International Journal of Scientific Research in Science, Engineering and Technology
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)
doi : https://doi.org/10.32628/IJSRSET

Enhanced Key Exchange Protocols to Maintain Data Integrity In Cloud Environments

Rajesh Gundla[1], Sunil Gupta[2], C. Srinivasa Kumar[3]

[1]Research Scholar, Department of CSE, Shridhar University, Pilani, Rajasthan, India
[2]Associate Professor, Department of Computer science and Engineering, Shridhar University, Pilani, Rajasthan, India
[3]Professor and Dean, Department of Computer science and Engineering, Vignan's Institute of Management and Technology for Women, Hyderabad, Telangana, India

Article Info
Volume 9, Issue 4

Page Number : 491-502

Publication Issue :
July-August-2022

Article History
Accepted : 06 July 2022
Published: 22 July 2022

ABSTRACT

This paper presents a cloud-based security and integrity framework that utilizes the KEP protocol, a three-party authentication protocol, to secure and authenticate data. The protocol includes a secure session key and auditor's authentication, and extensive documentation related to its implementation phase is available, including scientific experiments. Results of security tests, displayed in results, demonstrate that the KEP protocol is cost-effective and capable of withstanding various security concerns. Furthermore, it is significantly faster than existing methods in terms of computation time for authentication and encryption functions. Future work aims to increase the number of authentication points and further minimize computation time. Overall, the proposed KEP protocol provides sufficient safeguards against security breaches and is significantly less expensive than other alternatives.

Keywords: Key Exchange, Cloud Computing.

I. INTRODUCTION

The rise of cloud computing has created a huge amount of data that users can store and process. Due to the increasing number of users, the need for session key agreements has also become more prevalent. A simple and effective way to resolve these issues is by implementing multi-party or two-party authentication. To ensure that their data is protected from unauthorized access, many organizations have started using third-party Key Exchange protocols. These systems allow them to exchange key information in an unconstrained environment. The goal of the KEP protocol is to prevent unauthorized access to the data stored in your cloud. It uses an efficient one-way hash function and an ECC (ECC) to secure the data. In order to perform a proper analysis, it is recommended that you use the AVIPA (Automatic Identification and Protection of Protocols) method. Besides preventing unauthorized access, the protocol also helps minimize the risk of man-in-themiddle attacks and other key compromises.

II. Preliminaries

In Figure 1, we present a proposed method for securing communication within public networks using an enhanced three-party authentication system. Before we start talking about the details of the proposal, we must first identify the capabilities of the adversary. Since the KEP protocol is used in public communication, an attacker can potentially carry out arbitrary actions on the system. An attacker can potentially intercept the messages sent and received by the public channel by monitoring the conversations between the participants. The computational incapability of the password and identity of the users can be easily guessed by the attacker. An attacker can modify the encryption of intercepted transmissions if the message is altered. When a user’s account has been hacked, it is possible for an attacker to be a legitimate individual or a hacker who has access to the account. The proposed KEP protocol uses an efficient and secure algorithm for protecting data. It also provides various benefits such as confidentiality, authentication, and security.

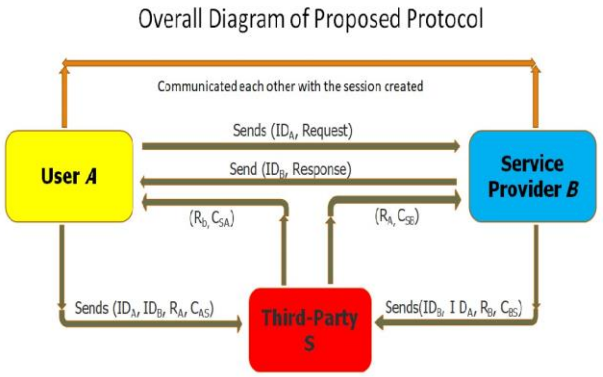

Overall Diagram of Proposed Protocol
Communicated each other with the session created
Sends (ID_A, Request)
Send (ID_B, Response)
User *A*
Service Provider *B*
(R_b, C_{SA})
(R_A, C_{SE})
Sends (ID_A, ID_B, R_A, C_{AS})
Third-Party S
Sends(ID_B, I D_A, R_B, C_{BS})

Figure 1: Overall Diagram of Proposed KEP Protocol.

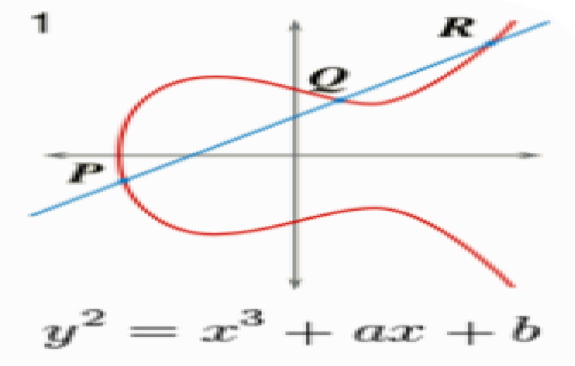
The proposed KEP protocol is divided into two phases. The first one is the System Initialization phase, while the second one is the Authenticated Key Exchange phase. The implementation of the KEP protocol is divided into two phases. The first one involves establishing the system and the second one involves the exchange of keys.

2.1 The startup step of the system

The System Initialization phase involves establishing the KEP system. The second phase involves the exchange of keys, which is the authentication phase. The third-party S initializes and chooses a variety of configuration options during the initialization phase. The cloud service provider and the user are also required to participate in order to register for the system. The proposed system is built on the hash function SHA-256 and the ECC. The third-party S that is involved in the creation of the system performs various configuration steps during the initialization phase. The operations of the ECC are defined by the elliptic curve known as y2. The variable a' and b are represented by different elliptic curves, which can be represented by the image. There are multiple points on the curve, and it has an infinity point. A public and a private key are respectively represented by a random number and a point on the curve. The generated public key can be multiplied by the generator point G to get the required result. The small size of the ECC's key is its most important advantage. It's believed that the 160-bit key is similar to the 1024-bit key used by RSA. It uses an encryption key of 164 bits long. This page aims to provide a simple overview of the elliptic curve cryptography principles. It also covers the various computational issues that are involved in implementing it. In order to understand the various features of the elliptic curve cryptography concept, let us assume that the given conjecture is a set of opinions on a prime field. The mod q value of the y2 conjecture is the sum of the mod values of the given conjectures.

