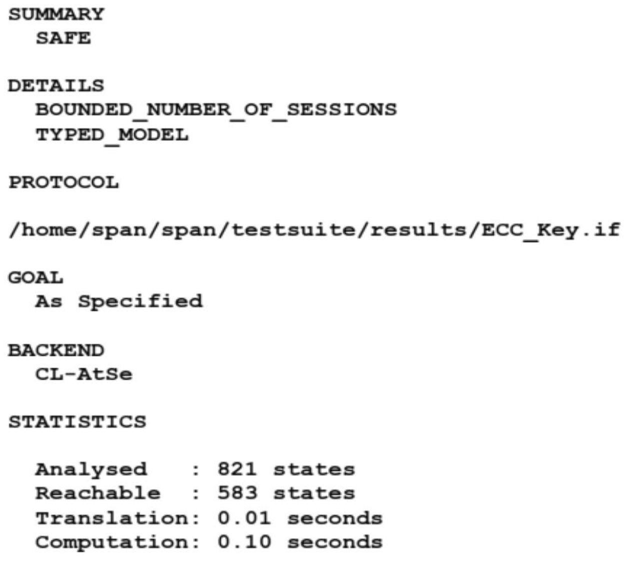
```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
/home/span/span/testsuite/results/lipkey-
spkm-known-initiator.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed    : 575 states
  Reachable   : 447 states
  Translation: 0.01 seconds
  Computation: 0.05 seconds
```

Fig. 17 Simulation result of OPE CL-AtSe code in cloud



```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL

/home/span/span/testsuite/results/ECC_Key.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 5.66s
  visitedNodes: 2119 nodes
  depth: 10 plies
```

Fig. 18 Simulation result of ECC scheme in cloud using OFMC code back end in cloud



```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL

/home/span/span/testsuite/results/ECC_Key.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed    : 821 states
  Reachable   : 583 states
  Translation: 0.01 seconds
  Computation: 0.10 seconds
```

Fig. 19 Simulation result of ECC using CL-AtSe code back end in cloud

2.9.1 Elliptic curve cryptography

The ECC is the most popular and widely usedcryptographic methodology in public key crypto system. In below section we present the brief reviews of key management methodologies in cloud servers. In many public key cryptographic it is observed that the Elliptic Curve Cryptography (ECC) provides enhanced security over the RSA, because it also achieves equal level of security strength minimum key size. The ECC uses 160-bit key length and the uses1024-bit key length, The EC equation is described in the form: $E_p(a, b)$: $y^2=x^3+ax+b(\text{mod } p)$ over a prime finite field F_p, where $a, b \in F$ and $8a^3+81b^2=0(\text{mod } p)$. In particular the ECC depends on DLP called ECDLP (Elliptic curve discrete logarithm problem). It is by taking two random points such that P and Q over $E_p(a, b)$, it is practically impossibleto define an integer $x \in \mathrm{F}_p^*$so that $Q=Xp$ when an integer exists.

2.9.2 Cryptanalysis of the Elliptic curve cryptography protocol

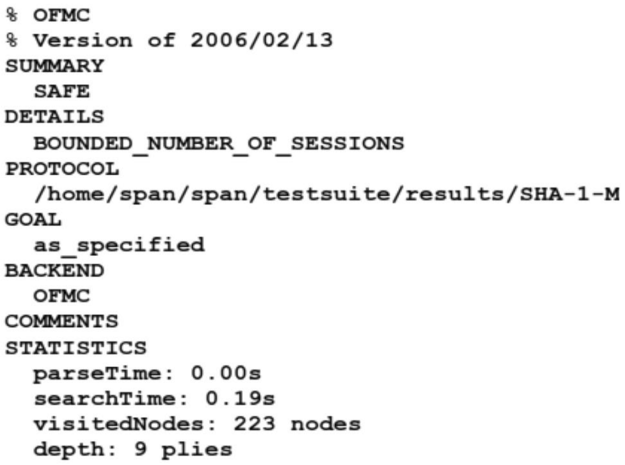
The ECC proposes simple and powerful key scheme for key management in cloud. In order to test this scheme the

AVISPA tool analyzes pitfalls in this protocol. In this we have described role for cloud user key generator as initiator and the cloud server as target. In our cryptanalysis the following security goals are verified. The Figs. 18 and 19 confirm that the ECC method is SAFE output in OFMC, CL-AtSe code backends using SPAN-AVISPA tool for cloud. Even though the ECC provides robust security, the computational cost is high in ECC when compared to other cryptographic methods, so it needs to be overcome with computations.

2.9.3 AES, MD5, SHA-1

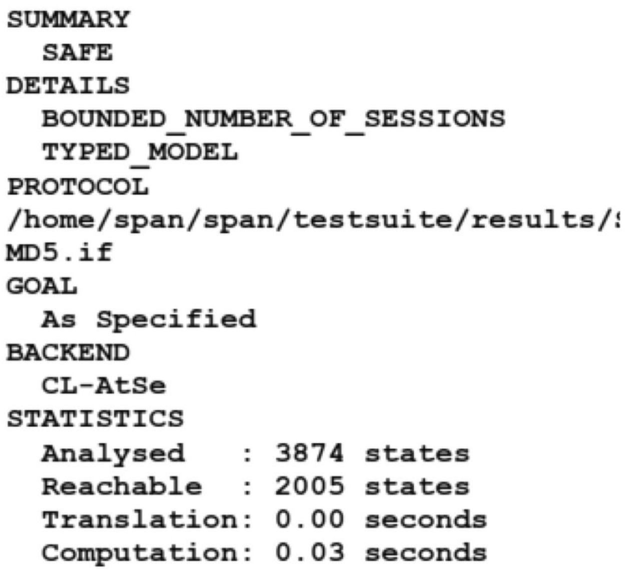
The third party auditor uses service level agreements (SLA) to verify cloud users cloud data integrity. It is obvious that storing cloud storage requires to enables that user of the cloud to validate the integrity, security and privacy of outsourced user data correctly. To store a file client first create suitable meta data file regarding the storage data, which will be utilized later stage to verifiability of the data integrity security and privacy at the cloud storage. The Advanced Encryption Standard-1 (AES) is also known as Rijindael, used protecting information based on the SLA's. Now-a-days the cloud data is analyzed extensively by symmetric block ciphers such as AES-1 and Message Digest-5 (MD-5) approaches. SHA-1 was obviously taken either MD5 or MD4, or both.

