Efficient and Verified Authentication Scheme for Cloud Computing.

Mohamed Mahmoud Zarad, Ahmed Ali Abd El-Hafez, Ismail Mohamed Hafez

Abstract: Cloud computing is a new style of computing in which dynamically scalable and virtualized services and resources are provided. Authentication processes in cloud computing represents a vital and critical factor that affects the cloud performance and availability. The adoption of cloud computing suffers from security and privacy deficiencies which became significant challenge. In this paper; an efficient and provably secure authentication mechanism is proposed to give a legitimate user the right to access and manage the cloud resources and services. The proposed authentication mechanism takes into consideration the resiliency against the most known security attacks on cloud computing. Further, it adopts the elliptic curve algorithm for authentication process taking the advantage of its small key size which decreases the estimated delay for authentication process. In order to assess the security performance of our proposed model, we provide a detailed verification process for the proposed model using Scyther tool which gives a detailed description for the proposed security protocol compared with a previously proposed protocol that uses Diffie-Hellman algorithm for the authentication process. The main advantage of using Scyther for verifying security protocols is that it provides a detailed track for any potential attack which helps security experts to design strong security architecture for their systems. The verification results presented in this paper show that the proposed scheme is resilient against all known attacks on cloud authentication

I. INTRODUCTION

Cloud computing is a new technology which provides a convenient way for accessing configurable resources among users (e.g., networks, servers, services and applications) by using the concept of virtualization, storage connectivity and processing power[1]. Security in cloud computing protects the rights of each user in using the cloud services and resources and protects the sensitive data of each user in the cloud from attacks or breaching. So there are some vulnerabilities and attacks that face the users of the cloud which hinder an effective usage of the cloud resources and services; and deprive the service provider from efficient delivery of resources and services to the users and customers. These attacks must be known to be able to build an efficient security architecture for cloud computing. **In** this work, an authentication scheme and key agreement protocol using Elliptic curve Diffie-Hellman will be introduced showing how to face these attacks. There are many reasons that lead to the using of Elliptic curve Diffie Hellman for authentication of cloud computing compared with RSA & Diffie-Hellman. The following sections will introduce a comparison between other security protocols from the key size point this comparison shows how elliptic curve introduce the same security level compared to other protocols with the same key size.We note that the work presented in this paper was published in 18 May in IJCA volume 42 number 8 meanwhile the verification and testing for the proposed scheme will be presented solely in the current manuscript.

A. Account Hijacking

Account hijacking means compromising the cloud legitimate users credentials and using them for illegal purposes [2]. With stealing users credentials attackers can compromise the confidentiality, integrity and availability of cloud services and resources.

B. Internet protocol vulnerabilities

Since the cloud services and resources are available anywhere .Theses services and resources are accessed via public networks such as internet which considered untrusted [3]. So internet protocol vulnerabilities as Man in The Middle Attack and IP spoofing will hinder the performance of the cloud.

C. Data recovery vulnerability

The cloud characteristics of pooling and elasticity entail that resources allocated to one user will be reallocated to a different user at later time [4]. For memory or storage resources it might be possible to recover data written by a previous user.

D. Shared Technology Vulnerabilities

The cloud characteristics of sharing of many services and resources provides multi tenancy, so underlying components that support cloud infrastructure may not be designed to offer strong and proper isolation properties for multi tenant architecture or multi customer applications [5]. This can lead to shared technology vulnerabilities which are related to virtual machines and operating systems.

E. Malware injection attack

These types of attacks are web based attacks in which hackers exploit vulnerabilities into web application and embed a malicious code which change the behavior of its normal execution [6].

F. Related Work

In 2017 A Two-Factor RSA-Based Robust Authentication scheme for cloud environment was introduced by Ruhul Amin using RSA as authentication protocol and using AVISPA as simulation tool for verifying the introduced authentication protocol. B y comparing the proposed

model with the introduced one it was found that using elliptic curve as authentication and key exchange protocol is more efficient than using RSA due to the smaller key size of elliptic curve which gives the same security level of RSA using the same key with larger size which reduce the computing power of authenticating process [7].

In Mar 2017 a secure and efficient authentication scheme using one time password was introduced by Chung-Huei Ling using one time password for authenticating users and customers to cloud environment [8]. The introduced scheme uses password for authentication which represent a weak parameters due the presence of supercomputers which has a very high computing power for estimating and guessing users' password and credentials in addition to the problem of storing users' password which represent a single point of failure if the storage server has be stolen or penetrated. In the proposed scheme the users' password and all credentials are encrypted with AES-256 for protection against any stealing or modification.