$$y2 \bmod q = (x3 + ax + b) \bmod q \qquad (1)$$

where x, y, a, b ∈ Eq. In addition, we can assume that the two conjectures, Q and P, are both situated on the curve. A point subtraction method is also used to describe the relationship between the conjectures Q and P. If the value of Q is -P, then the line connecting the two conjectures will go through the curve O. A point doubling occurs when two points on a given curve are merged. For instance, if the 2P = Q conjecture is true, then the line 2P crosses the curve (1) at Q. Hashing (SHA-256). The use of the SHA256 algorithm is commonly used to store and distribute hash data. This method ensures that each piece of information has a unique identifier. If the data is sent over the network, then a hash function is used to generate a value for the intended recipient. After the two parties have registered, the third-party S uses the private and public keys generated by the two parties to perform authentication. The two entities that are registered as a cloud service provider and user are referred to as A and B, respectively. During the registration phase, they are asked to create their public and private keys through the server S.



$y^2 \bmod q = (x^3 + ax + b) \bmod q$ (2)

where x, y, a, b ∈ F_q

2.2 Phase of exchange of authenticated keys

As per the KEP Protocol procedure whenever User A wants to interact with Cloud Service Provider for service request, they need to undergo authentication process by Trusted Third-Party. It is necessary to authenticate each other over a public network in this stage. Three entities are involved: User A, Cloud Service Provider B, and a trusted server (Third-Party) S, which assists A and B in authenticating each other.

When a user wants to interact with a cloud service provider, they need to perform an authentication process through a third-party. This process is carried out through a public network. The third-party S is a server that coordinates the authentication process between the user and the cloud service provider.

2.3 The operation of KEP

The KEP system is composed of three steps. These include the registration of the user, the creation of the private and public keys, and the authentication process by the third-party S. The user must first register with a third-party to start interacting with a cloud service. This process can be done by sending a request to the third-party. According to the algorithm, the user will need to send two requests to the third-party to register. One of these requests is to the cloud service provider, while the other is to the trusted third party. The user must first register with a third party to start interacting with a cloud service. This process can be done by sending a request to the third-party. According to the algorithm, the user will need to send two requests to the third-party to register. One of these requests will be sent to the cloud service provider, while the other will be to the trusted third party. As per the algorithm, the cloud service provider will need to register at the trusted third party once for the first time. The third-party S handles the authentication process for cloud service providers. In accordance with the KEP Protocol, whenever a user wants to interact with the cloud service provider, they need an authentication process from the trusted third party. The details of the operational process are provided below.

The user will first send two requests to the cloud service provider. One of these requests is to the cloud service provider, while the other is to the third-party S. After receiving the request from the user, the third-party S will then check if the user is a valid user. It will then send the reply message to the user. The process for the cloud service provider B is similar. After the authentication process has been completed, the cloud provider will send a response message to the third party. The third party will then perform a test on the user to see if they are a valid user. When the user and the cloud service provider have been able to confirm their connection, a Session Key is generated. This will allow the two parties to communicate securely. For the protection of data confidentiality, we have utilized the elliptical curve cryptography technique. This method is based on the ECC algorithm and can provide more efficient keys. We have also utilized the SHA-256 algorithm for the authentication process.

3. The proposed KEP protocol's

Table 1. This work employs a variety of notation styles.

Notations	Meaning
A	The protocol participant (User)
B	The protocol participant (Cloud Service Provider)
S	The protocol participant (Third-Party)
k	The security parameter (Key)
Q	A large prime number of *k*-bit length and q>3
ID_x	The identity of communication party x
p, q	Two large primes
R	P + Q = R, where the line joining P and Q intersects the Curve
F_q	A field of prime order *q*
E_q (a, b)	A set of elliptic curve points of order *n*, where a, b ∈ F_q
Q	A base point of order *n* over E_q (a, b)
(d_i, U_i)	The private/public key pair of the entity *i*, where i = A, B, S, where di ∈ Z q * and Ui = di·Q
H (.)	One-way cryptographic hash function
S_K	Session Key
||	The message concatenation operator
(·)	The elliptic curve scalar point multiplication
A	The Adversary

3.1 Algorithm of KEP

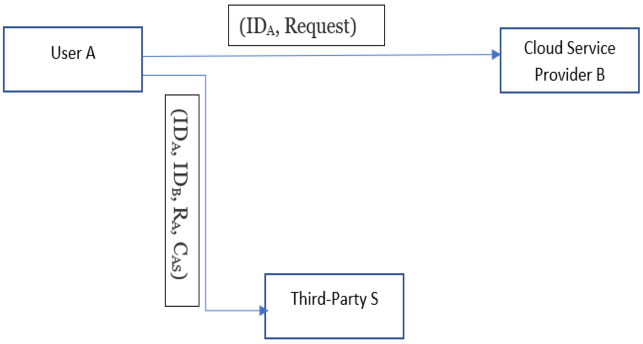


Figure 2: Flow Chart (User Registration)

The KEP algorithm is presented in three phases. The first one is the registration process for user A. Figure 2 shows the flow chart of the proposed algorithm. The flow chart shows the various steps involved in the registration process for user A. The pseudo code for the registration algorithm is provided below.

Algorithm 1 (User Registration)

```
Step 1
        Pick an integer $r_A \in Z^*q$ randomly and then computes
        $H_A = H(r_A || d_A)$ and $R_A = H_A \cdot Q$.
Step 2
        Then compute $K_A = d_A \cdot U_S = d_A \cdot d_S \cdot Q$ and
        $C_{AS} = H(ID_A || ID_B || R_A || K_A)$.
Step 3
        Send ($ID_A$, Request) to $B$ and ($ID_A$, $ID_B$, $R_A$, $C_{AS}$) to $S$.
```

The KEP registration algorithm provides a flow chart that illustrates the various steps in the process of establishing a user's identity. When users want to interact with a cloud service provider for the first time, they need to first register with the third-party. According to the KEP algorithm, users will need to send two requests in order to register with the third-party. One of these is to the cloud provider, while the second is to the trusted third-party. The second round of the KEP registration process is shown in Figure 3. This involves the registration process for cloud service providers.