This section discusses regarding the cryptanalysis of MD-5 and SHA-1 scheme for integrity, security and privacy verification using AVISPA tool. Here the role for encryptions in cloud user as client and the cloud server as server. In our cryptanalysis the following security goals are verified.



```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/SHA-1-M
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.19s
  visitedNodes: 223 nodes
  depth: 9 plies
```

Fig. 20 Simulation result of cloud data integrity scheme in cloud using OFMC code back end



```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
/home/span/span/testsuite/results/
MD5.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed    : 3874 states
  Reachable   : 2005 states
  Translation: 0.00 seconds
  Computation: 0.03 seconds
```

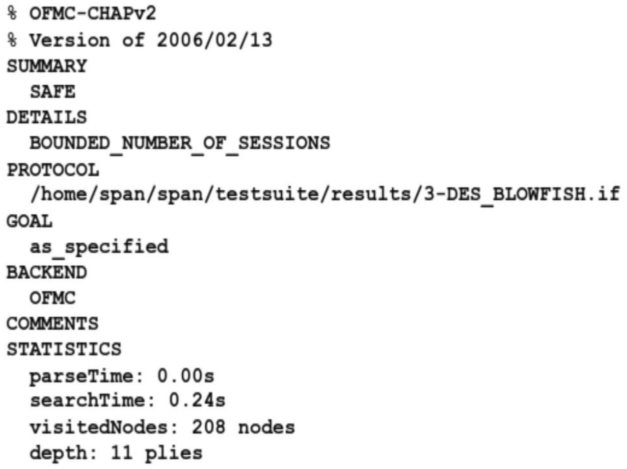
Fig. 21 Simulation result of cloud data integrity in cloud using CL-AtSe code back end

The Figs. 20 and 21 clarifies that the AES, SHA-1_MD-5 schemes are SAFE in OFMC, CL-AtSe code back ends in SPAN-AVISPA tool.

2.9.4 Blowfish and 3-DES

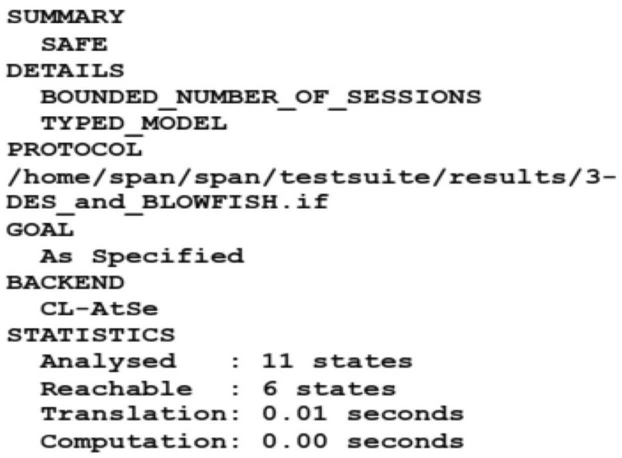
This section, discuss the DES, BlowFish methodologies and the brief reviews for data integrity, security and privacy in cloud storage outsourced data. The Data Encryption Standard (DES) is a block encipherment methods used to encipher the cloud user data blocks of 64 bits each size. The cycle of 64 bits text is input to again DES another round. Here it generates 64 bits of Cipher text. For both encryption and decryption, the same key is used. The key length of this DES algorithm is 56 bits, although the 64 bits key is actually input. The Blowfish encrypts the cloud data using 64 bit blocks with a changeable bit length key of up to 128–448 bits. The Blowfish was designed by Schneier, here the Blowfish key remains constant to suit many security applications where the keys are using used long time. This section discusses regarding the cryptanalysis of 3-DES and Blowfish schemes for privacy and security in cloud environment using AVISPA tool. Here the role for private key encryptions/decryptions in cloud user as client and the cloud server as server. In our cryptanalysis the following security goals are verified.

The Figs. 22 and 23 depicts that the SHA-1_MD-5 schemes are SAFE in OFMC and CL-AtSe codes in SPAN-AVISPA tool for cloud security and privacy.



```
% OFMC-CHAPv2
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/3-DES_BLOWFISH.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.24s
  visitedNodes: 208 nodes
  depth: 11 plies
```

Fig. 22 Simulation result of BLOWFISH and 3-DES scheme using OFMC in cloud



```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
/home/span/span/testsuite/results/3-
DES_and_BLOWFISH.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed    : 11 states
  Reachable   : 6 states
  Translation: 0.01 seconds
  Computation: 0.00 seconds
```

Fig. 23 Simulation result of BLOWFISH and 3-DES using CL-AtSe in cloud

3 Proposed lattice based cryptosystem

A third-party data storage audit service lets users permit a third party for that party to look at their data that is stored with a cloud service provider. The Verifier signature, which looks like a signer's signature, is one of the essential tools for this kind of service. Considering how cloud computing operates, cutting-edge computing approaches, such as quantum computing, must be tested and refined in a secure setting. The goal of this study is to come up with a CDIVM method based on lattices that can protect against quantum attacks.

The proposed Certificate less Data Integrity Verification Model (CDIVM) protocol developed using lattice integrity verification to combat from quantum-based supercomputing bouts (Genise et al. 2020). It contains only essential functions such as arithmetic processes on the matrix. The following segment gives assumptions for a lattice (Ni et al. 2020).

3.1 Security analysis

Definition 1 In Euclidean space, E^I is the M-dimensional lattice LN of vital combinations of linear vectors b1….bn. Lattice showed.

$$Lat(b1, ..., bn) = \{\textstyle\sum_{i=1}^{n} x_i b_i, x_i \in Z\} \tag{6}$$

If B = (b1, …, bn) is the lattice, then n some vectors and m are lattice dimensions.

Definition 2 Lettice (Lat) is a fraction of a number. Lat is any duplicates $qZ \subseteq L$ additionally closed. Only if the vector xi is in vector $x_i \in Z^m$ lattice Lat and x_i mod q lattice, let Z_q^{nxm} be a numbers M matrix with the same order.

$$\mathrm{Lat}\left(A^T\right) = \{ v \in Z^m : \exists s \in Z_q^n, \tag{7}$$

$$\text{such that } v = A^T s \bmod q \} \tag{8}$$

Here, vectors represent module q correspond to the lines of the first q-lattice and second formed by the lines of A.

