LTE Authentication Protocol (EPS-AKA) Weaknesses Solution

Mohammed Aly Abdrabou, Ashraf Diaa Eldien Elbayoumy, and Essam Abd El-Wanis
Dept. of Communication
Military Technical College
Cairo, Egypt
abdrabou@outlook.com, adiaa@afmic.com, mohwanees@yahoo.com

***Abstract*—Extensible Authentication Protocol (EAP) is an authentication framework in Long Term Evolution (LTE) networks. EAP-AKA is one of the methods of EAP which uses the Authentication and Key Agreement (AKA) mechanism based on challenge-response mechanisms, EAP-AKA is used in the 3rd generation mobile networks then modified and inherited to 4th generation mobile networks (LTE) as Evolved Packet System Authentication and Key Agreement (EPS-AKA) mechanism which is used when the user access the network through E-UTRAN. EPS-AKA vulnerabilities are disclosure of the user identity, Man in the Middle attack and Denial of Services (DoS) attacks so a robust authentication mechanism must replace EPS-AKA to avoid such attacks. In this paper, Modified Evolved Packet System Authentication and Key Agreement (MEPS-AKA) protocol based on Simple Password Exponential Key Exchange (SPEKE) and symmetric key cryptography is proposed to solve these problems by performing a pre-authentication procedure to generate a dynamic key every time user access to the network, also each message send or received is confidentially protected. Scyther tool is used to verify the efficiency of the proposed protocol. EPS-AKA and MEPS-AKA are simulated using C programming language to calculate the execution time for both algorithms. The proposed protocol is simulated using a client-server application program using C# programming language.**

Keywords—EAP-AKA, LTE, SPEKE, AES, EPS-AKA, Scyther

I. LTE Network and Security Architecture

Radio-Access Network (RAN) and Core Network (CN) was redesigned and this known as System Architecture Evolution (SAE) (which is evolution of the overall network architecture). The RAN which and the EPC are referred to as the Evolved Packet System (EPS). It was decided to separate the user data (UP) and the signaling (CP) to make operators can operate their network easily independent as shown in Fig 1 [1]. The RAN in LTE is called E-UTRAN and the CN is called EPC as shown in Fig 2, it consists of several different types of nodes as follow [2]:

1) Mobility Management Entity (MME): Control-plane node of the EPC, handles the signaling related to mobility and security for E-UTRAN access handling of security keys.

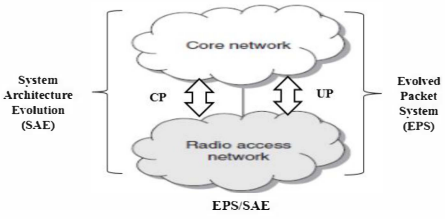

Fig 1 Overall system architecture.

2) Home Subscriber Service (HSS): database that contains subscriber information user authentication.
3) The Serving Gateway (S-GW): User-plane node transports the IP data traffic between the UE and the external networks.
4) The Packet Data Network Gateway (P-GW): Connects the EPC to the internet, Allocation of the IP address for a terminal, the mobility anchor for non-3GPP RAN, as CDMA2000.
5) ENodeB: One eNodeB can be connected to multiple MMEs/S-GWs for the purpose of load sharing and redundancy.

UE reach the EPC using E-UTRAN which is not the only access technology supported: 1 - 3GPP RAN, by interworking between E-UTRAN (LTE and LTE-Advanced), GERAN (GSM) and UTRAN (UMTS). 2 - Non-3GPP RAN (e.g. cdma2000).

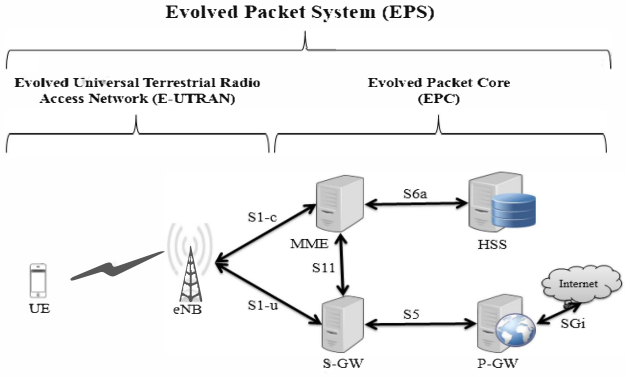

Fig 2 LTE network nodes architecture.

A security feature is a service capability that meets one or several security requirements. A security mechanism is an element that is used to realize a security feature. All security features and security mechanisms form the security architecture as shown in Fig 3. Example of a security feature is user data confidentiality. A security mechanism that may be used to implement that feature is a stream cipher [3].
The 3GPP classify security feature in the LTE network into five groups as follow:

1) Network access security: the set of security features that provide users with secure access to 3G services, and which in particular protect against attacks on the (radio) access link.
2) Network domain security: the set of security features that enable nodes in the provider domain to securely exchange signaling data, and protect against attacks on the wireline network.
3) User domain security: the set of security features that secure access to mobile stations
4) Application domain security: the set of security features that enable applications in the user and in the provider domain to securely exchange messages.
5) Visibility and configurability of security: the set of features that enables the user to inform himself whether a security feature is in operation or not and whether the use and provision of services should depend on the security feature.