In 2016 a provably secure authentication scheme was introduced by Mohammad Wazid using biometric parameters for authenticating users and customers to the cloud environment. Using biometric parameters for authentications is not accurate 100% due to the effect of environmental factors and the requirement of extra hardware to be implemented which increase the computational power of the system decreasing the system performance. due to increasing the estimated time of authentication process [9].By comparing the introduced scheme with the proposed one it is obvious that using encrypted passwords with encrypted timestamps will be more efficient than using biometric parameters due to the speed of authentication process without accessing extra hardware for authentication.

In 2014 an authentication scheme was introduced by Faraz fatema *et al* [10] depending on the concept of agents. Two agents were used in this scheme: client based user authentication agent (CUA): It is an extension that installed in end user web browser to confirm the identity of user before accessing cloud servers[10]. Modified Diffie Hellman agent (MDHA) another agent used for accessing cloud servers with unregistered devices. By using MDHA temporary access permission has been provided for user for accessing from unregistered device. This model uses AES-256 and RSA-2048 during authentication and before Storing data to the cloud servers which improves the rate of trust in the framework. In 2013; Iman Ghavam *et. al*. [11] introduce an authentication scheme including two main steps: client based encryption algorithm for encrypting data before uploading to cloud servers and user authentication secure key exchange algorithm for validating user legal identities and control acquiring services from the cloud. According to this model RSA small e algorithm has been chosen for encryption process by using public exponent much smaller than **q(n).** Using small exponent will decrease the effective cost of encryption process. The main drawbacks of Diffie Hellman &RSA Algorithms are the slower processing time and the requirement of high storage capacity of 1024 bits.In the following table a key size comparison among the elliptic curve, RSA and Diffie- Hellman will be showed (table-1):

Symmetric key size (bits)	RSA and Diffie-Hellman key size (bits)	Elliptic curve key size (bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

Table -1
(Comparison between Elliptic curve, RSA and Diffie Hellman)

From the previous table it is obvious that the main advantages of using Elliptic Curve Diffie-Hellamn as an authentication scheme is the smaller key size. This helps in that the implementation of the encryption and decryption process will be much faster than of that of RSA as 1024bits key size of RSA is equivalent to 160 bits of ECC. Consequently, the use of Elliptic Curves can be more efficient and reliable than RSA. Otherwise the implementation of RSA is easier than that of Elliptic Curve.

II. Proposed model

Due to lack of trust, efficiency and scalability in user authentication and access process to the cloud environment. Therefore an efficient and scalable mutual authentication scheme algorithm is proposed. In this paper a mutual authentication scheme is proposed using 160 bits Elliptic Curve Diffie Hellman to authenticate the users requesting access to the cloud environment and the authentication server according to pre-defined parameters. The smaller key size of Elliptic curve and the lower storage capacity compared with other authentication algorithms enhance the security performance of the authentication process. The proposed model tries to enhance the performance of the authentication process and defend against security attacks as Man-in-The-Attacks and replay attack. The encryption of the generated random numbers from the user and the server protect against replay attacks and using SHA-1as signature algorithm provides a pre authentication tool before accessing the cloud services. The steps of the proposed model are as follows:

A-User calculates the hash value for his password using sha1 generating hash value less than 160 bits. The using of Hash algorithm will verify the message integrity and assure that the message has not been changed or corrupted. This hash value will be used after that as a reference when it compared with the stored hash code in the server to verify user authentication attempts. Since this hash value produced from a one way function so it can't be regenerated. It is only compared with the hash value that will be generated from the server with the function.

B-The user then generates random number **R1** from pseudo random generator. The generated random number is used in the proposed protocol to ensure that old communications cannot be reused in any attacks. This random number also can be used in stream cipher to ensure that the key stream is different for every session by generating random nonce for every session.

C-The user generates a timestamp T1. The generated timestamp protect against replay attack. The value of the timestamps must be with acceptable range of time. A precision time must be identified between the server and the user to make it possible for every communicating parties to determine whether the message timestamp is too old or fresh. By using timestamps concatenated with the hash value of the server. The user can assure that the message from authorized party.

The user then sends his ID concatenated with the hash of his password concatenated with timestamp T1 concatenated with random number R1 encrypted with AES-256 to the server. The encryption of these parameters over the communicating channel will protect against modification or alteration from unauthorized opponent.