Definition 3 There exists a $u \in Z^m$ vector. The value of $AX = u \bmod q$ describes the displaced cosset or lattice, which considered as follows.

$$\mathrm{Lat}^{\perp}(A) = \{ v \in Z^m = Av = u \bmod q\} = \mathrm{Lat}\perp(A) + X \tag{9}$$

Definition 4 Gaussian is different in latitudes.

Gaussian function is centered on any vector $c \in R^m$, positive $s > 0$ and parameters in R^m and C

$$R^m \forall x \in R^m, s, c\ (X) = \exp(-\| x - c \| / s^2) \tag{10}$$

Gaussian distribution defined as more than m-dimensional lattice L:

$$\forall x \in L,\ D_{L,S,C}(X) = \frac{\rho_{s,c}(x)}{\rho_{s,c}(L)} \text{ where } \rho_{s,c}(L) = \sum_{x \in A} \rho_{s,c}(x) \tag{11}$$

Trapdoor and model-based processes.

Our digital signature generation algorithm uses some one-way trap-way and model-based functions as follows.

Definition 5 TA produces a small source of a lattice with $q \geq 2$ and $m \geq 5nlgq$ Trapezoidal Trap-door (n, m, q), n, m, q and matrix A.

$L^{\perp}q$ (A) and $\left|\left|T_{\tilde{A}}\right|\right|$ <= m $\omega\sqrt{\log m}$

Yong et al. RDIC model based on specific identification with full data privacy protection. It designed with digital signatures based on identity. In particular, their Trial-Valid Reply Technique is an unequal group key contract technique between the validator and the environment. In severe extraction, Yong et al. techniques apply to bone and others. A small digital signature algorithm for authenticating sender identities and obtaining a sender's secret key using Public Kety Generator (PKG). Yong et al. technique used digital signatures based on unified identification to compute the metadata of data chunks. Due to the attractive property, Yong et al. technique can be stored for a single value when responding to a Trial. In the trial phase, the third-party verification authority (TPVA) makes the Trial by selecting a subgroup of chunk indicators and random values. The cloud environment evaluates the Trial chunk evidence in Valid Replay and transmits it to the TPVA. TPVA verifies the evidence using coupled based integrity verification. The algorithm shows a swift of the ID-RDIC.

Definition 6 (Lattice Another Definition Form). Lattices are discrete subgroups of the Riemann zeta function Rn. Let $V = v1, v2, \ldots, vn \in R^{nxn}$, where v1,v2,…,vn are n linear independent vectors.

The n-dimensional Euclidean space formed by the basis of V lattice $\mathcal{L}(V)$ is represented by $\mathcal{L}(V)=$ $\alpha(V) = \sum_{i=1}^{\mathbf{n}} \mathcal{L}(\mathbf{V}) = \alpha(\mathbf{V}) = \left\{\sum_{i=1}^{\mathbf{n}} \mathbf{c}_i \mathbf{v}_i : \mathbf{c}_i \in \mathbb{Z}\right\}$.

For $q, \mathbf{n}, m \in \mathbb{N}$, let $q \geq 2$ and $M \in \mathbb{Z}^{\mathbf{n} \times m}$ and define the q -ary lattice as follows:

$$\alpha^{\perp}(M)\text{or } A_q^{\perp}(M) : \mathbf{x} \in \mathbb{Z}^m : M\mathbf{x} = 0\text{mod}\mathbf{q},$$
$$w \in \mathbb{Z}_q^{\mathbf{n}}, \alpha_w^{\perp}(M) : \mathbf{x} \in \mathbb{Z}^m : M\mathbf{x} = w\text{mod}\mathbf{q}.$$

Let q > =2 and A ∈ ℕ ∈Znxm be constants for q, n, and m ∈ N, and define the q-ary lattice as follows:

"A q (M):x ∈ Zm:Ax = 0 mod q,@w ∈ Z qn, w (M):x ∈ Zm:Ax = w mod q" or " A q (M):x ∈ Zm:Ax = w mod q,@w ∈ Z qn, w (M):x ∈ Zm:Ax = w mod q" or " A q (M):x ∈ Zm:Ax = w mod q,@w ∈ Z qn.

The dual lattice of is symbolised by the symbol, which is defined as * = x ∈ Rn:v,x,v ∈ Z in the formula.

Definition 7 Small integer problem solution (SIPS) is defined as a problem with a tiny integer solution. Finding a nonzero integer vector x ∈ Zm fulfilling Ax = 0 mod q and xm is a problem for any n ∈ Z given positive numbers q, m ∈ Z in a matrix A ∈ Z q(nm) and an R, SIS (n, m, q) problem for any n ∈ Z given positive integers q, m ∈ Z in a matrix A ∈ Z q(nm).

Definition 8 SIPS issue with inhomogeneous solution. It is defined as follows for any n ∈ Z and positive integers q, m ∈ Z, A ∈ Z q(nm), and R: For any n ∈ Z and positive integers q, m ∈ Z, A ∈ Z q(nm), and R: The SIPS _(n,m,q) problem is defined as follows: Given a vector y ∈ Z (q′)n, find a vector x ∈ Zm such that Ax = y mod q and x is a positive integer. When considering the worst-case scenario, the difficulty of SIPS is based on lattice issues.

Hardness Problem: Solving (I)SIS issues on an arbitrary n-dimensional lattice is as difficult as solving GapSVP and SIVP problems on an arbitrary n-dimensional lattice for any polynomial limits m, and a prime integer q(nlogn).

This security challenge attempts to capture the many need for valid evidence of challenging chunks without chunking the opponent's entire file. In this challenge, the communication between the opponent and the Trialr is as follows:

Step 1: Run the master secret key (MSK) and setup algorithm to generate public parameters and transfer public parameters to keep MSK secret.
Step 2: To get the index indicator from the Trial, the contestant selects some data chunks. Such as the critical question, hash question and digital signature generation. Trailer computes the standard index for each data chunk and sends it to Reversible.
Step 3: The Trailer makes the Trial and asks the opponent to assess the Trial's relevant evidence.