Our work will be in the network access security; where Entity authentication comprises two steps and should occur at each connection setup between the user and the network. First, user authentication ensures that the serving network corroborates the user identity of the user. Next, network authentication ensures that the user's connection is to a serving network with an up-to-date authorization from the user's home environment (HE) to provide services; to achieve mutual authentication between the user and the network.

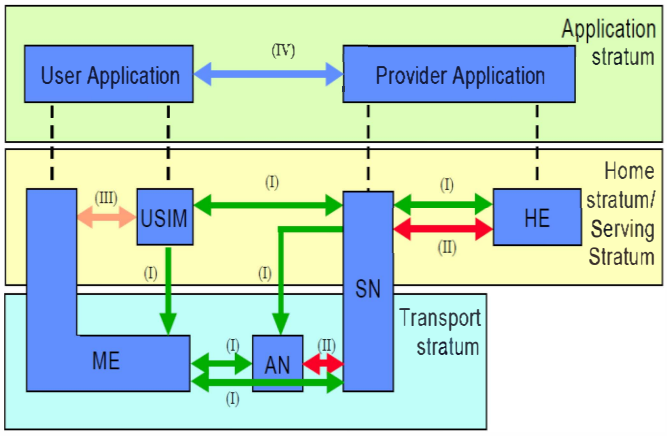

Fig 3 Overview of the security architecture

II. EPS-AKA PROTOCOL

In LTE network, EPS-AKA protocol is used as an authentication mechanism, where Extensible Authentication Protocol (EAP) is an authentication framework, not a specific authentication mechanism, providing for the transport and usage of keying material and parameters generated by EAP methods [4], EAP-AKA is one of the methods of EAP uses the Authentication and Key Agreement (AKA) mechanism based on challenge-response mechanisms and symmetric cryptography. The authentication procedure starts after connection establishment as shown in Fig 4 between UE and MME as follow [3-6]:

1) MME send an ID request to the UE via eNB.
2) UE respond by IMSI.
3) MME request an EPS authentication vector (AV) from the HSS, Based on the IMSI, the HSS looks up the key K and a sequence number associated with that IMSI, the AuC increases the SQN and generates a random challenge (RAND) beside master key K as input to cryptographic functions, and generates AV, this AV consists of: XRES, AUTN, KASME and RAND.
4) HSS/AuC send the AV to MME, the MME keeps the KASME and XRES but forwards RAND and AUTN to UE.
5) Both RAND and AUTN are sent to the UE to calculate its own version of AUTN using its own key K and SQN and compare it with the AUTN received from the MME, to make user authenticate the network, if matched UE compute RES using cryptographic functions with the key K and the RAND. Also computes CK and IK.
6) Sends the RES back to the MME, MME authenticates the terminal by verifying that the RES is equal to XRES. This completes the mutual authentication. The UE then uses the CK and IK to compute KASME in the same way as HSS. Then both UE and MME now have the same key KASME.
7) Then MME send to UE the success or failure authentication process.

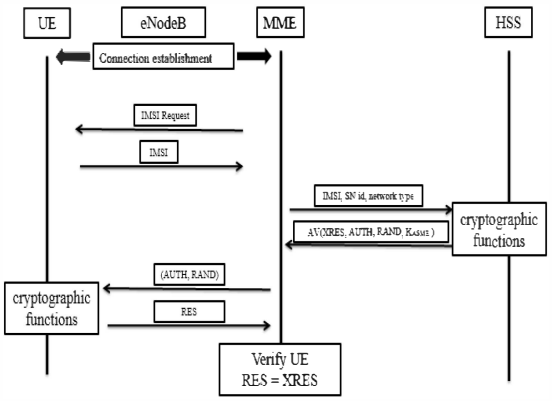

Fig 4 EPS-AKA procedure.

III. WEAKNESS IN THE EPS-AKA PROTOCOL

The first weakness is; Disclosure of the user identity which caused when the UE registers to the network for the first time, So UE transmits IMSI in plaintext. So attacker can capture the IMSI as shown in Figure 5. The attacker can impersonates the UE afterwards and send the IMSI to the MME to gain some information. Once the IMSI has been obtained, the adversary could acquire subscriber information, location information, and even conversation information, and then hide the real UE and launch the other attacks such as DoS attacks to destroy the network [3, 7].

Figure 5 Disclosure of the user identity.