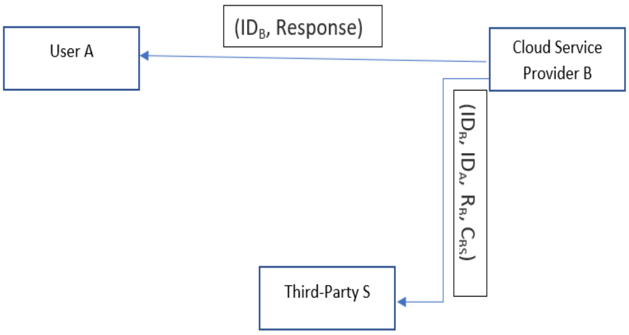
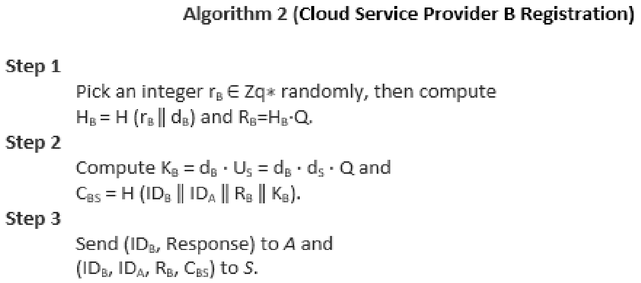


Figure 3: Flow Chart (Cloud Service Provider B Registration)

The flow chart below illustrates the various steps involved in the registration process for cloud service providers. The pseudo code for the B registration algorithm is provided below. In this round, B completes the following steps. As per the algorithm, a cloud service provider must first register with the trusted third party after an interaction. After receiving A's message, B will then send a registration request to the third party. The third round of the KEP authentication process is shown in Figure 4. This involves the third-party's algorithm for authentication.



Algorithm 2 (Cloud Service Provider B Registration)

```
Step 1
    Pick an integer $r_B \in Zq*$ randomly, then compute
    $H_B = H (r_B \| d_B)$ and $R_B=H_B \cdot Q$.
Step 2
    Compute $K_B = d_B \cdot U_S = d_B \cdot d_S \cdot Q$ and
    $C_{BS} = H (ID_B \| ID_A \| R_B \| K_B)$.
Step 3
    Send ($ID_B$, Response) to A and
    ($ID_B$, $ID_A$, $R_B$, $C_{BS}$) to S.
```

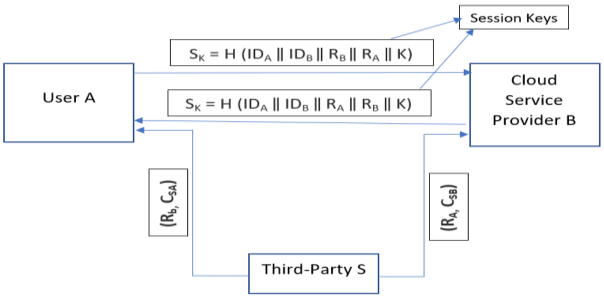
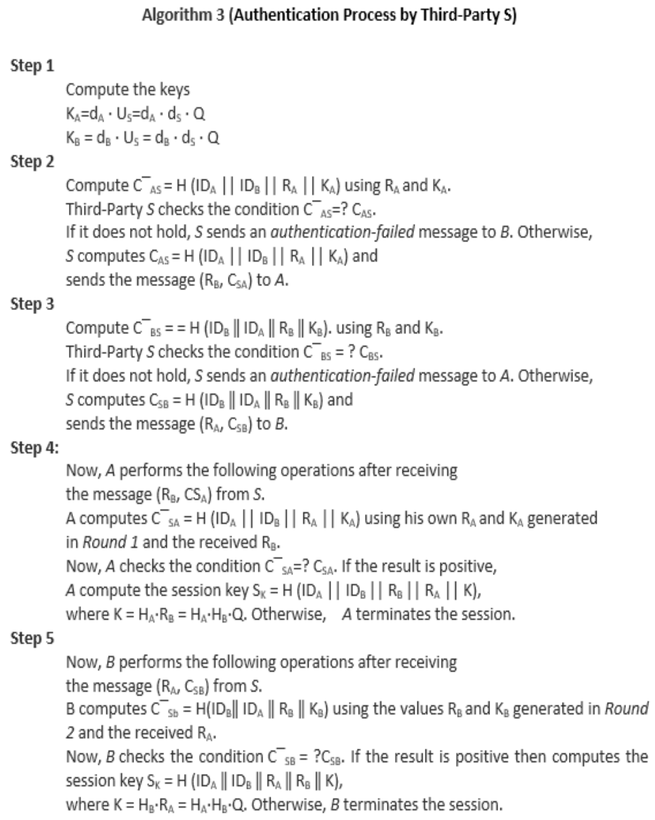

Session Keys
$S_K = H (ID_A \| ID_B \| R_B \| R_A \| K)$
User A
Cloud Service Provider B
$S_K = H (ID_A \| ID_B \| R_A \| R_B \| K)$
(R_b, C_{SA})
(R_A, C_{SB})
Third-Party S

Figure 4: Flow Chart (Authentication Process by Third-Party S)

The flow chart below illustrates the various steps in the process of establishing a user's identity with the help of the third-party's authentication process. The S can perform the following tasks after A and B have sent you their IDB, IDB, RB, and CBS. The characterization of the third-party authentication process is shown in the following diagram. The KEP protocol ensures that whenever a user wants to interact with a certain cloud service provider, they must undergo the authentication process initiated by the trusted third party. For instance, user A will send two requests to the cloud provider: one for the IDB, and another for the registration request. fter receiving the request from user A, the S will determine if the user is a valid user or not. It will then send the reply message to the user based on the CAS value. For instance, if user A is a valid user, the cloud service provider will send a reply message to the user after completing the registration process and verifying its authenticity. The third-party will then perform a test on the user to determine if it is a valid user. After this, a session key will be calculated based on the data collected during the authentication process.