Step 4: The respondent evaluates the Trial's evidence and sends it to the respondent in Valid Replay.

Step 5: If the proof is valid, the cynical challenge wins.

Confidentiality against TPVA indicates that no data leaked to TPVA during the integrity testing process.

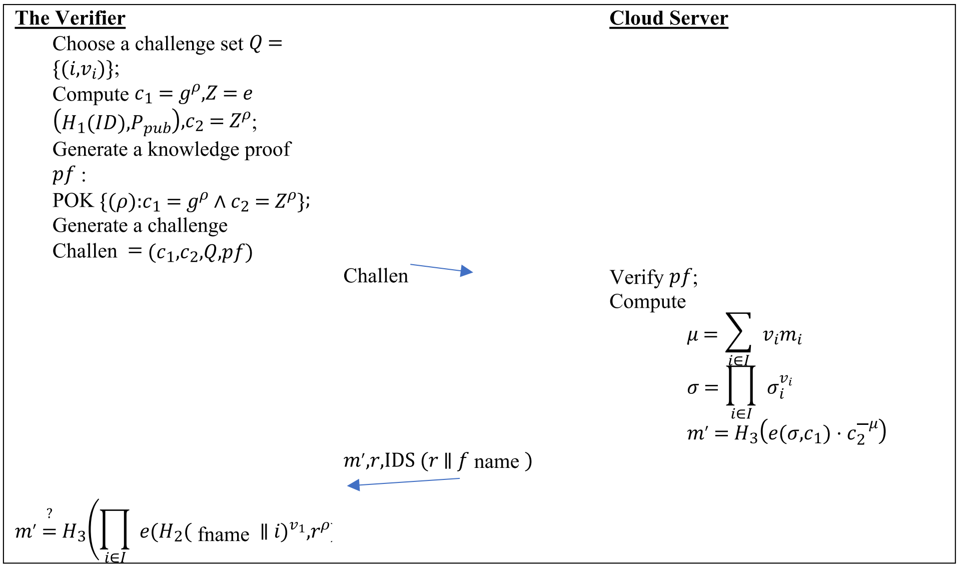
3.2 Assumption

The most challenging part of the lattice setting is the small integer solution of the SIS problem. It is when you are given a matrix A and need to find a non-zero vector v with mod q. It is because the search starts with random matrix A→←Z_ q^(n × m) find a vector v→ ∈ Z^m∖{0} such that A→v→ = 0 and ‖v→‖ ≤ β (Figs. 24, 25).

3.3 CDIVM Scheme

Setup: After a security parameter has been considered, a probabilistic time technique is used to get system parameters.

Key Generation Phase: Based on the system parameters sp, a probabilistic time algorithm creates key pairs for a signer S and a verifier V.



The Verifier

Choose a challenge set $Q = \{(i,v_i)\}$;

Compute $c_1 = g^{\rho}$,$Z = e\big(H_1(ID),P_{pub}\big)$,$c_2 = Z^{\rho}$;

Generate a knowledge proof pf :

POK $\{(\rho):c_1 = g^{\rho} \wedge c_2 = Z^{\rho}\}$;

Generate a challenge

Challen $= (c_1,c_2,Q,pf)$

Challen

Cloud Server

Verify pf;

Compute

$$\mu = \sum_{i\in I} v_i m_i$$

$$\sigma = \prod_{i\in I} \sigma_i^{v_i}$$

$$m' = H_3\big(e(\sigma,c_1) \cdot c_2^{-\mu}\big)$$

m',r,IDS ($r \parallel f$ name)

$$m' \overset{?}{=} H_3\Big(\prod_{i\in I} e(H_2(\text{ fname } \parallel i)^{v_1},r^{\rho}$$

Fig. 24 Verifier and Server Technique Phases

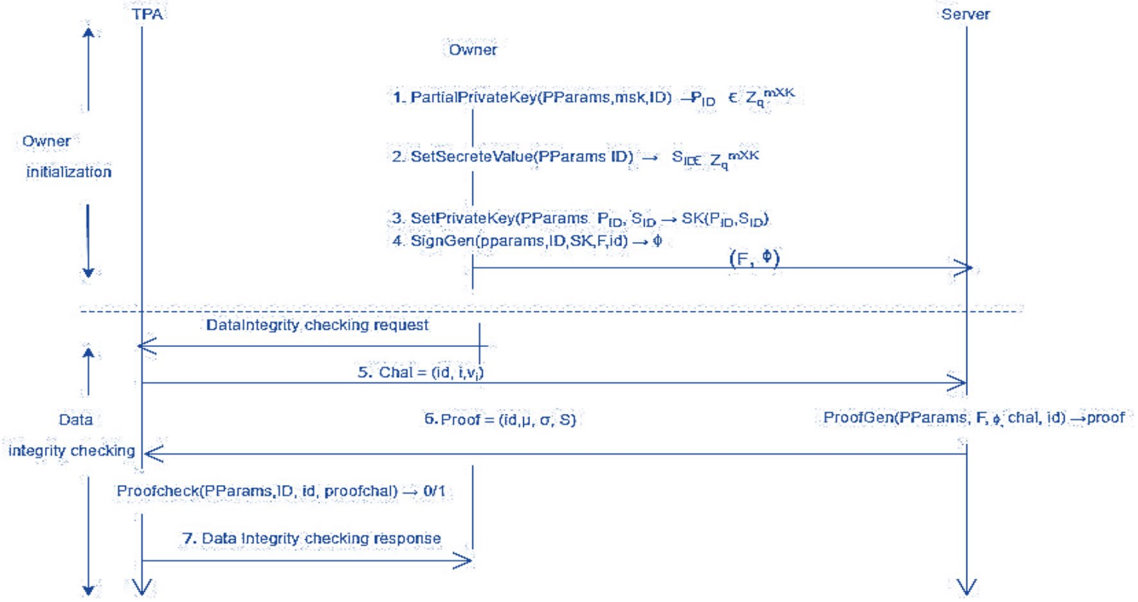

TPA
Server
Owner
Owner
initialization
1. PartialPrivateKey(PParams,msk,ID) →P_ID ∈ Z_q^mXK
2. SetSecreteValue(PParams ID) → S_ID∈ Z_q^mXK
3. SetPrivateKey(PParams P_ID, S_ID → SK(P_ID,S_ID)
4. SignGen(pparams,ID,SK,F,id) → ϕ
(F, Φ)
DataIntegrity checking request
5. Chal = (id, i,v_i)
Data
integrity checking
6. Proof = (id,μ, σ, S)
ProofGen(PParams, F, ϕ, chal, id) →proof
Proofcheck(PParams,ID, id, proofchal) → 0/1
7. Data Integrity checking response

Fig. 25 Diagram of CDVM flow

Signature Generation Phase: Based on the parameters sent to it by the system, the algorithm makes a signature z. It looks at the message, as well as the verifier's public key (y V) and private key (x S).