The second is; Man In The Middle attack (MITM) where the attacker obtain the UE's IMSI, then tries to register with genuine BS by this IMSI then network sends RAND and AUTN. Attacker disconnects when these parameters are received. Afterwards the genuine UE register with a false BS by sending the original the RAND and AUTN and getting it to calculate RES. The false base station re-initiates an authentication request to the network. This time the false station has the correct RES as shown in Figure 6.

The third is; Denial of Services (DoS) attack, since MME manages numerous eNBs in the flat LTE architecture, the base stations in the LTE networks are more susceptible to the attacks compared with those in the UMTS architecture, where the serving network in the UMTS only manages a couple of RNCs in a hierarchical way. Once an adversary compromises base station, it can further endanger the entire network due to the all-IP nature of the LTE networks; Adversary can launch DoS attacks to the HSS and the MME as shown in Fig 7. The adversary can hide a legitimate UE to constantly send fake IMSIs to Fool the HSS. Thus, the HSS has to consume its computational power to generate excessive authentication vectors for the UE. On the other hand, the MME has to consume its memory buffer to wait overly long period of time for a legitimate or false response from the corresponding UE [7].

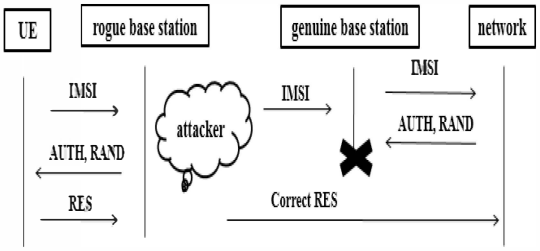

Figure 6 MITM attack.

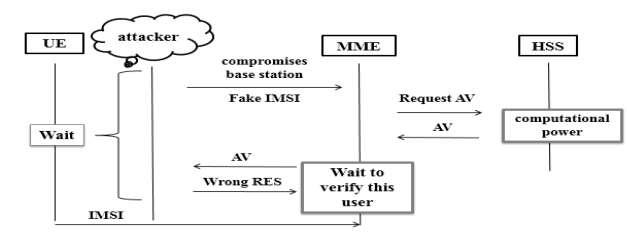

Fig 7 DoS attack.

IV. RELATED WORK

A Security Enhanced Authentication and Key Agreement (SE-EPS AKA) based on Wireless Public Key Infrastructure (WPKI) has been proposed in [8]. The scheme ensures the security of user identity and the exchanged message with limited energy consumption by using Ellipse Curve Cipher (ECC) encryption. It has been pointed out in [9] that the SE-EPS AKA protocol is vulnerable to brute force and intelligent force attacks and thus it cannot guarantee the security of the user identity. Then, an ensured confidentiality authentication and key agreement (ECAKA) has been proposed to enhance the user's confidentiality. By the scheme, all the AKA messages are fully protected on the integrity by encryption, which can prevent the disclosure of identity of the users and the users being tracked, due to using public-key based protection mechanisms in [9] to overcome the shortcoming of the EPS AKA protocol, and thus can achieve a mutual authentication and ensure the security communication between the UE and HSS/AuC by the use of the UE and/or the HSS/AuC of public key certificates, these will cause a large number of computational costs, storage costs and communication costs for mobile devices with resource limitation. A slightly modified version of the EPS-AKA protocol has been presented in [10]. The scheme introduces a new subscriber module ESIM instead of the USIM and provides a direct online mutual authentication between the ESIM and the MME/HSS to overcome the shortcomings of the EPS-AKA protocol only with minor modifications of the access security architecture. However, it may suffer compatible problems in the LTE networks due to the use of the new ESIM. Since the HSS needs to participate in every authentication procedure for each UE, it may incur a large number of communication delays and thus cause signaling congestion on the HSS. In addition, it cannot overcome the disclosure of user identity [7].

V. PROPOSED MODIFIED EPS-AKA PROTOCOL (MEPS-AKA)

A Modified EPS-AKA protocol (MEPS-AKA) is proposed to overcome those weakness mentioned above. Based on a Simple Password Exponential Key Exchange (SPEKE) [11, 12] protocol and symmetric key cryptography (AES), SPEKE includes a little change than a Diffie-Hellman key exchange where a password is hashed at the start.