D- The server receives the parameters from the user and starts the checking process by verifying the ID of the user from the pre stored database in the server. This assures the identity of the sender and that the message comes from authorized user

E-The server checks the timestamp to assure that replay attacks doesn't occurs by comparing the received time of the message with the timestamp of sending the message. This assures the freshness of the message by keeping a track of creation and modification time of the message.

F-Server calculates the hash value and compares it with the received value from the user. Since the creation of the hash value in the server from a specific function must be matched with the created value in the server using the same function.

G -Server checks the previous parameters (comparing the received timestamp with the transmitted one, checking the ID of the user and compare the received hash with the computed one) and if the check succeeded.

H- Server will generate random number R2 from its pseudo generator which will assures that old communications and Man in The Middle attack is not involved in communication from the server to the user.

I- A timestamp T2 is generated as a reference time compared with the received time from the user. The timestamp T2 must be with acceptable range relative to T1. The server will send the generated random number R2 concatenated with R1 and timestamp T2 encrypted with AES-256 to the user. The sending of R1 to the user will give assurance to the user that the message is originated from the server by comparing it with the generated one.

J-The user will check the received message from the server and check the random number R1 to assure that the received message is from the server by comparing the received value with the stored one.

K-The user checks the received value R2 which assures that the received message comes from the server

L-The user checks the timestamp T2 to assure that no message replay occurred and the received time is within acceptable range with the sending time.

M- After the check succeeded, user generates Q_U by multiplying the hash of the random number R_1 by the generating point of the curve G and send Q_U to the server.

$$Q_u = \text{Hash}(R_1)*G$$

N- The server also generates Q_S by multiplying the hash of the random number R_2 with the same generating point of the curve and send Q_S to the user.

$$Q_s = \text{Hash}(R_2)*G$$

After exchanging the value of Q_S and Q_U between the user and the server a mutual authentication between the server and user is achieved. So that every time the value of the random number changed the hash will be changed and the value of Q will be also changed due to changing the value of hash of the random number.

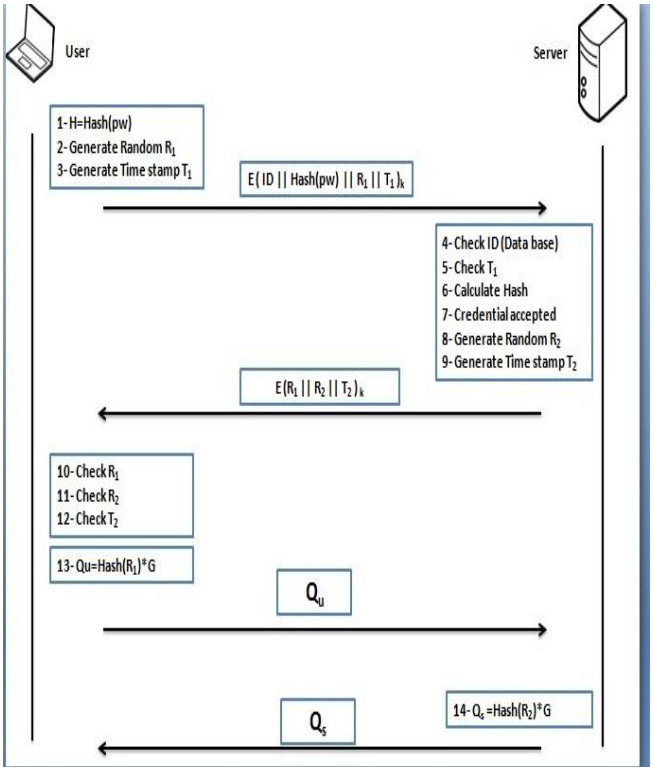

Figure 1
(Proposed Scheme)

III. Security Analysis

Evaluation process of the proposed model has been done according to the following parameters:

A- The protection against unauthorized access by the using SHA-2 algorithm to verify that the service request is from authorized user by comparing the hash of the password of the user by the stored hash in the data base of the authentication server.

B -The use of timestamp between the user and the server protects against replay attack as the time comparison between the user and the server assure that old communication is not replied or take part in communications.

C- The use of AES-256 for encrypting the data between the user and the server will protect against Man in The Middle attack since the opponent cannot compute the key that is used for encryption or decryption.

D-Mutual authentication: The proposed model provides a strong mutual authentication between the user and the authentication server.

The user challenges the authentication server in the authentication request message encrypting his identity by using secret key computed by the user. Only the server can recomputed the secret key and retrieve the user identity.