Verification Phase: A deterministic time algorithm considers the system parameters, such as the public key y S of a signer, the message, and the signature z, when making a verification decision.

Simulation Phase: A probabilistic time method is used to get a signature z from the parameters put into the system. It looks at the message and the verifier's public key, y S, and private key, x V.

If the data storage in the cloud is intact then specific CDIVM is built in two stages, the initial stage and the data integrity verification stage. Early stages include {Setup, Extract-Partial-Private, Set-Privacy-Value, Season Algorithms, secret key created using the setup algorithm and the trap vision function. The extract-partial-secret key algorithm uses the model ID to compute the partial secret key 0,1} * consistent to the sender ID. The set-secret-value algorithm returns a randomly selected matrix as a hidden sender value. The set-private-key algorithm, the secret key, is partially secured with a secret value, and the sender gets the entire secret key. The signage algorithm uses the sample function to obtain the digital signature of the file. This subdivision discusses the analysis of the CDVM technique. Unreliable CSPs protect the security of a specific chunk. The integrity verification phase includes the Trial, Proof-Generation (ProGen) and Proof-Check (ProChe) algorithms. In ProGen, after accepting the Trial, the CSP receives the evidence and forwards to TPVA. In the Signature Proof Verification ProChe, the TPVA examines the evidence, i.e., If it is valid evidence thatthe cloud data is protected if valid, Otherwise the data reverified.

Let M > Cn log q and q (log n) be the most massive charger of the shift (n) as security parameters be the Gaussian variable. Setup (n): Considering the security variable using PKG runs the small mesh base of the trapdoor hashgen (n, m, q) algorithm and lattice (a) to obtain the matrix A. It also runs the cloud trapdoor hashgen. (n, m, q) To construct matrix B using small b. PKG then selects three secure hash functions; additionally, TA has msk = TA.Technique for storage accuracy. Privacy against TPVAs displayed in the following subdivisions.If both the cloud environment and the TPVA are working correctly, the CSP expects the integrity check to be successful. Evidence for the accuracy of the proposed scheme is as follows. This subdivision highlights the specific technique's security techniques, i.e., the latitude protected from the CSP without the severe problem of SIS in the data (Table 1).

Table 1 Notations used in this paper

Symbol	Description
TTPA	Third-party verification authority
PKG	Public Kety Generator
CSP	Cloud server provider
PPara	Public key Parameters
ID_i	$User_i$'s identity
PW_i	User i's password
ID_j	Cloud-Server j's identity
σ	User Signature using secret key
Hf(.)	Hash Function
□	Adversary
f_i	File Chunks
δ	Independent vector

```
Algorithm 1: Signature Generation Phase.
Input: $S_k$, F, ID, Lat-ID, PPar,
Output: $\sigma$
Step 1: Compute
$\alpha_j \leftarrow Hf(IDi \| IDj \| k) \in Z_q^m$ ,where $i, j, k \le n$
Step 2: For file Chunk fi, calculate
$\delta_i \leftarrow Hf(IDj \| k) \in Z_q^m$ ,where $\delta_i \in Z_q^{m*n}$
Step 3: Calculate the Dot Product
$hf_{ij} = (hf_{i1}hf_{i2}....hf_{in})$ So $hf = \delta_i / C$
Step 4: Calculate Sample Image Function $\sigma_i$ =SamPreImg
(A, $PW_{i,}$ $hf_{ij}$ , $S_k$)
Step 5: If Cloud user $U_j$ sends the File $F_i$ to cloud server $U_j$
with signatures $\mu = \{\sigma_1, \sigma_2, ...\sigma_n\}$ then deletes the file F
chunks $f_{i.}$
```

```
Algorithm 2: Signature ProofVerification Phase.
Input: Ch, F, $\mu$ , ID, PPar,
Output: SigProof=( $\sigma$ , S, LatID, $\phi$ )
Step 1: Compute Data Blocks
$\phi' = \sum_{i=C_1}^{Cr} Ve_i fi \mod q$, Where $\phi' \in Z_q^m$
$\sigma = \sum_{i=C_1}^{Cr} Ve_i fi \mod q$, Here the $\sigma \in Z_q^m$
Step 2: Cloud Verifies Vector $\lambda \in Z_q^m$ , here $\lambda$
Randomparameter
Step 3: Verifies If $\| \lambda \| \le \delta$ Repeat Until $\| \lambda \| \le \delta$
Satisfies Otherwise Skip
Step 4: If CS calculates $\sigma'$ = B $\lambda$ Mod q and
$\zeta = Hf2(\sigma')$
Step 5: Then $\phi'$ and $\lambda$ are $\phi = \zeta . \phi' + \lambda$ mod q.
Step 6: CS Returns SigProof=( $\sigma$ , S, LatID, $\phi$ ) to TPA
```

3.4 Proof of correctness

During security-resistant production, CSPs are considered hostile CSS, which Trial the sender/TPVA. Here the Tracer follows random crack oracles and answers hash questions appropriately. Suppose the challenge has multiple times (PPT) to win. The performance of the proposed CDVM project is estimated by dividing 40 MB F into 40 MB size chunks. |N| Validation function based on existing network = 20 bits, | q = 160 bits, lattice size of 2nlogq m. The estimated cost of the CDVM is computed based on the cryptographic activity shown in Table 2.

