Journal of Ambient Intelligence and Humanized Computing (2020) 11:2163–2185
https://doi.org/10.1007/s12652-019-01248-8

ORIGINAL RESEARCH

An improved authentication and security scheme for LTE/LTE-A networks

Prabhat Kumar Panda[1] · **Sudipta Chattopadhyay**[1]

Received: 17 August 2018 / Accepted: 9 February 2019 / Published online: 23 February 2019

Abstract
Long term evolution (LTE) and LTE-Advanced networks support highly developed authentication and encryption mechanisms. However, these systems still suffer from various security problems such as replay attack, impersonation attack, known key attack, eavesdropping attack and so on. To mitigate these security weaknesses, an improved authentication and security scheme has been proposed for LTE/LTE-A networks. The proposed scheme employs Elliptic Curve Cryptography (ECC), Elliptic Curve Diffie–Hellman (ECDH) and Salsa20 algorithm to improve end to end security and provide faster data transmission for 4G environment. The proposed scheme uses several powerful encryption techniques and also provides proper mutual authentication between User Equipment (UE) and Message Management Entity (MME). The performance of the proposed system has been compared with LTE-A and existing systems in terms of several security attributes and performance parameters. The comparative results show that the proposed scheme outperforms LTE-A as well as other existing schemes.

Keywords Authentication · Elliptic curve cryptography · LTE-A · Salsa20 stream cipher · Security · Shared key

1 Introduction

The rapid development of mobile communication technology demands for various multimedia applications such as multimedia online gaming, video and audio streaming, mobile TV etc., which involves high usage of data. To meet these requirements, 3rd Generation Partnership Project (3GPP) evolved prominent widespread technologies such as LTE and LTE-A technologies for the next generation mobile wireless communication networks or 4G standard (Akyildiz et al. 2010; Cao et al. 2014). The LTE system (Cao et al. 2014) mainly provides high data rates, flexible bandwidth and low access latency. It also improves the coverage as well as capacity of the system. It supports the flexible integration with other wireless communication networks as well. LTE-A provides much higher data rates, throughput, coverage, spectral efficiency and lower latency than the existing LTE (Akyildiz et al. 2010; Cao et al. 2014). To secure the high speed LTE network, an Authentication and Key Agreement (AKA) scheme called Evolved Packet System AKA (EPS-AKA) was used in LTE system (Alezabi et al. 2014; Lai et al. 2013). However, LTE system still suffers from various security issues such as replay attack, Denial of Service (DoS) attack, eavesdropping attack, impersonation attack, known key attack etc. Another drawback of LTE technology is that it does not provide perfect forward secrecy.

Peyravian and Zunic (2000) proposed a secure scheme for password protection and password update by employing 'collision resistant one way hash function' without using any symmetric or public key encryption technique. Meanwhile, Hwang and Yeh (2002) presented an enhanced version of the proposed scheme Peyravian and Zunic (2000) by using public key cryptosystem. In this paper, the authors identified that the scheme described in Peyravian and Zunic (2000) suffered from password guessing attack, data eavesdropping attack and server spoofing attack. These security issues were rectified and subsequently mutual authentication was achieved in Hwang and Yeh (2002). One major drawback of the scheme Hwang and Yeh (2002) was that it was not free from DoS attack. Another demerit was that it could not provide perfect forward secrecy. To overcome these difficulties, Lin and Hwang (2003) developed an enhanced system based on the Diffie–Hellman key ex-change algorithm. In

✉ Prabhat Kumar Panda
prabhatjdvu@gmail.com

Sudipta Chattopadhyay
sudiptachat@yahoo.com

[1] Department of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata 700032, India

the meantime, Zhu et al. (2008) also pointed out that the scheme Hwang and Yeh (2002) was vulnerable to replay attack, impersonation attack, DoS attack and stolen-verifier attack. In Zhu et al. (2008) scheme, the authors proposed an improved password authentication system based on strong hash functions to mitigate the above security issues. However, this scheme was prone to impersonation attack. Islam and Biswas (2013) analyzed the scheme proposed in Lin and Hwang (2003) and identified that it suffered from various attacks such as, insider attack, impersonation attack, stolen-verifier attack, many logged in users attack and known session specific temporary information attack. To eliminate these security flaws, the authors developed an ECC based improved password authentication and updated scheme. The authors in Islam and Biswas (2013) claimed that their proposed scheme brought a considerable improvement in scheme Lin and Hwang (2003). Moreover, the work described in Islam and Biswas (2013) removed many of the security weaknesses of the scheme Zhu et al. (2008) and established that the proposed scheme Islam and Biswas (2013) was protected from all related attacks. Afterwards, Li (2013) analyzed the scheme described in Islam and Biswas (2013) and pointed out that it could get affected by stolen verifier attack, password guessing attack and insider attack. In Li (2013), the author removed these security flaws by proposing a new password authentication and updated scheme based on ECC with smart cards in two different versions. However, Xu and Wu (2015) identified that two versions of the scheme Li (2013) could not provide enough security. To enhance the security as described in Li (2013), the authors proposed an improved scheme by employing ECC with user anonymity in Xu and Wu (2015).