The protocol should implement various security attributes to ensure that the user's identity is protected. The protocol's known-key security ensures that only the two individuals who are authorized to access the user's account are able to generate a private session key. These sessions' unique attributes are different from those generated during the subsequent executions. It's important to note that if one of these is compromised, the other session keys should not be affected. The secret keys of the server, the third party, and the user should not be affected by a security breach in the current session. A resilient network is built against key-compromise impersonation. An adversary can't access one of the parties' secrets and corrupt it, which means they can't perform impersonation on the other one. A vulnerability in the security of the protocol can occur when one of the two parties believes that they share a key, but the other one doesn't. This is an example of a key-sharing misconfiguration. To prevent this type of attack, the protocol should implement strong security measures. One of the most important factors that should be considered when it comes to implementing the protocol's security measures is the cooperation between the two parties. Even the server doesn't have a good idea what to do when it comes to generating a session key. This tutorial introduces a simulation of the KEP protocol that uses the AVISPA tool for formal security verification. This section thoroughly describes the steps involved in carrying out security verification of the KEP method. The protocol was subjected to various active and passive threats.



```
Algorithm 3 (Authentication Process by Third-Party S)

Step 1
    Compute the keys
    $K_A = d_A \cdot U_S = d_A \cdot d_S \cdot Q$
    $K_B = d_B \cdot U_S = d_B \cdot d_S \cdot Q$
Step 2
    Compute $\bar{C}_{AS} = H (ID_A \,||\, ID_B \,||\, R_A \,||\, K_A)$ using $R_A$ and $K_A$.
    Third-Party S checks the condition $\bar{C}_{AS} =? C_{AS}$.
    If it does not hold, S sends an authentication-failed message to B. Otherwise,
    S computes $C_{AS} = H (ID_A \,||\, ID_B \,||\, R_A \,||\, K_A)$ and
    sends the message $(R_B, C_{SA})$ to A.
Step 3
    Compute $\bar{C}_{BS} == H (ID_B \| ID_A \| R_B \| K_B)$. using $R_B$ and $K_B$.
    Third-Party S checks the condition $\bar{C}_{BS} = ? C_{BS}$.
    If it does not hold, S sends an authentication-failed message to A. Otherwise,
    S computes $C_{SB} = H (ID_B \| ID_A \| R_B \| K_B)$ and
    sends the message $(R_A, C_{SB})$ to B.
Step 4:
    Now, A performs the following operations after receiving
    the message $(R_B, CS_A)$ from S.
    A computes $\bar{C}_{SA} = H (ID_A \,||\, ID_B \,||\, R_A \,||\, K_A)$ using his own $R_A$ and $K_A$ generated
    in Round 1 and the received $R_B$.
    Now, A checks the condition $\bar{C}_{SA} =? C_{SA}$. If the result is positive,
    A compute the session key $S_K = H (ID_A \,||\, ID_B \,||\, R_B \,||\, R_A \,||\, K)$,
    where $K = H_A \cdot R_B = H_A \cdot H_B \cdot Q$. Otherwise, A terminates the session.
Step 5
    Now, B performs the following operations after receiving
    the message $(R_A, C_{SB})$ from S.
    B computes $\bar{C}_{Sb} = H(ID_B \| ID_A \| R_B \| K_B)$ using the values $R_B$ and $K_B$ generated in Round
    2 and the received $R_A$.
    Now, B checks the condition $\bar{C}_{SB} = ?C_{SB}$. If the result is positive then computes the
    session key $S_K = H (ID_A \| ID_B \| R_A \| R_B \| K)$,
    where $K = H_B \cdot R_A = H_A \cdot H_B \cdot Q$. Otherwise, B terminates the session.
```

3.2 Implementation

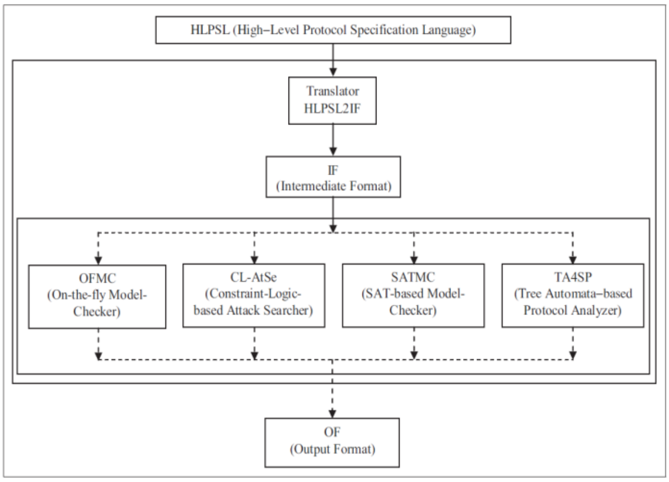

HLPSL (High–Level Protocol Specification Language)
Translator
HLPSL2IF
IF
(Intermediate Format)
OFMC
(On-the-fly Model-Checker)
CL-AtSe
(Constraint-Logic-based Attack Searcher)
SATMC
(SAT-based Model-Checker)
TA4SP
(Tree Automata–based Protocol Analyzer)
OF
(Output Format)

Figure 5: AVISPA Tool Architecture

The results of the simulation are summarized below. It is recommended to use the tool if the system is secure against both active and passive attacks. The evaluation tool known as AVISPA is designed to analyze cyber-security systems and provide a security assessment if they meet the UN's standards. The original implementation of the security verification technique used a symbol-based model known as the On-the-Fly model. This method was used to investigate the topic subspace of the protocol. The second component, the CL-Atse, is then used to check if the model has the necessary predicate formulations. The tree automaton returns the expected information of the suspect in the final back-end. Additional developers can benefit from the use of the HLPSL-2IF converter, which is a tool that allows them to fill the void in the Intermediate Format between the demands of another program and the demands of HLPSL. The process is carried out by the servers of AVISPA. Every piece of artwork follows a set of laws, and each principle has specific duties. This book provides a comprehensive overview of the various aspects of the ETP–AKE system. The goal and setting of a session are explained using the HLPSL code system. Following that, examples of the points made are shown.