There is pre-shared password (Psw), prime number (P), a shared static key (Kum) and hash algorithm (H ( )) between MME and UE, also a shared key (K (h, m)) between MME and HSS, also a shared key (K) between UE and HSS. Note that Kum is a static key between MME and UE and every time the user access the network a dynamic key is produced k (u, m) then MME and UE continue the authentication procedure with k (u, m). The authentication procedure starts after connection establishment as shown in Fig 8 between UE and MME through the following steps:

1) UE compute A and generate two random nonce (Ru1, u) we send a number assigned to this IMSI in the network to be easy to the network to manage their network in the key management procedure; send ({Ru1, A} kum, Ru1| related number to UE) to MME.
$$A = H(psw)^{u} \bmod p$$
2) MME compute B and generate two random nonce (Rm1, m); send ({Rm1, Ru1, B} kum, Rm1) to UE, step 1 and 2 is a pre-authentication steps to get a dynamic key between MME and UE every time the user access to the network step 1 and 2 is called a preauthentication process.
$$B = H(psw)^{m} \bmod p$$
3) From step 1 and 2 (UE and MME) compute their new dynamic key (k (u, m)) which is based on (SPEKE); then UE generate a random nonce (Ru2) and generate a timestamp (TS) with the assumption that synchronization is done between UE and MME; then UE send ({IMSI, TS, Ru2, Rm1} k (u, m), Ru2) to MME.
4) MME retrieve IMSI and then if this IMSI has no relation with the number assigned to it the MME detect that there is attack and disconnect the UE and if related then create a random nonce (Rm2) also check the timestamp; send ({IMSI, k (u, m), Ru2, Rm2} k (h, m), Rm2) to HSS.
5) HSS retrieve the IMSI and get the corresponding key for this IMSI (k), calculate the shared key between UE and HSS (K (u, h)), generate nonce (Rh), calculate the expected response to authenticate the UE and compute AUTH_HSS to allow UE authenticate HSS; then send ({AUTH_HSS, XRES , Rm2, Rh } k(h , m), Rh) to the MME.
$$AUTH\_HSS = \{Rh, Ru2\}\ k\ (u, h)$$
$$K\ (u, h) = k\ (u, m) \oplus k$$
6) MME generate random nonce (Rm3) and compute AUTH_MME to allow UE authenticate MME; send ({AUTH_HSS, AUTH_MME} k (u, m), Rm3, Rh) to UE.
$$AUTH\_MME = \{AUTH\_HSS \oplus Rm3\}\ K\ (u, m)$$
7) UE authenticate MME by decrypting AUTH_MME then XORing the result with Rm3 then compare AUTH_HSS retrieved with the one received. UE authenticate HSS by checking the values of retrieved Rh and Rh received, if they are matched then calculate the AUTH_UE and generate a random nonce (Ru3); send ({AUTH_UE, Ru3, Rm3} k (u, m), Ru3) to MME.
$$AUTH\_UE = \{R\ h\}\ k\ (u, h)$$
8) Network authenticates UE by checking the values of AUTH_UE to the value of the expected response. If all these steps go successfully then the user access to the network successfully.
$$Check: AUTH\_UE = XRES$$

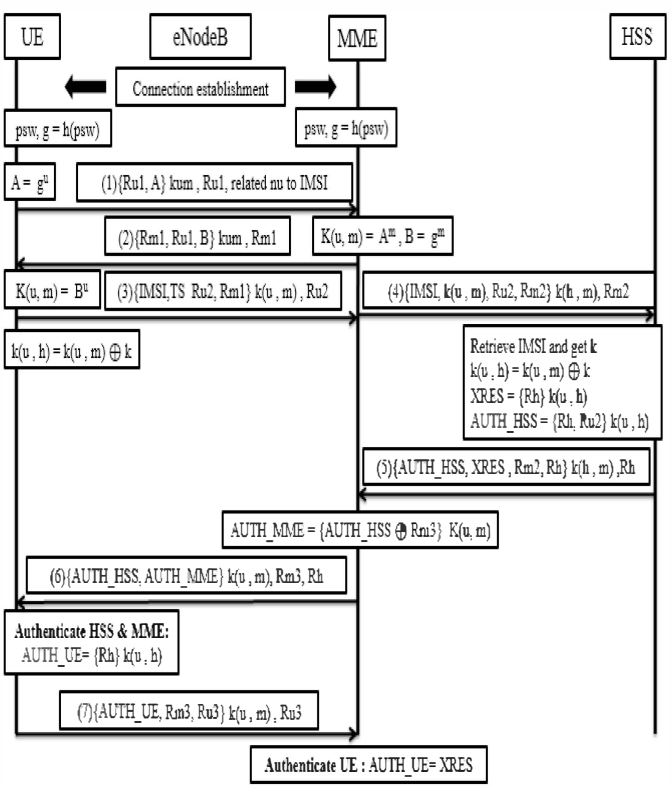

Fig 8 MEPS-AKA procedure.

VI. Security Analysis

1) User identity Protection (encrypted IMSI): The user identity is protected and can't be captured by attackers. The UE encrypts its IMSI in third message using AES-128 in CTR mode.
2) Secure against man-in-the-middle attack: The attacker can't retrieve the IMSI from the third message as the attacker cannot calculate the dynamic key (k (u, m)), also the attacker can't re-send the encrypted IMSI in later time due to the presence of the timestamp. So the proposed protocol eliminate MITM.
3) Mutual authentication: UE authenticates MME and HSS; by checking AUTH_MME and AUTH_HSS respectively. MME authenticate UE: The MME

verifies the UE by comparing AUTH_UE with the XRES.