E- The using of ECC as authentication algorithm provides an efficient and scalable tool for authenticating the user with the server due to its smaller key size. Complex multiplication method is used in generation of the curve due to smaller space storage that is used in storing the field parameters, which improves the computational cost and efficiency.

F- Identity based encryption of the user and the authentication server which allow the sender to encrypt a message to an identity without access to a public key certificate authority. The user uses his id and calculates the hash value for his password and encrypts these parameters then sends these parameters to the server which makes the same procedures and makes the required comparison using the received parameters from the user to assure the identity of the user.

G -Confidentiality and message integrity between the user and the authentication server which assure that the received messages contain no modification, insertion, deletion or replay is achieved by using the hash function SHA-2 which calculate the hash value of his password generating 160 bits (hash value).This hash value concatenated with other parameters are encrypted using AES 256 before sending to the authentication server this provides authentication it also provides digital signature because only the user can produce the encrypted hash value.

IV. ASSESSMENTS AND VERIFICATION

Verification is one of the most important tools for evaluating security protocols. There are many tools for evaluating and describing security protocols, these tools help security experts to find and evaluate weakness in many security architectures. Scyther is one of the most important tools for verification of security protocols which is designed for evaluating and analyzing security protocols. Scyther has its own language for describing and evaluating security protocols which provides different traces for many attacks for security protocols. Different traces for the attacks in each protocol [12].

A. Verification for protocol 1*:*

Protocol 1 claims that the communication between the two communicating parties satisfies the following security requirements:

1- Mutual authentication. 2- Secure Against Passive Adversaries

The following code describes protocol 1 as transmitted and received message between two communicating parties representing how the message fulfills the protocol claims.

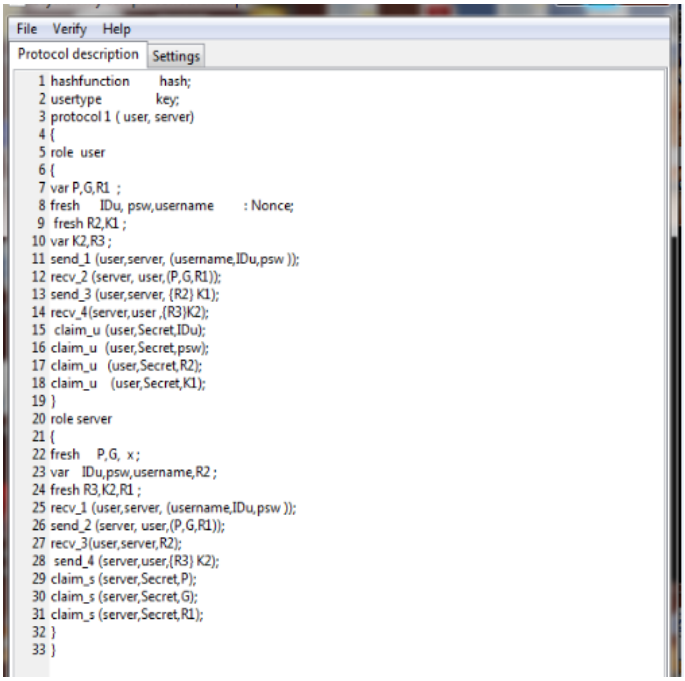

Figure 2
(Scyther code for protocol 1)

The previous code describes protocol 1 as two roles with detailed description for each received and transmitted messages between the user and the authentication sever.

The code also describes the security claims that protocol 1 claim that these claims are fulfilled by protocol 1.

After verifying the protocol the following attacks appear:

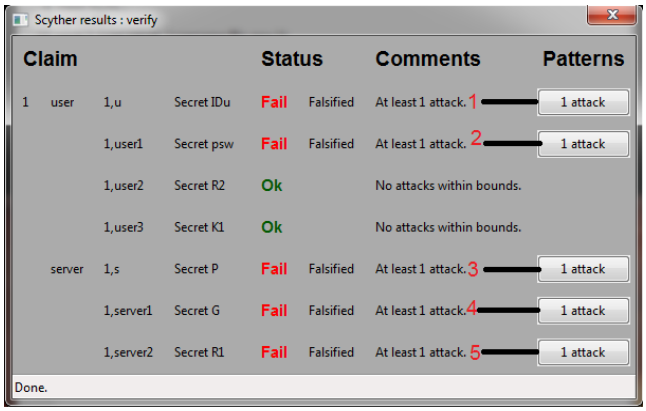

Figure 3
(Attacks Details for Protocol1)

From the verification for protocol 1five security attacks were discovered and a single track for each attack is illustrated and a brief description for each attack is shown in the following figures. One of the advantage of using scythe as a verification tool is the detailed description for each attack which gives an efficient helping tool for security experts to build efficient and strong security architecture for their systems. Figure 4-shows a detailed description for attack 1 and the other attacks will be shown in the following figures.