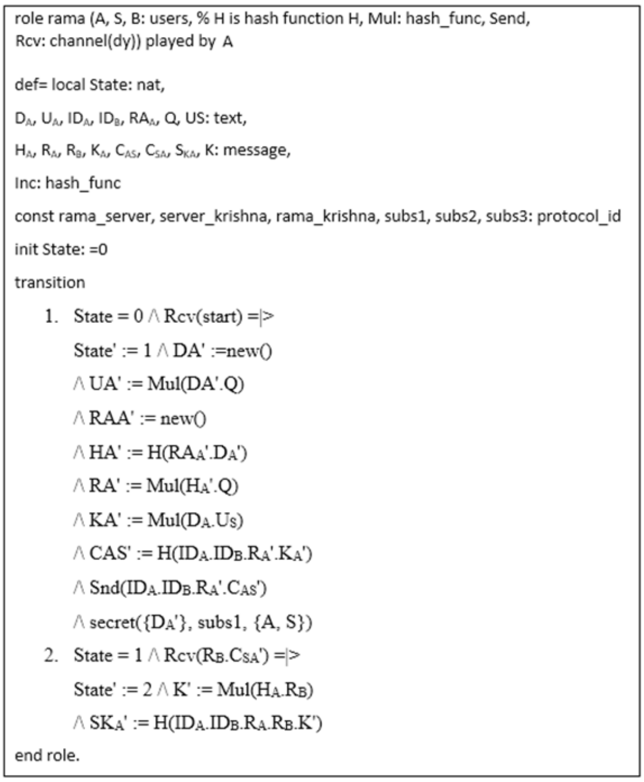
The two back-end platforms used to develop the CL-Atse and OFMC protocols were utilized to create a secure replica of the KEP framework. The researchers found that it is very safe to implement the protocol against both active and passive threats. In addition, they noted that the use of patterns in this part can help minimize the risks associated with security breaches.

In this tutorial, we introduce the three parties that are involved in the development of the KEP protocol. These include the User A, the Cloud Service Provider B, and the Trustworthy Third-party S.

In the HLPSL language, User A is the central component of the protocol. The third-party S provides the private and public key pairs to A, and it then sends it to the user. This method results in a random number being calculated and sent to the user. According to the declaration, the private key is only used by the users and is only accessible to them by the trusted third party. In transition 2, the user receives a message from the S, and it then computes its session key.

User A is the central component of the protocol. The third-party S provides the public and private key pairs to A, and it then sends it to the user. This method results in a random number being calculated and sent to the user. According to the declaration, the private key is only used by the users, and it is only accessible to them by the trusted third party. In transition 2, the user receives a message from the S, and it then computes its session key.

The authentication and registration process of the cloud service provider is similar to that of the user registration process. The source code for this is also large. The implementation phase successfully completed the objectives and introduced two authentication methods. The public and private keys of A and B are kept secret. The server's private key is also kept private. The user's authentication process involves creating a random number that is only A's knowledge about. If the S is successful in getting it, it will verify the authenticity of the user. The S secures B by producing a random number that's only known to it and sending it to it.



```
role rama (A, S, B: users, % H is hash function H, Mul: hash_func, Send,
Rcv: channel(dy)) played by A

def= local State: nat,

DA, UA, IDA, IDB, RAA, Q, US: text,

HA, RA, RB, KA, CAS, CSA, SKA, K: message,

Inc: hash_func

const rama_server, server_krishna, rama_krishna, subs1, subs2, subs3: protocol_id

init State: =0

transition
    1.  State = 0 ∧ Rcv(start) =|>
        State' := 1 ∧ DA' :=new()
        ∧ UA' := Mul(DA'.Q)
        ∧ RAA' := new()
        ∧ HA' := H(RAA'.DA')
        ∧ RA' := Mul(HA'.Q)
        ∧ KA' := Mul(DA.US)
        ∧ CAS' := H(IDA.IDB.RA'.KA')
        ∧ Snd(IDA.IDB.RA'.CAS')
        ∧ secret({DA'}, subs1, {A, S})
    2.  State = 1 ∧ Rcv(RB.CSA') =|>
        State' := 2 ∧ K' := Mul(HA.RB)
        ∧ SKA' := H(IDA.IDB.RA.RB.K')
end role.
```

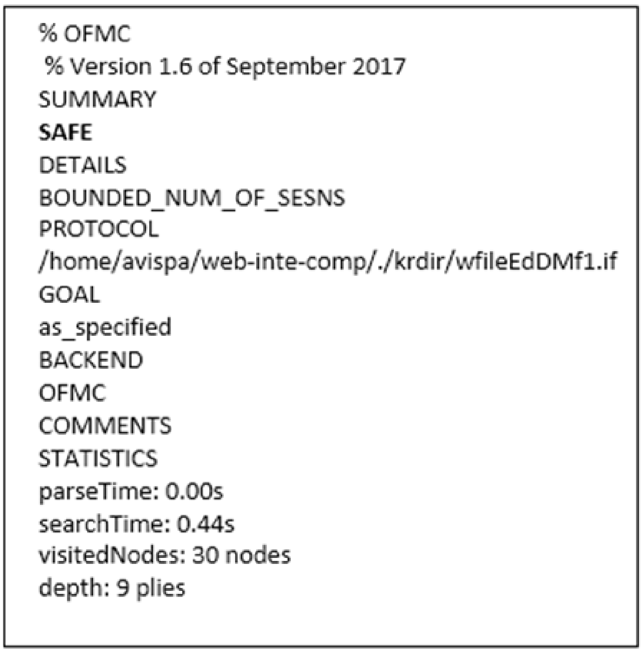
4. Experimentation

This part of the study evaluates and contrasts the various KEP methods currently in use. We also took into account the security concerns and costs associated with each implementation. Although the schemes are generally efficient when it comes to total rounds, they are not ideal for practical use due to their weaknesses. The results of the experiment were analyzed by the two different simulation tools, the AviSPA and the MATLAB.

4.1 Experimental Setup:

The experiments were performed on an Intel Core i5 with 16 GB of RAM. The simulations were then replicated using the AviSPA tool. We were able to show that the proposed KEP method is secure against both passive and active attacks. The registration process is only one of the steps in the process of securing a user's identity. It involves entering a password and a key. The user's account is then subjected to various levels of verification. Before the implementation is deployed, a security test is conducted to ensure that the system is secure. An extensive security test was performed to ensure that the KEP system is secure. Although the exact details of the security test are not known, it was performed in order to minimize the impact of theoretical issues.