4) Data confidentiality: EPS-AKA could not provide the data confidentiality during the authentication process, because it does not encrypt some of messages between UE, MME and HSS. MEPS-AKA provides the data confidentiality during the authentication process to prevent the attacker from sniffing the entire message during the authentication process, by encrypting all the entire messages using symmetric key cryptography.
5) Secure against replay attack: The MME verifies that the timestamp (TS) is in the correct range that occurs in step (3) that makes the MEPS-AKA protocol secure against replay attacks.
6) Elimination of SQN synchronization: The MEPS-AKA protocol does not use SQN mechanism in synchronization which is a great add to the bandwidth consumption, and also diminishing re-synchronization processes which used to occur due to SQN errors.
7) DOS attack: the network detect DOS attack in the third message not as the standard which wait message sixth to detect this type of attack.

VII. The Proposed Protocol Verification Results

Scyther is a tool for the formal analysis of security protocols under the perfect cryptography assumption, in which it is assumed that all cryptographic functions are perfect: the adversary learns nothing from an encrypted message unless he knows the decryption key. The tool can be used to find problems that arise from the way the protocol is constructed.

Fig **9** show an overview for protocol verification flow chart using Scyther tool. Scyther tool used to check security properties for EPS-AKA and MEPS-AKA, to verify that the performance of the proposed protocol is more secure than the standard protocol[13, 14].

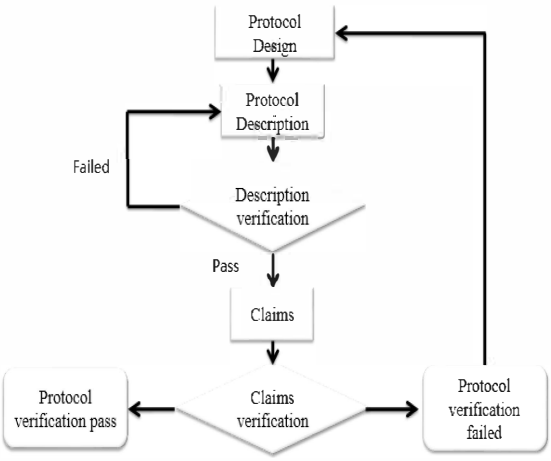

Fig 9 Protocol Verification Flow Chart

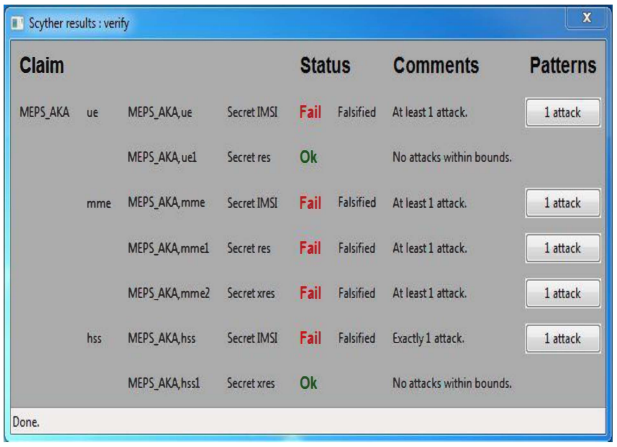

Figure 10 output result of EPS-AKA

The description of the EPS-AKA is as specified in section II. We found that the IMSI is sent plain to the MME, so in message 1 from UE and MME side the IMSI can be captured by intruder, also in message 2 from HSS and MME side the IMSI can be captured by intruder and perform multiple attack as mentioned. Also res and the xres is insecure from the point of view of MME. the output of the verification is shown in Figure 10.

The description of the proposed protocol is accomplished in two steps. The first step is the exchange of five messages between (UE, MME) as specified in section V, where the messages (1, 2, 3, 6 and 7) are exchanged in the first step as shown in Fig 8. The second step is the exchange of two messages between (MME, HSS) as specified in section V, where the messages (4 and 5) are exchanged in the second step as shown in Fig 8. The following claims (Secrecy, SKR, Alive, Weakagree, Niagree, and Nisynch) are checked for the elements in the exchanged messages. The output result of the first and second step are shown in Fig 11 and Fig 12.