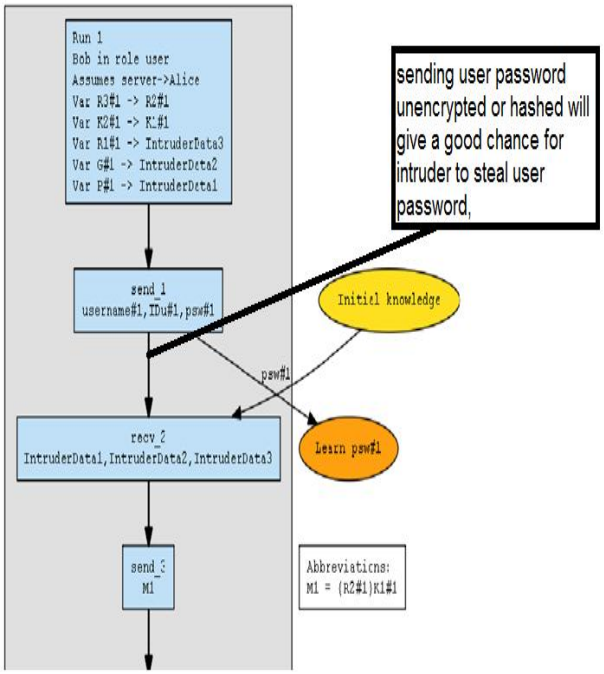

Figure 4
(Description for attack 1)

Figure 4 shows a detailed description for attack 1 as sending user password without over encryption or hashing over communication channel will make it easy for stealing and compromising.

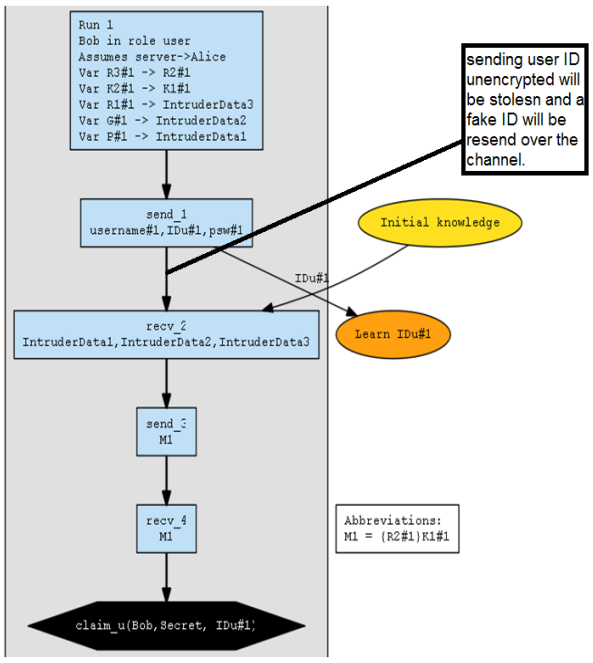

Figure 4-a
(Description for attack 2)

Figure 4-a describes a detailed description for attack 2 as sending user ID unencrypted over public communication channel will be compromised and easy to be stolen and a fake ID can be resend over the communication channel .

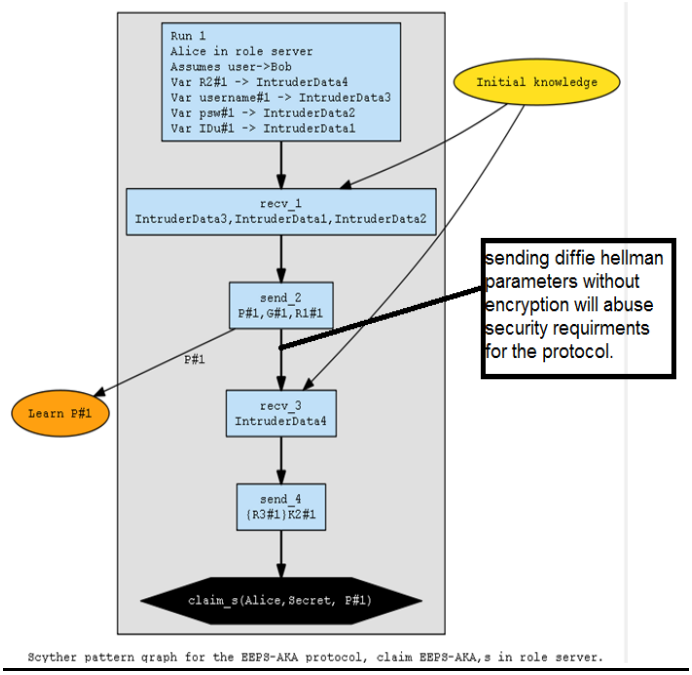

Figure 4-b
(Description for attack 3-4)