The experiment used the OFMC tool in the AviSPA simulator to perform security verification for the proposed KEP protocols. The results of the security test were then analyzed and transformed into high-level protocol specifications for the various roles in the KEP system. These include the user, the cloud service provider, and the trusted third-party. After the conversion has been completed, the HLPSL code will be imported into the tool AVISPA. The tool will then convert it into an intermediate format. The proposed KEP protocols will then be subjected to a security test against predefined threats. This was done using the help of two different back-end tools, namely the CL-Atse and OFMC. This is the execution process of the proposed KEP protocol that was performed using the OFMC tool. The KEP experiment was performed with the help of the OFMC tool in the AviSPA simulator. In addition to the protocol, the experiment also performed security verification on other protocols in the AviSPA simulator.



```
% OFMC
 % Version 1.6 of September 2017
SUMMARY
SAFE
DETAILS
BOUNDED_NUM_OF_SESNS
PROTOCOL
/home/avispa/web-inte-comp/./krdir/wfileEdDMf1.if
GOAL
as_specified
BACKEND
OFMC
COMMENTS
STATISTICS
parseTime: 0.00s
searchTime: 0.44s
visitedNodes: 30 nodes
depth: 9 plies
```

This research aimed to compare and analyze the various KEP methods currently in use. In addition to the security features, the computational costs of each procedure were also taken into account. Since the algorithms being used for encryption are designed to perform better than their predecessors, the computational costs of each method should be compared.

4.2 Results analysis and Discussion:

Protocols proposed by various authors and Protection against treats

Security threats	Yang and Chang	Pu et al.	Tan 1	Tan 2	He et al.	Hafizul et al
Temporary session-specific information that is known	✗	✗	✗	✗	✗	✗
Provide known key security	✓	✓	✓	✓	✓	✓
Provide key control	✓	✓	✓	✓	✓	✓
Man-in -the -middle attack	✗	✗	✗	✓	✓	✓
Key offset attack	✓	✗	✓	✓	✓	✓
Impersonation-of-initiator attack	✗	✗	✗	✓	✓	✓
Free from clock synchronization	✓	✓	✗	✗	✗	✗
Ensure utmost confidentiality going forward.	✓	✓	✓	✓	✓	✓
Attacks based on the impersonation of a responder	✗	✗	✗	✓	✓	✓
An impersonation attack on a compromised key	✓	✓	✓	✓	✓	✓
An assault on many sessions in progress simultaneously	✗	✓		✓	✓	✓
An assault on an unidentified key share	✗	✓	✓	✓	✓	✓

Table 2: Results of security comparisons between the proposed protocol and other comparable protocols achieved using the AVISPA simulator tool

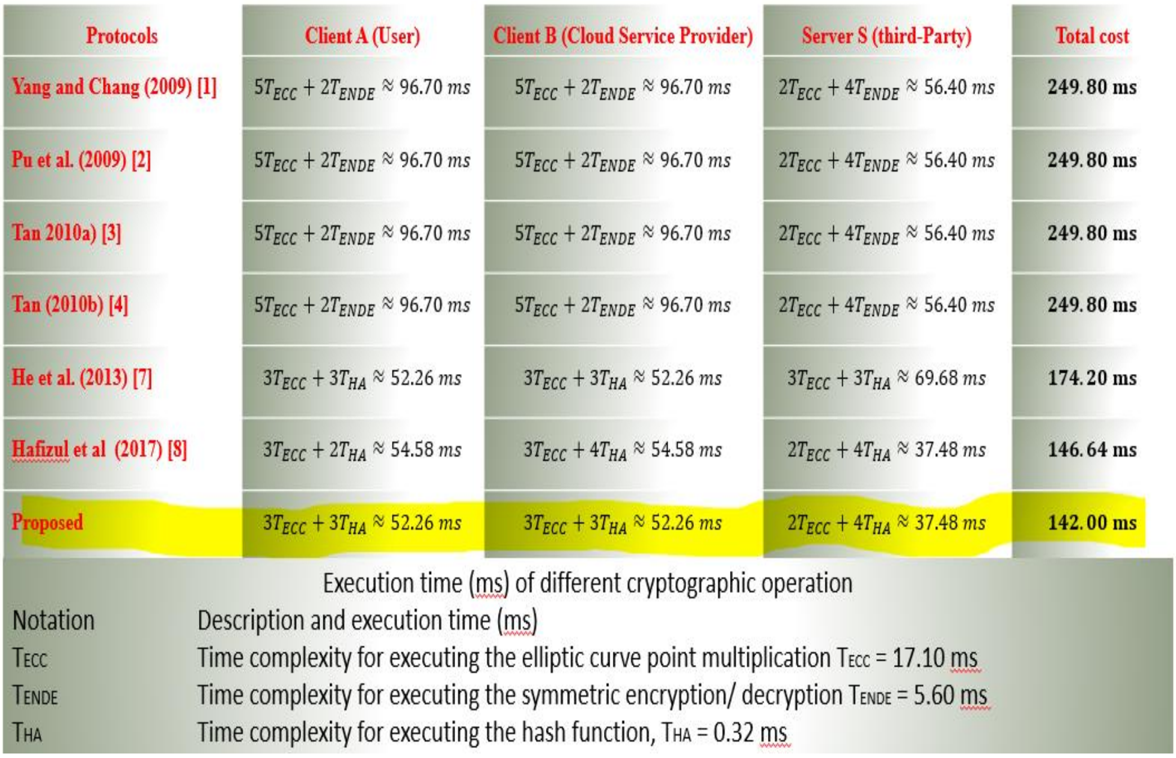
The security of the proposed protocol was evaluated against the existing protocols in the AVISPA simulator. Although the proposed protocol was able to successfully perform against all the threats, the other protocols failed to protect against all the attacks.

4.3 Comparison of proposed protocol's computation costs (MS) with those of similar existing protocols:

To evaluate the effectiveness of the proposed system, the computational efficiency was compared with that of previous cryptographic tests. Data can be encoded and decrypted using mathematical notations. The security test was performed on a four-core, 3.2 GHz computer with 16 GB of RAM. The security test was performed using a simulation created by the researchers, which they referred to as a MATLAB simulator. The suggested protocol had a lower computational cost than the others tested.