Step1; description of the messages between (UE) and (MME) are as follow:

1) from the UE point of view as follow:

send_1 (ue, mme, {Ru1, exp (hash (psw), u)} kum, Ru1);
recv_2 (mme, ue, {Rm1, Ru1, z} kum, Rm1);
send_3 (ue, mme, {IMSI, Ts, Ru2, Rm1} k (ue, mme), Ru2);
recv_4 (mme, ue, {authhss, authmme} k (ue, mme), Rm3, Rh);
send_5 (ue, mme, {res, Rm3, Ru3} k (ue, mme), Ru3);

- And the claims are written as follow:

claim_ue (ue, SKR, exp (z, u));
claim_ue (ue, Secret, IMSI);
claim_ue (ue, Secret, res);
claim_ue (ue, Alive);
claim_ue (ue, Weakagree);
claim_ue (ue, Niagree);
claim_ue (ue, Nisynch);

2) from MME point of view as follow:

recv_1 (ue, mme, {Ru1, z} kum, Ru1);
send_2 (mme, ue, {Rm1, Ru1, exp (hash (psw), m)} kum, Rm1);

recv_3 (ue, mme, {IMSI, Ts, Ru2, Rm1} k (ue, mme), Ru2);
send_4 (mme, ue, {authhss, authmme} k (ue, mme), Rm3, Rh);
recv_5 (ue, mme, {res, Rm3, Ru3} k (ue, mme), Ru3);

- And the claims are written as follow:

claim_mme (mme, SKR, exp (z, m));
claim_mme (mme, Secret, IMSI);
claim_mme (mme, Secret, res);
claim_mme (mme, Alive);
claim_mme (mme, Weakagree);
claim_mme (mme, Niagree);
claim_mme (mme, Nisynch);

Step2; description of the messages between (MME) and (HSS) are as follow:

1) from the MME point of view as follow:

send_1 (mme, hss, {IMSI, kum, Ru2, Rm2} k (hss, mme), Rm2);
recv_2 (hss, mme, {authhss, xres, Rm2, Rh} k (hss, mme), Rh);

- And the claims are written as follow:

claim_mme (mme, Secret, IMSI);
claim_mme (mme, Secret, xres);
claim_mme (mme, Alive);
claim_mme (mme, Weakagree);
claim_mme (mme, Niagree);
claim_mme (mme, Nisynch);

2) from HSS point of view as follow:

recv_1 (mme, hss, {IMSI, kum, Ru2, Rm2} k (hss, mme), Rm2);
send_2 (hss, mme, {authhss, xres, Rm2, Rh} k (hss, mme), Rh);

- And the claims are written as follow:

claim_hss(hss, Secret, IMSI);
claim_hss(hss, Secret, xres);
claim_hss(hss, Alive);
claim_hss(hss, Weakagree);
claim_hss(hss, Niagree);
claim_hss(hss, Nisynch);

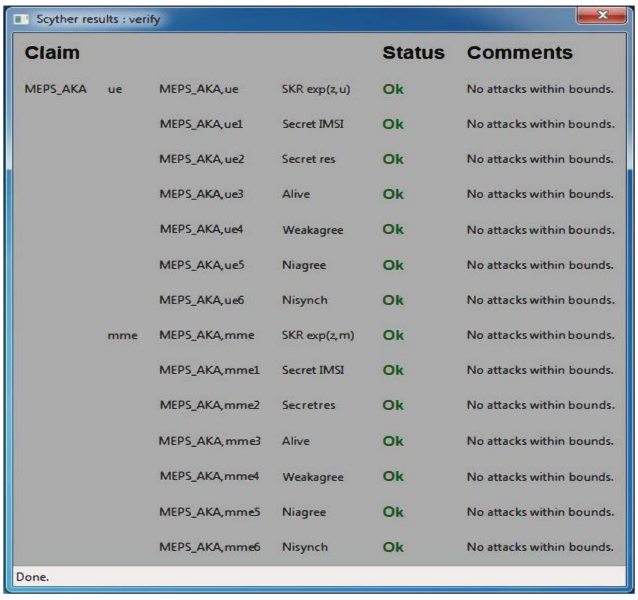

Fig 11 output result of first step

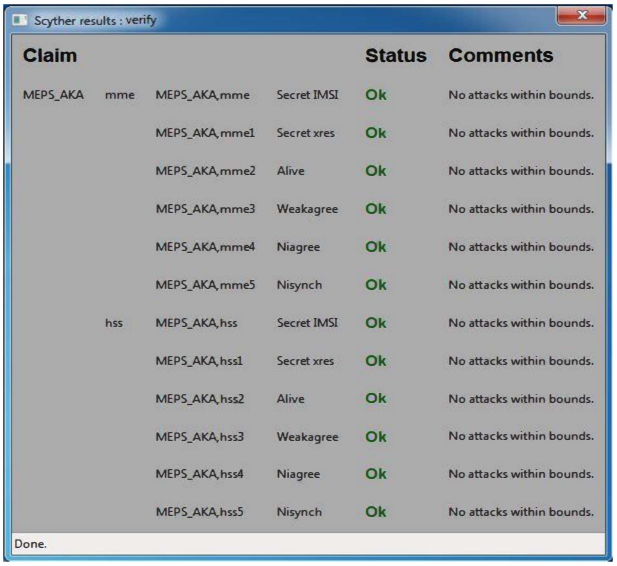

Fig 12 output result of second step

As shown in results the proposed protocol provides a strong mutual authentication between UE and HSS, protects User identity, increase the resistant to MITM attack and secure against replay attack. Also user identity cannot be retrieved or altered by the attacker, since it is protected by a strong secret key, only UE and HSS Owned this key. The secrecy claim of the IMSI passed the verification since it was sent encrypted. In addition, RES pass the verification. Also the element exp (hash (psw), u) and exp (hash (psw), m) is secure which is used in the preauthentication protocol to calculate the dynamic key. In addition, XRES pass the verification.