Figure 4-b shows a detailed description for attack 3 and 4 as protocol 1 uses Diffie Hellman as authentication protocol so Diffie Hellaman parameters must be agreed between the two communicating parties. So sending these parameters unencrypted over communication channel will abuse the authentication process due compromising the authentication parameters.

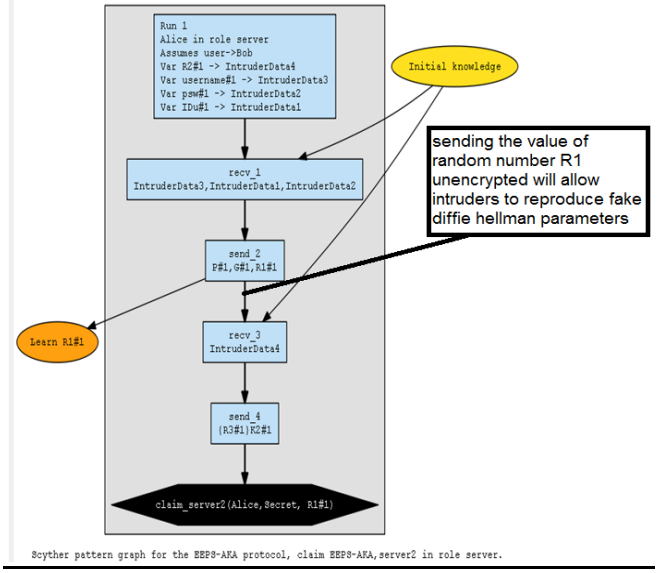

Figure 4-b
(Description for attack 5)

Figure 4-c shows a detailed description for attack 5 as sending the value of generated random number R1 unencrypted will make compromising the session key as every session key is generated with a new generated random number. The following section will show how the proposed model countermeasures the security breaches in protocol 1 and how to overcome each attack.

V. VERIFICATION FOR PROPOSED MODEL

The proposed model has been proposed to countermeasures protocol 1 attack as follow:

A-The protection against unauthorized access by the using SHA-2 algorithm to verify that the service request is from authorized user.

B- The use of AES-256 for encrypting the data between the user and the server will protect against Man in The Middle attack. Encrypting the produced random numbers will give an accurate time track for the transmitted and received messages so preventing replay attack for any message.

C- Mutual authentication: The proposed model provides a strong mutual authentication between the user and the authentication server.

- **The Following Figures Show How the Proposed Model Countermeasures protocol 1 attack.**

Figure 5 shows how the proposed model uses AES-256 to encrypt the user ID before sending over public communication to protect against any opponents or attackers.

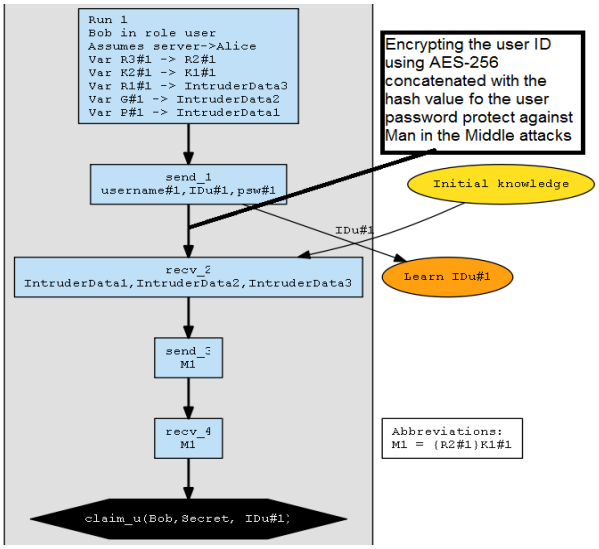

Figure 5
(Description for proposed model solution)

Figure 5-a show how the proposed model uses SHA-2 to protect the user password from being stolen by any opponent. The server can only recalculate the hash value of the user password to verify that the request is from authorized user.