Table 3: Comparison of the proposed protocol's computation costs (MS) with the costs of similar current

protocols.



Protocols	Client A (User)	Client B (Cloud Service Provider)	Server S (third-Party)	Total cost
Yang and Chang (2009) [1]	$5T_{ECC} + 2T_{ENDE} \approx 96.70\ ms$	$5T_{ECC} + 2T_{ENDE} \approx 96.70\ ms$	$2T_{ECC} + 4T_{ENDE} \approx 56.40\ ms$	**249.80 ms**
Pu et al. (2009) [2]	$5T_{ECC} + 2T_{ENDE} \approx 96.70\ ms$	$5T_{ECC} + 2T_{ENDE} \approx 96.70\ ms$	$2T_{ECC} + 4T_{ENDE} \approx 56.40\ ms$	**249.80 ms**
Tan 2010a) [3]	$5T_{ECC} + 2T_{ENDE} \approx 96.70\ ms$	$5T_{ECC} + 2T_{ENDE} \approx 96.70\ ms$	$2T_{ECC} + 4T_{ENDE} \approx 56.40\ ms$	**249.80 ms**
Tan (2010b) [4]	$5T_{ECC} + 2T_{ENDE} \approx 96.70\ ms$	$5T_{ECC} + 2T_{ENDE} \approx 96.70\ ms$	$2T_{ECC} + 4T_{ENDE} \approx 56.40\ ms$	**249.80 ms**
He et al. (2013) [7]	$3T_{ECC} + 3T_{HA} \approx 52.26\ ms$	$3T_{ECC} + 3T_{HA} \approx 52.26\ ms$	$3T_{ECC} + 3T_{HA} \approx 69.68\ ms$	**174.20 ms**
Hafizul et al (2017) [8]	$3T_{ECC} + 2T_{HA} \approx 54.58\ ms$	$3T_{ECC} + 4T_{HA} \approx 54.58\ ms$	$2T_{ECC} + 4T_{HA} \approx 37.48\ ms$	**146.64 ms**
Proposed	$3T_{ECC} + 3T_{HA} \approx 52.26\ ms$	$3T_{ECC} + 3T_{HA} \approx 52.26\ ms$	$2T_{ECC} + 4T_{HA} \approx 37.48\ ms$	**142.00 ms**

Execution time (ms) of different cryptographic operation

Notation	Description and execution time (ms)
TECC	Time complexity for executing the elliptic curve point multiplication TECC = 17.10 ms
TENDE	Time complexity for executing the symmetric encryption/ decryption TENDE = 5.60 ms
THA	Time complexity for executing the hash function, THA = 0.32 ms

Execution time (in milliseconds) of various cryptographic procedures	
Notation	Duration of the description and implementation (ms)
THA	Executing the hash function has a high time complexity. , THA = 0.32 ms
TENDE	Executing the symmetric encryption/decryption procedure has a high time complexity. TENDE = 5.60 ms
TECC	The elliptic curve point multiplication takes a lot of time to complete. TECC = 17.10 ms

Table 4: Execution time (in milliseconds) of various cryptographic procedures.

It has been widely believed that symmetric encryption and decryption perform at least a hundred times faster than their asymmetric counterparts. This has led to the development of various low-cost alternatives such as XOR, which are often used in combination with other methods. In terms of efficiency and speed, symmetric key encryption is significantly faster than conventional key encryption.

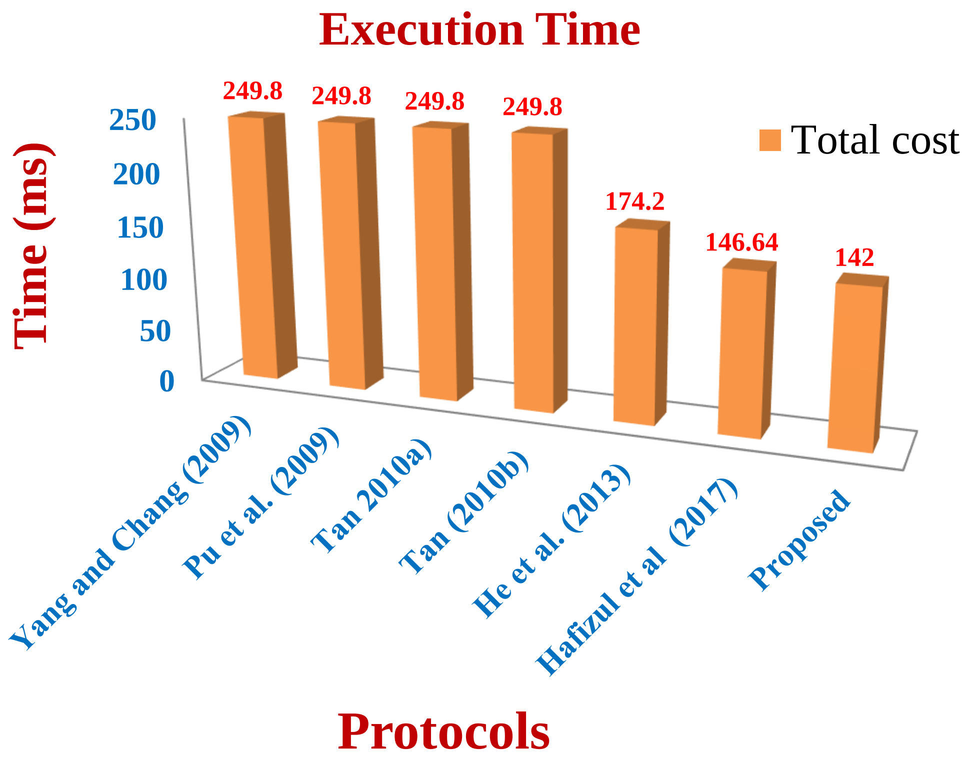


Figure 6: Timing (ms) Protocol comparison with similar protocols

The time it takes to perform one-way hash algorithms and cryptographic approaches is generally around 0.0005 seconds and 0.0087 seconds, respectively. To ensure the security of the proposed protocol, cloud security measures should be implemented. The timing of the proposed KEP protocol compared with the existing protocols is shown in Figure 6. The Y-axis shows the computation time it takes to calculate a session's time.