Table 2 CDIVM computational time comparison

Chunks	Proposed protocol			Yu et al. (2017)		
	ProofGeneration (ms)	ProofVerify (ms)	Total computation time (ms)	ProofGeneration (ms)	ProofVerify (ms)	Total computation time (ms)
100	36.75	17.31	54.06	175.65	32.925	208.575
200	73.5	33.81	107.31	336.96	56.175	393.135
300	110.25	50.31	160.56	498.15	79.425	577.575
400	147	66.81	213.81	659.4	102.645	762.045
500	183.75	83.31	266.85	820.65	125.925	826.575
600	220.5	99.81	320.31	981.9	149.175	1131.075

Algorithm 3: Post Quantum Self-Certified Signature Scheme.

(1) Initialization: It takes as input a security parameter n and returns the system parameter pp.

(2) Initialization: It accepts as input a security parameter n and returns the system parameter pp. Second, the program's operation is as follows: it receives as an input the security parameter n and outputs the system parameter pp. CA selects the master private key s and generates the CA public key P CA from the master private key.

(3) Extract: Each user UI selects his private key and the partial public key Y id, after which he sends the partial public key to the Certificate Authority (CA). In response to receipt of the request, CA extracts the "id" of the user's partial private key from the request and stores it in its database. The letters (P CA,ID,Y id) represent the whole of the public key, while the letters (P CA,ID,Y id) represent the entirety of the private key (x id,s id).

(4) To finish off, the user creates a digital signature sig for the message m by using the previously produced private key (x id) and the public key (s id).

5. Verify the signature signature is valid: A verifier checks to see if the signature signature is correct.

4 Experimental design and discussion

The broad adoption of cutting-edge technology may be directly attributed to the advent of cloud computing. Consider the case of a government agency that has developed its computerized data service to be delivered to citizens through cloud computing. Users of these services may also download and save extensive media on their mobile gadgets. Most cloud storage companies provide users with free space to store their data such as Amazon Web Services (AWS), Google Cloud Platform (GCP). However, there are some ways customers' data is lost, including hardware failures and inefficient optimization strategies.

A data audit service is a part of a cloud storage system that ensures the data is correct. Three different people are involved in this method: the data auditor, the cloud service provider (also called CSP), and the data owner. The person who owns the data can use a mobile device to make and save multimedia content. In contrast, the person who audits the data can use cloud computing. The information will stay private if the following steps are taken, which can be done by all three parties.

The owner of the data sets creates a DVS signature for each block in the data set. It is done after the files are split into Chunks. The data auditor then checks to make sure the data are correct. The data owner saves its blocks and signatures in a cloud storage system. After that, the data auditor will decide whether the information is accurate.The data auditor takes random pieces of the user's information. The data auditor and owner must agree on a policy for the number of blocks, and that policy is met. After that, the data auditor looks at the signature for each block to make sure it is valid. After figuring out if the data in the query are accurate, the data auditor will make a report for the data owner. The owner of the data and the data auditor will then have to develop a policy that will let them check that the data stored in the cloud is accurate. For example, the data auditor should look over the data regularly or whenever it is asked of them.

A data auditor who is not trustworthy could try to get the user's data file back by getting each data block one at a time. But the DVS can stop this behavior by showing the user's data file and the signatures that go with it. Since the data auditor is the person who checks the data, the signatures on the files that were attached to the audit are useless.

The computational cost of generating digital signatures for data chunks in the initial model is lower than the Yong protocol. CDVM implemented using matrix–vector or matrix–matrix multiplication and additional lattice-based certificate-less digital signatures. Therefore, doing those tasks is less likely than Yong-style parsing and exponential processes (Xu et al. 2020).

It counts as the length of the file we source. F | And digital signature/tag length. File F divided into plan and n chunks, respectively, in the specific plan and the Yong plan. The amount of data stored in the CDVM. F |+(m × m) 160 bits. It is more than the long protocol because the digital signature size is larger in a particular system. In general, the impact of these storage costs on the overall performance of the system is minimal; it is because it is computed only once by a lifetime data collection system that lasts for tens of thousands of years. Here, the cost of communication between the TPVA and the environment computed in the Trial-Valid Replay process. = {ID, I} Here, the authentication process begins by asking the environment to compute the Valid Replay to TPVA. Then, the environment TPVA will have proof = (id, s, s)… C (|q|) bits is a challenging communication cost, Also, in the proof message and S. For a proof generation, only hashing arithmetic operations on the matrix are required for a specific CDVM, as shown in the table. In contrast, the Yong protocol requires exponential and coupling processes. These activities affect the total calculation time when dealing with challenging chunks in practical applications. CDVM price is proof of the prices of the matrix–vector–multiplier.

Specifies three modules: Sender Module (U-Module), Environment Module (S-Module) and Verifier Module (V-Module), The U-module implements sender resources such as desktop computers, including keygen and syngenic algorithms. The S-module embedded in the Amazon EC2 (Amazon Elastic Compute Cloud), which includes a proof gene algorithm. The V-module is implemented towards the verifier and contains a proof check algorithm. In use, Amazon uses the EC2 is an example of this type. The EC2 Compute unit supports a CPU capacity of 1.0—1.2 GHz, similar to the 2007 Octetron or 2007 Zion processors. The U-Module and V-Module powered by a 2.33 GHz Intel Core 2 Duo processor running on Ubuntu 14.04 System OS with 4 GB of RAM. The algorithm used NTRU-CRYPTO tool. If the file size is F, then 40 MB is divided by 4 KB, i.e., = 10,000; |n|= 20 bits |q|= 160 bits.

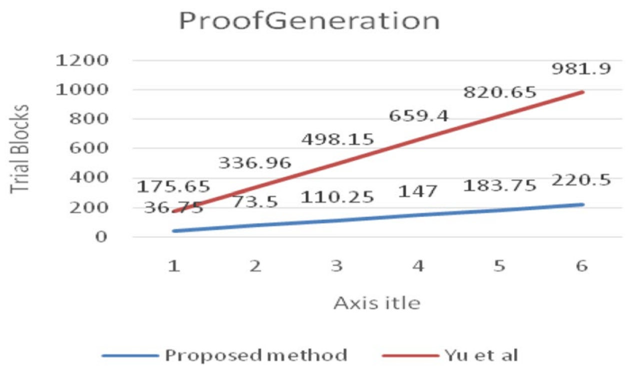


Fig. 26 Proof generation comparison in milli seconds (ms) X-axis depicts the number of rounds trial, Y-axis depicts the trial cloud data chunk blocks

The proposed CDIVM scheme implemented using lattice-based integrity verification. The Yong protocol based on alignment-based integrity verification. In the experimentation, the challenging chunks enlarged from 100 to 600 by growing the number of chunks. The specific protocol for tracking the total calculation time in both the Yong protocol and the Yong protocol with each test's algorithm shown in Figs. 26, 27 and 28. It shows that the proposed protocol achieves a shorter calculation time than Yong protocol, as the CDVM implemented using lattice-based digital signatures. The Yong protocol involves matrix–matrix or matrix instead of exponential processes, exponential processes. Contains vector multiplication or addition processes. In both of these protocols, the calculation time increases as the Trial chunks increase. As the number of challenging chunks increased, the number of random values added to the Trial group. Therefore, the environment is required to create the evidence, and the TPVA must verify that evidence. The counting time increases as the Trial chunks increase.