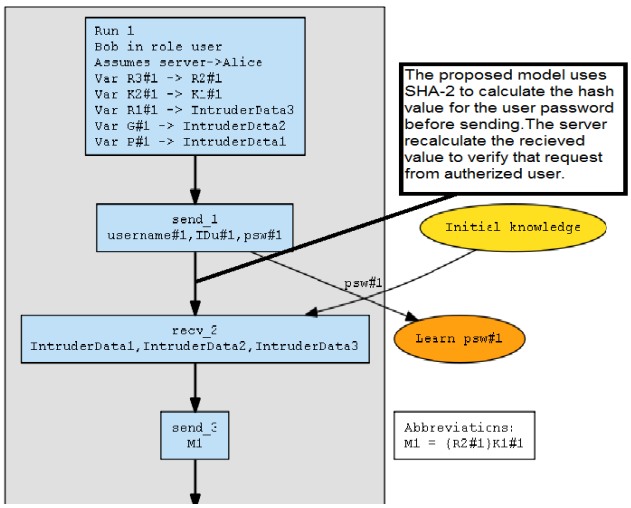

Figure 5-a
(Description for proposed model solution)

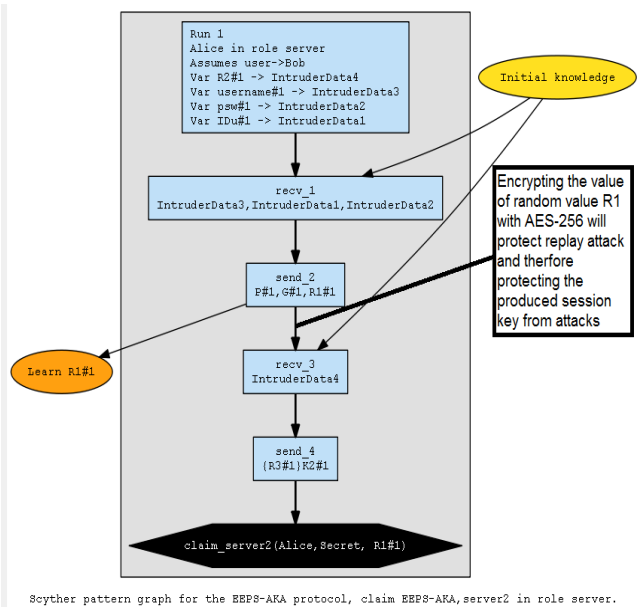

Figure 5-b
(Description for proposed model solution)

Figure 5-b shows how the proposed model countermeasures the replay attacks. Encrypting the generated random number R1 will protect against any replay attack so protecting the produced session keys from compromising by any attackers. Figure 6 shows the scyther code for the proposed model and the verification results after the proposed solutions for the security breaches in protocol 1.

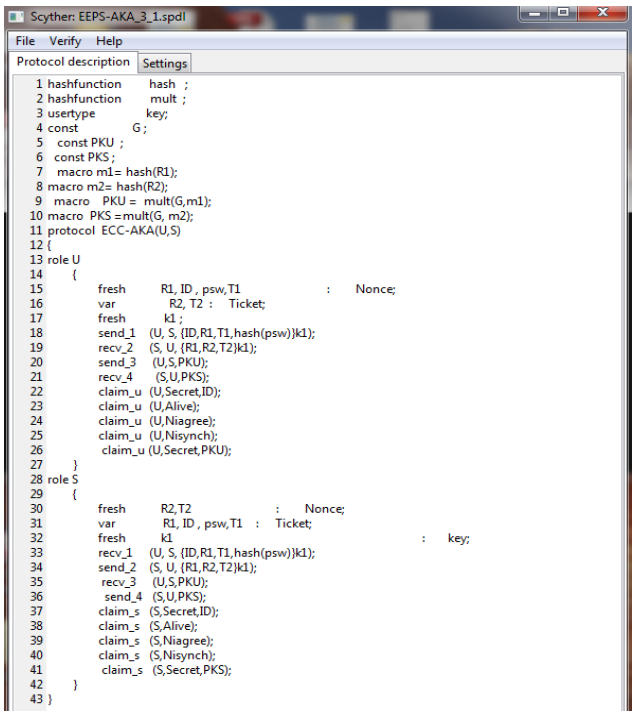

Figure 6
(scyther code for the proposed model)

After verification of the proposed model solution no attacks or security breaches appear during message transmission between the user and the server.