VIII. SIMULATION RESULTS

Two examples for the cryptographic functions of EPS-AKA protocol introduced by the 3GPP and they are not mandatory to be used. These examples gives a detailed description of the cryptographic function at UE side and at HSS side and implementation by C programming language. MILENAGE Algorithm is one of these two algorithms and is specified in [15-18]. The MILENAGE Algorithm is used to be compared with the proposed protocol. Implementation of MEPS-AKA is performed using eclipse environment and C programming language, the simulations have taken place on a Laptop using a 64-bit windows 7 operating system. The Laptop is running with processing speed of 2.4 GHz, to calculate the time consumed for EPS-AKA (515 mSec) and MEPS-AKA (840mSec). The output result from the Scyther tool show that modified more secure than the standard but unfortunately with a negative effect on time from the result of eclipse. The proposed protocol is simulated using a client-server application program.

The program was implemented using C# language. Assume that MME as a server that open a socket between it and the UE as a client on session 1 and the HSS as a client on session 2, where the message in session 1 is (1, 2, 3, 6 and 7) and the message on session 2 is (4 and 5), we implement all the procedure of the proposed protocol as detailed in section V, for example in sending message 1 which is the preauthentication process in the side of UE (client) we first calculate g = H(password), then get A = g^u mod P then encrypt the value (Ru1|A) using the static key Kum then send this message ({Ru1, A} kum , Ru1,related number to the IMSI of the user) to MME side through the socket, and all the other message is implemented using these sequence as message 1 as specified in section V. Visual studio used for implementation as shown in Fig 13.

Table 1 shows a comparison between the communication overheads of the proposed protocol and the standard protocol. The overheads increased from 1208 bits to 1920 bits which increased by 712 bits which reflects on the time consumption and make it increased [19]. But this increase is to overcome the security weaknesses in the standard protocol, as disclosure of user identity is overcome by sending the IMSI in the third message encrypted from the UE to the MME after generating a dynamic key from the preauthentication process (message 1 and 2) to encrypt the user identity reverse to the standard where UE send its identity plain to MME. also at the denial of services attack the time that the network take to detect that it is exposed to denial of services attack is decreased where the user is detected that he is a legitimate on or not on receiving message 3 in the MME, where MME compare the decrypted IMSI with the number related to this user and check the time stamp for replay attack. Also man in the middle attack is not achievable in the proposed protocol due using SPEKE protocol.

TABLE 1. COMMUNICATION OVERHEADS

Number of message	MEPS-AKA	EPS-AKA
1	256	16
2	320	64
3	256	104
4	384	640
5	320	256
6	192	128
7	192	-----
Total	1920	1208

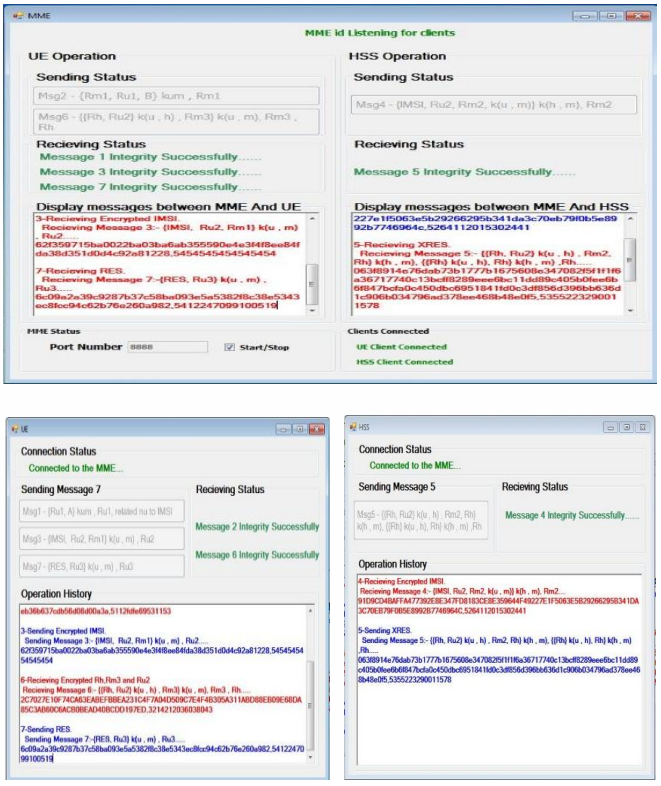

Fig 13 The UE, MME and HSS sides in the simulated results.