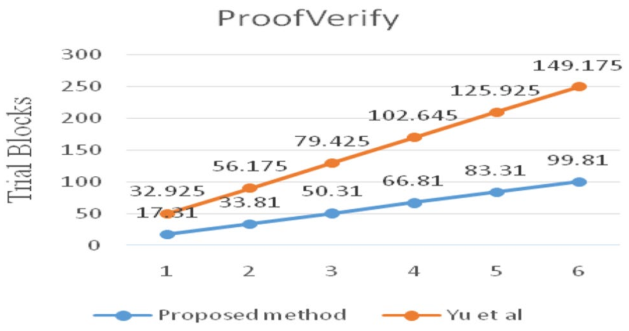


Fig. 27 Proof verify comparison milli seconds (ms). X-axis depicts the number of rounds trial, Y axis depicts the trial cloud data chunk blocks

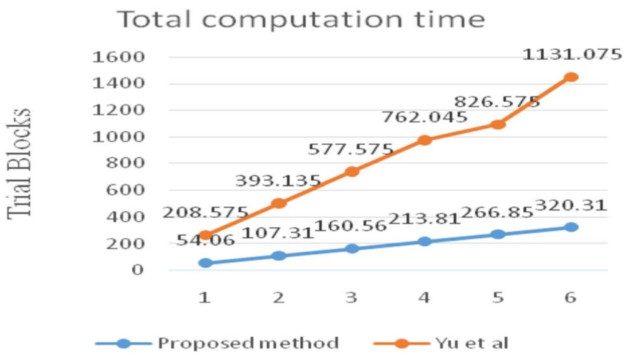


Fig. 28 Computational Time comparison milli seconds (ms). X-axis depicts the number of rounds trial, Y axis depicts the trial cloud data chunk blocks

Table 3 The comparisons of computational costs

Scheme	Signing cost	Verify cost
Li et al. (2021)	$t_{\mathrm{bpm}} + t_h$	$3t_{\mathrm{bp}} + t_h$
Tahat et al. (2020)	$4t_{\mathrm{eccm}} + 3t_h$	$2t_{\mathrm{bp}} + t_h$
Li et al. (2020)	$7t_{\mathrm{mul}} + t_h$	$t_{\mathrm{eccm}} + 3t_h$
Yang and Li (2021)	$4nt_{\mathrm{mul}} + t_h$	$4nt_{\mathrm{mul}} + 2t_h$
Proposed certificateless data integrity verification model	$3nt_{\mathrm{mul}} + t_h$	$3nt_{\mathrm{mul}} + t_h$

Table 3 depicts the Cost comparisons depending on the amount of processing power available. We examine at the cost related with computing resources in this paper. Assign the following values to the execution times: t bp, t bpm, t eccm, t mul, and t h. The following values should be assigned to the execution times: Multiplication operations that include a bilinear pairing operation, a scale multiplication for the bilinear pairing and an arithmetic multiplication for the ECC are included. There are also polynomial and arithmetic multiplying operations.

We will examine the storage overheads in Table 4 between the two options. In order to lower the dimensionality of the trapdoor, we use an enhanced trapdoor generation technique to determine the key size. In order to decrease the length of the signature, we only employ Gaussian sampling in the key extraction phase and the rejection sampling approach in the signature creation phase. This technique helps to reduce the length of the signature. Additionally, Lemma 2 in [14 states that if the key is distributed as a discrete Gaussian distribution with the parameter so that $\| \mathbf{x} \| \leq s\sqrt{m}$, the bit size of x is limited to mlog2s bits.

We are comparing the properties of security systems. Compared to Li et al. with |G|= 320 bits and |G_1|= 1024 bits the SCS systems over lattice implemented by Li et al. and Tahat et al. are more efficient, as indicated in the previous Tables 4 and 5. Because the security is based on pairing or elliptic curve discrete logarithm challenges, they are not secure against other sorts of attacks, despite the fact that their approaches are safe against quantum assaults (ECDLP). According to Li et al., the method they present is the first SCS approach to be applied over a lattice. Because of the use of the NTRU lattice, it has a reduced key size and signature length than other algorithms, but it does not offer demonstrated security. Due to the fact that our technique is based on the standard lattice, it is less efficient than the Li et al. scheme in terms of key size and signature length, which are both important considerations. Using the SIS assumption, on the other hand, our solution is provably safe in the ROM, but the previous approach is not. As a consequence, our method is more resistant to being attacked by quantum computers than previous techniques.

5 Conclusion

The goal of this study is to come up with a unique and reliable security model for cloud-based applications that are based on quantum mechanics. With the presentation of this paper, many efforts have been made to cryptanalysis post-quantum strategy used in cloud storage environments, data integrity verification by third-party audits, and other things. BN logic rules ensure that cyber-physical cloud systems are secure and users can log in. There are two steps to think about regarding cyber-physical Cloud Storage Systems.