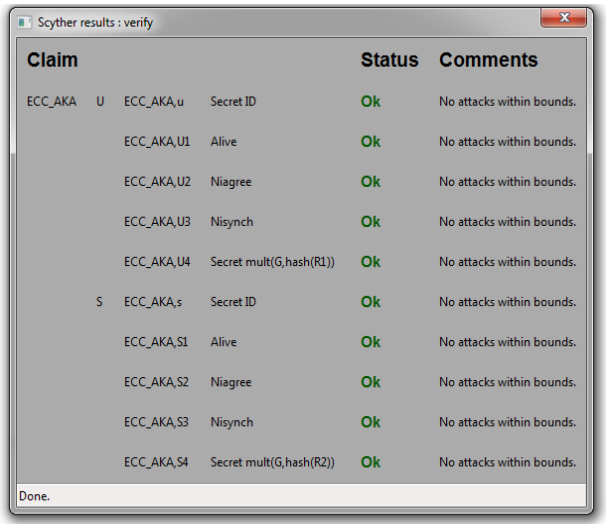

Figure 7
(Final Result for Verification of Proposed model)

VI. CONCLUSION AND FUTURE WORK

With the development of cloud as a new emerging technology and the challenging issue during user authentication and access control in cloud based environment. A strict authentication scheme is proposed to control the access to the cloud services .The proposed authentication scheme and key agreement protocol has been proved to be more secure and efficient than other authentication schemes. Since using small key size of Elliptic Curve will improve the authentication process due to fast processing of the keys. Using AES-256 as an encryption algorithm protects the data exchange between the user and the server from various attacks. ECC will enhance the computational efficiency and enhance the usage of available storage resources which make it an efficient choice as authentication protocol for cloud computing. There are different lines for future work to be addressed as using quantum cryptography for encrypting the transmitted and received messages between the sender and the receiver.

REFERENCES

1. Chaowei Yang, zenlong Li,"*Big Data and cloud computing: innovation opportunities and challenges*", International Journal of Digital Earth, 2017 vol. 10, no.1
2. Gururaj Ramachandra ,Farrouka Aslam Khan, "*A Comprehensive Survey on Security in Cloud Computing*", The 3rd International Workshop on Cyber Security and Digital Investigation (CSDI 2017) Procedia Computer Science 110 (2017) 465–472.
3. Anwaar AlDairi and Lo'ai Tawalbeh, " *Cyber Security Attacks on Smart Cities and Associated Mobile Technologies*",The International Workshop on Smart Cities Systems Engineering (SCE 2017), Procedia Computer Science 109C (2017) 1086–1091.
4. R.Kalaiprasath, R.Elankavi," *Cloud Security and Compliance - a Semantic Approach in end to End Security"*, International Journal on Smart Sensing and Intelligent Systems Special Issue, September 2017
5. Yi Han, Jeffrey Chan and Christopher Leckie, *"Using Virtual Machine Allocation Policies to Defend against Co- Resident Attacks*", IEEE Transaction on Dependable and Secure Computing, Vol. 14, No. 1, February 2017
6. Omar Achbarou, Salim Elbouanani, "*Cloud Security: A Multi Agent Approach Based Intrusion Detection System*", Indian Journal of Science and technology, *Vol10(18),DOI:10.17485/ijst/2017/v10i18/109044, May2017*.
7. Ruhul Amin, Debasis Giri ," *A Two-Factor RSA-Based Robust Authentication System for Multiserver Environments*", Security and Communication Networks, Volume 2017, Article ID 5989151.
8. Chung-Huei Ling, Cheng-Chi Lee, " *A Secure and E_cient One-time PasswordAuthentication Scheme for WSN*", International Journal of Network Security, Vol.19, No.2, PP.177-181, Mar. 2017.
9. Mohammad Wazid, Ashok Kumar Das," *Provably secure biometric-based user authentication and key agreement scheme in cloud computing"*, Security and Communication Networks Security Comm. Networks 2016; 9:4103–4119.
10. Faraz Fatemi,Shiva Gerayeli Moghaddam", *A scalable and efficient user authentication scheme for cloud Computing environments*", *IEEE 2014 Region10 Symposlum*.
11. Shorab Rouzbeh, Iman Ghavam, "*A client -based user authentication and encryption algorithm for secure accessing to cloud servers*",2013 IEEE Student Conference on Research and Development 16-17 December 2013, putrajaya, Malaysia.
12. Anil Sangwan, V R Singh," *A Secure Authentication Scheme for WiMax Network and Verification using Scyther Tool"*, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 11 (2017).