IX. CONCLUSION

The EAP-AKA protocol used in the 3G mobile networks and inherited to the 4G mobile networks (LTE) with minor modification leading to the Appearance EPS-AKA. This protocol fails to investigation full protection to the LTE network because it is exposed to disclosure of the user identity, MITM attack, replay attack and DoS attack. Due to these vulnerabilities a new authentication and key agreement protocol based on combination of SPEKE and symmetric key cryptography to overcome several vulnerabilities of EPS-AKA. Moreover, the proposed protocol provides strong mutual authentication between the user and the network, resistance to replay attack, protect user identity, and resist MITM attack. Therefore, the proposed protocol is more secure and efficient. Furthermore, formal analysis of the proposed protocol is done using Scyther tool, Scyther tool can be used to find problems that arise from the way the protocol is constructed, the follow claims (Secrecy, SKR, Alive, Weakagree, Niagree, and Nisynch) are checked for the elements in the exchanged messages. This verification showed that there are no flaws in the protocol design, and the proposed protocol is secure against well-known attacks. Also simulation of the EPS-AKA protocol and the MEPS-AKA is performed using C programming language, but unfortunately the proposed protocol execution time is more than the standard. Also the proposed protocol is simulated using a client-server application program. The application program was implemented using C# programming language.

REFERENCES

[1] E. Dahlman, S. Parkvall, and J. Skold, *4G: LTE/LTE-advanced for mobile broadband*: Academic press, 2013.

[2] 3GPP, "3GPP TS 23.002 Network architecture," ed, 2015.

[3] 3GPP, "3GPP TS 33.401 3GPP System Architecture Evolution (SAE); Security architecture," ed, 2015.

[4] S. Winter and J. Salowey, "Update to the Extensible Authentication Protocol (EAP) Applicability Statement for Application Bridging for Federated Access Beyond Web (ABFAB)," 2013.

[5] F.-Y. Leu, I. You, Y.-L. Huang, K. Yim, and C.-R. Dai, "Improving security level of LTE authentication and key agreement procedure," in *Globecom Workshops (GC Wkshps), 2012 IEEE*, 2012, pp. 1032-1036.

[6] C. Tang, D. Naumann, and S. Wetzel, "Analysis of authentication and key establishment in inter-generational mobile telephony," in *High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), 2013 IEEE 10th International Conference on*, 2013, pp. 1605-1614.

[7] J. Cao, M. Ma, H. Li, Y. Zhang, and Z. Luo, "A survey on security aspects for LTE and LTE-A networks," *Communications Surveys & Tutorials, IEEE,* vol. 16, pp. 283-302, 2014.

[8] L. Xiehua and W. Yongjun, "Security enhanced authentication and key agreement protocol for LTE/SAE network," in *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on*, 2011, pp. 1-4.

[9] J. B. Abdo, H. Chaouchi, and M. Aoude, "Ensured confidentiality authentication and key agreement protocol for eps," in *Broadband Networks and Fast Internet (RELABIRA), 2012 Symposium on*, 2012, pp. 73-77.

[10] G. M. Koien, "Mutual entity authentication for lte," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*, 2011, pp. 689-694.

[11] K. A. Alezabi, F. Hashim, S. J. Hashim, and B. M. Ali, "An efficient authentication and key agreement protocol for 4G (LTE) networks," in *Region 10 Symposium, 2014 IEEE*, 2014, pp. 502-507.

[12] A. K. Rai, V. Kumar, and S. Mishra, "An efficient password authenticated key exchange protocol for WLAN and WiMAX," in *Proceedings of the International Conference & Workshop on Emerging Trends in Technology*, 2011, pp. 881-885.

[13] C. Cremers, "Scyther," *Semantics and Verification of Security Protocols, Thesis, University Press Eindhoven,* 2006.

[14] C. Cremers, "Scyther User Manual, Department of Computer Science, University of Oxford," ed.

[15] 3GPP, "3GPP TS 35.205 3G Security; Specification of the MILENAGE algorithm set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 1: General," ed, 2014.

[16] 3GPP, "3GPP TS 35.206 3G Security; Specification of the MILENAGE algorithm set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 2: Algorithm specification," ed, 2014.

[17] 3GPP, "3GPP TS 35.207 3G Security; Specification of the MILENAGE algorithm set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 3: Implementors' test data," ed, 2014.

[18] 3GPP, "3GPP TS 35.208 3G Security; Specification of the MILENAGE algorithm set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 4: Design conformance test data," ed, 2014.

[19] J. Arkko, V. Lehtovirta, and P. Eronen, "Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA\ x27)," 2009.