Table 4 The comparisons of storage costs

Scheme	Public key size	Private key size	Signature length
Li et al. (2021)	$\lvert G_1 \rvert$	$\lvert G_1 \rvert$	$2\lvert G_1 \rvert$
Tahat et al. (2020)	$2\lvert G\rvert$	$4G_1 \mid$	$3\lvert G\rvert$
Li et al. (2020)	$n\log q$	$n\log q$	$n\log q$
Yang and Li (2021)	$nk\log q$	$mk\log(2s) + mk\log(2d + 1)$	$2m\log 12\sigma$
Proposed certificateless data integrity verification model	$n\log q$	$m\log(2s) + m\log(2d + 1)$	$m\log 12\sigma$

Table 5 The comparisons of security properties

Scheme	Provable security	Assumption	Postquantum
Li et al. (2021)	Yes	Bilinear	No
Tahat et al. (2020)	No	Eairings	
Li et al. (2020)	No	NTRU CVP	Yes
Yang and Li (2021)	Yes	Lattice SIS	Yes
Proposed certificateless data integrity verification model	Yes	Certificateless lattice	Yes

Many cryptographic security protocols have been used in these fields to keep data in the cloud secure, private, and integrity. These identified algorithms are most important for protecting privacy, ensuring data integrity, controlling access, and storing data safely. Most of these protocols are looked at with the BAN-AVISPA tool and mathematical logic codes in a BAN. An informal way to show that the protocol is essential. Most of the protocols that are already in place are only good for stopping or discouraging basic attacks, not for eliminating or preventing more advanced attacks. But these algorithms may not stop attacks like guessing passwords, DDOS, and other cyberattacks. So, these protocols are not safe against cyberattacks at an advanced level. These results and proofs show the detailed CDIVM protocol checks the integrity of our source data without saving the complete file from the cloud environment. An analysis of security shows that the CDIVM protocol protects against various CSPs that are not trustworthy. It keeps the information secret from Third Party Verification Algorithm. (TPVA) Analysis of performance and test results are compared to the Yong protocol, Yongqiang Zhang et al., and Yang et al. Crypto Scheme, and the proposed CDIVM scheme works better.

Data availability No external dataset source is taken for this paper. Completely the data is generated using own simulation experiments.

References

Amin R, Biswas GP (2015) Design and analysis of bilinear pairing based mutual authentication and key agreement protocol usable in multi-server environment. Wirel Pers Commun 84(1):439–462

Amin R, Kumar N, Biswas GP, Iqbal R, Chang V (2018) A light weight authentication protocol for IoT-enabled devices in distributed Cloud Computing environment. Futur Gener Comput Syst 78:1005–1019

Babu SD, Pamula R (2020) An effective block-chain based authentication technique for cloud based IoT. In: International conference on advances in computing and data sciences. Springer, Singapore, pp. 305–319

Boldyreva A, Chenette N, O'Neill A (2011) Order-preserving encryption revisited: Improved security analysis and alternative solutions. In: Annual cryptology conference. Springer, Berlin, pp. 578–595

Genise N, Micciancio D, Peikert C, Walter M (2020) Improved discrete Gaussian and subgaussian analysis for lattice cryptography. In: IACR International conference on public-key cryptography. Springer, Cham, pp. 623–651

Li D, Chen H, Zhong C, Li T, Wang F (2017) A new selfcertified signature scheme based on ntrusing for smart mobile communications. Wirel Pers Commun 96(3):4263–4278

Li H, Liu L, Lan C, Wang C, Guo H (2020) Lattice-based privacy-preserving and forward-secure cloud storage public auditing scheme. IEEE Access 8:86797–86809. https://doi.org/10.1109/ACCESS.2020.2991579

Li H, Guo F, Wang L, Wang J, Wang B, Wu C (2021) A blockchain-based public auditing protocol with self-certified public keys for cloud data. Secur Commun Netw 2021(1):6623639–6623710

Ni J, Zhang K, Yu Y, Yang T (2020) Identity-based provable data possession from RSA assumption for secure cloud storage. IEEE Trans Dependable Secure Comput 19(3):1753–1769

Salvakkam DB, Pamula R (2022a) MESSB–LWE: multi-extractable somewhere statistically binding and learning with error-based integrity and authentication for cloud storage. J Supercomput 78:1–30

Salvakkam DB, Pamula R (2022b) Design of fully homomorphic multikey encryption scheme for secured cloud access and storage environment. J Intell Inf Syst. https://doi.org/10.1007/s10844-022-00715-7

Sasikala C, Bindu CS (2019) Certificateless remote data integrity checking using lattices in cloud storage. Neural Comput Appl 31(5):1513–1519

Sen J (2015) Security and privacy issues in cloud computing. In: Cloud technology: concepts, methodologies, tools, and applications. IGI global, pp. 1585–1630

Singh G (2013) A study of encryption algorithms (RSA, DES, 3DES and AES) for information security. Int J Comput Appl 67(19):33–38

Tahat N, Alomari AK, Al-Hazaimeh OM, Al-Jamal MF (2020) An efficient self-certified multi-proxy signature scheme based on elliptic curve discrete logarithm problem. J Discrete Math Sci Cryptogr 23(4):935–948

Team TA (2006) AVISPA v1. 1 User manual. Information society technologies programme (June 2006), http://avispa-project.org

Xu R, Lang B (2015) A CP-ABE scheme with hidden policy and its application in cloud computing. Int J Cloud Comput 4(4):279–298

Xu Z, He D, Vijayakumar P, Choo KKR, Li L (2020) Efficient NTRU lattice-based certificateless signature scheme for medical cyber-physical systems. J Med Syst 44(5):1–8

Yang Q, Li D (2021) Provably secure lattice-based self-certified signature scheme. Secur Commun Netw 2021:1–9

Yu Y, Au MH, Ateniese G, Huang X, Susilo W, Dai Y, Min G (2016) Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage. IEEE Trans Inf Forensics Secur 12(4):767–778

Yu Y, Au MH, Ateniese G, Huang X, Susilo W, Dai Y, Min G (2017) Identity-based remote data integrity checking with perfect data privacy-preserving for cloud storage. IEEE Trans Inf Forensics Secur 12(4):767–778

Zhang J, Dong Q (2016) Efficient ID-based public auditing for the outsourced data in cloud storage. Inf Sci 343:1–14

Zhang Y, Xu C, Zhao J, Zhang X, Wen J (2015) Cryptanalysis of an integrity checking scheme for cloud data sharing. J Inf Secur Appl 23:68–73

Zhang Y et al (2015) A lattice-based designated verifier signature for cloud computing. Int J High Perform Comput Network 8(2):135–143

Zissis D, Lekkas D (2012) Addressing cloud computing security issues. Futur Gener Comput Syst 28(3):583–592

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.