III. CONCLUSION

The results of the security test were shown in Table 3. The suggested KEP protocol was found to be cost-effective. Through extensive testing, it was able to successfully withstand various security concerns. The computation of the session key included the time it takes to perform authentication and encryption functions. The results of the security test indicated that the proposed protocol is significantly faster than the existing methods. In the next contribution, we will explore the possibility of increasing the number of authentication points and minimizing the computation time for encryption. Sufficient safeguards against various security breaches can be expected from the KEP protocol. The study's analysis revealed that the proposed KEP algorithm is significantly less expensive than the other alternatives.

IV. REFERENCES

[1]. Anbuchelian, S.; Sowmya, C.M.; Ramesh, C. Efficient and secure auditing scheme for privacy preserving data storage in cloud. Clust. Comput. 2019, 22, 9767–9775.

[2]. Babu, S. Dilli, and Rajendra Pamula. "An effective block-chain based authentication technique for cloud based IoT." Advances in Computing and Data Sciences: 4th International Conference, ICACDS 2020, Valletta, Malta, April 24–25, 2020, Revised Selected Papers 4. Springer Singapore, 2020.

[3]. Domingo-Ferrer, J.; Farras, O.; Ribes-Gonlez, J.; Sánchez, D. Privacy-preserving cloud computing on sensitive data: A survey of methods, products and challenges. Comput. Commun. 2019, 140, 38–60.

[4]. Dunne, N.J.; Brennan, N.M.; Kirwan, C.E. Impression management and Big Four auditors: Scrutiny at a public inquiry. Account. Organ. Soc. 2021, 88, 101170.

[5]. Girma, A.; Garuba, M.; Li, J. Analysis of Security Vulnerabilities of Cloud Computing Environment Service Models and Its Main Characteristics. In Proceedings of the 2015 12th International Conference on Information Technology-New Generations, Las Vegas, NV, USA, 13–15 April 2015; pp. 206–211.

[6]. Hussien, Z.A.; Jin, H.; Abduljabbar, Z.A.; Yassin, A.A.; Hussain, M.A.; Abbdal, S.H.; Zou, D. Public auditing for secure data storage in cloud through a third-party auditor using modern ciphertext. In Proceedings of the IEEE 2015 11th International Conference on Information Assurance and Security (IAS), Marrakech, Morocco, 14–16 December 2015.

[7]. Karthiban, K.; Smys, S. Privacy preserving approaches in cloud computing. In Proceedings of the IEEE 2018 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 19–20 January 2018; pp. 462–467.

[8]. Pavithra, S.; Thangadurai, E.; Mailsamy, M. Secure Data Storage in Cloud using Code Regeneration and public audition. Int. J. Emerg. Technol. Comput. Sci. Electron. 2016, 20, 65–68.

[9]. Perez-Botero, D.; Szefer, J.; Lee, R.B. Characterizing hypervisor vulnerabilities in cloud computing servers. In Proceedings of the ACM 2013 International Workshop on Security in Cloud Computing, Dresden, Germany, 9–12 December 2013; pp. 3–10.

[10]. Razaque, A.; Amsaad, F.; Hariri, S.; Almasri, M.; Rizvi, S.S.; Frej, M.B.H. Enhanced grey risk assessment model for support of cloud service provider. IEEE Access 2020, 8, 80812–80826.

[11]. Razaque, A.; Nadimpalli, S.S.V.; Vommina, S.; Atukuri, D.K.; Reddy, D.N.; Anne, P.; Vegi, D.; Malllapu, V.S. Secure data sharing in multi-clouds. In Proceedings of the IEEE 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, India, 3–5 March 2016; pp. 1909–1913.

[12]. Razaque, A.; Rizvi, S.S. Privacy preserving model: A new scheme for auditing cloud stakeholders. J. Cloud Comput. 2017, 6, 7.

[13]. Ru, J.; Yang, Y.; Grundy, J.; Keung, J.; Hao, L. A systematic review of scheduling approaches on multi-tenancy cloud platforms. Inf. Softw. Technol. 2020, 132, 106478. 19. Albugmi, A.; Alassafi, M.O.; Walters, R.; Wills, G. Data

security in cloud computing. In Proceedings of the 2016 Fifth International Conference on Future Generation Communication Technologies (FGCT), London, UK, 17–19 August 2016; pp. 55–59.

[14]. Shakarami, A.; Ghobaei-Arani, M.; Shahidinejad, A.; Masdari, M.; Shakarami, H. Data replication schemes in cloud computing: A survey. Clust. Comput. 2021, 24, 2545–2579.

[15]. Shen, W.; Yu, J.; Xia, H.; Zhang, H.; Lu, X.; Hao, R. Light-weight and privacy-preserving secure cloud auditing scheme for group users via the third-party medium. J. Netw. Comput. Appl. 2017, 82, 56–64.

[16]. Shrinivas, D. Privacy-preserving public auditing in cloud storage security. Int. J. Comput. Sci. Nad Inf. Technol. 2011, 2, 2691–2693.

[17]. Wang, B.; Li, B.; Li, H. Knox: Privacy-preserving auditing for shared data with large groups in the cloud. In Proceedings of the International Conference on Applied Cryptography and Network Security, Kamakura, Japan, 21–24 June 2012; Springer: Berlin/Heidelberg, Germany, 2012.

[18]. Wang, B.; Li, B.; Li, H. Panda: Public auditing for shared data with efficient user revocation in the cloud. IEEE Trans. Serv. Comput. 2013, 8, 92–106.

[19]. Wei, L.; Zhu, H.; Cao, Z.; Dong, X.; Jia, W.; Chen, Y.; Vasilakos, A.V. Security and privacy for storage and computation in cloud computing. Inf. Sci. 2014, 258, 371–386.

[20]. Worku, S.G.; Xu, C.; Zhao, J.; He, X. Secure and efficient privacy-preserving public auditing scheme for cloud storage. Comput. Electr. Eng. 2014, 40, 1703–1713.

Cite this article as :

Rajesh Gundla, Sunil Gupta, C. Srinivasa Kumar, "Enhanced Key Exchange Protocols to Maintain Data Integrity In Cloud Environments", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 9 Issue 4, pp. 491-502, July-August 2022.

Journal URL : https://ijsrset.com/IJSRSET2022