An AKA scheme called Evolved Packet System AKA (EPS-AKA) (Alezabi et al. 2014; Lai et al. 2013) was proposed by 3GPP to secure LTE network. Lai et al. (2013) found that the EPS-AKA protocol was associated with some security problems, such as, lack of privacy preservation and Key Backward/Forward Secrecy (KBS/KFS). It also faced a big challenge for group based authentication. To address these security related issues, the authors presented a Secure and Efficient AKA protocol named SE-AKA, based on ECDH and an asymmetric key cryptosystem. The asymmetric key cryptosystem provided privacy preservation; whereas, ECDH provided KBS/KFS for the system. Moreover, it could effectively authenticate group devices by providing a group authentication mechanism. However, the system failed to authenticate the group of devices. Another Efficient EPS-AKA protocol called EEPS-AKA was developed by Alezabi et al. (2014) based on Simple Password Exponential Key Exchange (SPEKE) (Jablon 2013). The authors in Alezabi et al. (2014) identified that the EPS-AKA protocol had the possibility of getting affected by some security issues, such as, Man in the Middle (MITM) attack, disclosure of the user identity, authentication delay and computational overhead. The authors in Alezabi et al. (2014) established that their proposed scheme was efficient enough to overcome these security problems. Moreover, the authors claimed that the EEPS-AKA was faster than previously developed methods due to the employment of secret key method into it. The proposed method also reduced the storage overhead and authentication delay effectively. Furthermore, the formal verifications showed that the proposed protocol was secure from both active and passive attacks. In the context of EPS-AKA, Abdrabou et al. (2015) showed that the said protocol was vulnerable to replay attack, DoS attack, MITM and disclosure of the user identity. To overcome these weaknesses, the authors proposed a Modified EPS-AKA (MEPS-AKA) protocol based on SPEKE and symmetric key cryptography. It was found that the execution time for MEPS-AKA was more than the EPS-AKA. To mitigate the security weakness of LTE networks, an improved technique called enhanced AKA was approached by Degefa et al. (2016) without adding any extra cost to the environment. The authors employed the secret key cryptographies to enhance the security, computation and communication cost of the LTE networks. However, the scheme assumed that the secret function $f()$ would be kept secret even if the Home Subscriber Server (HSS) is compromised, which is more impractical. Moreover, the scheme could not achieve key forward secrecy (Chien 2018). In 2017, Hamandi et al. (2017) developed a computationally efficient privacy enhanced scheme for LTE networks. To reduce the overhead, the authors minimized the use of asymmetric and symmetric encryptions. However, the scheme was found to be vulnerable to DoS attack, replay attack and could not provide perfect forward secrecy (Singh and Shrimankar 2018). Several improved versions of EPS-AKA were proposed in (Cao et al. 2012; Køien 2011; Singh and Shrimankar 2018; Xiehua and Yongjun 2011) which pointed out different drawbacks associated with EPS-AKA and afterwards removed them by using different cryptographic techniques.

To address the security issues present in two security protocols namely, Internet Protocol Security (IPsec) and Security Socket Layer (SSL), Huang et al. developed a secure communication system defined as Wireless Security System with Data Connection Core (WiSDC) in Huang et al. (2012). This system adopted the Data Connection Core (DCC) as its security base to protect the secrecy, integrity and authenticity of the transmitted messages. To increase the security level of the system, the authors introduced three mechanisms. Firstly, to protect the DCC from hackers, the system produced internal keys in order to derive the communication keys, which were transmitted through medium rather than DCC. Secondly, to lower the probability of information being captured, the system reduced the key exchange level. Finally, to encrypt and decrypt the transmitted message,

it employed two dimensional stream cipher technique. A secure authentication scheme called Security system with Pseudo random number generator, Diffie–Hellman algorithms and Data Connection Core (SPDiD) was proposed by Huang et al. (2013) for wireless environment. The system employed DCC to establish a strong connection between UE and HSS and employed Diffie-Hellman algorithm to exchange common secret keys. Moreover, Pseudo Random Number Sequences (PRNSs) were used to generate more symmetric keys for the purpose of encrypting the key and messages without reducing the security levels. Further, the authors compared the performance of the proposed SPDiD with LTE-A and WiMAX systems which showed that the proposed system provided better security than the existing systems in terms of forgery attack, reply attack, eavesdropping attack and DoS attack. Another novel Security Scheme for 4G Environment called Se4GE was developed by Huang et al. (2014). To overcome some of the security issues found in LTE-A such as replay attack and eavesdropping attack, the system integrated RSA and DH algorithm. This work analytically showed that the security level of Se4GE was higher than LTE-A system though the authentication phase required longer processing time. It was also found that the scheme suffered from some security attacks like impersonation attack and known key attack. Related to this work, Kanani et al. (2014) proposed a modified security scheme based on symmetric key, RSA, random number generator and Se4GE. In this work, the authors analyzed the Se4GE scheme and modified it by providing secured DCC in order to bring improvement in the performance of the said system. The authors also claimed that the proposed system achieves better security than the Se4GE system. However, it was observed that the proposed scheme was not immune to impersonation attack and known key attack.

Meanwhile, many secure authentication schemes were also proposed for the LTE environment. Abdeljebbar and Kouch (2018) established an improved EPS-AKA to provide a new solution to remove the security weakness of LTE network. The scheme protected the key exchange messages by the use of asymmetric cryptographic. However, this scheme was incapable to prevent the DoS attack because of the fact that the scheme did not use any authentication mechanisms to protect some of the transmitted messages. To overcome the security issues found in the existing AKA schemes, several group based efficient and secure AKA scheme for Machine to Machine Communication (MTC) in LTE/LTE-A networks was established by (Gupta et al. 2018; Parne et al. 2018). Both of the schemes used a symmetric cryptosystems and adopted group authentication techniques to verify the group of Machine Type Communication Devices (MTCDs) simultaneously. Ferrag et al. (2018) made a survey on the security for 4G and 5G cellular networks. The authors analyzed different existing privacy models of 4G and 5G networks with respect to several security attributes and performance parameters. Zikria et al. (2018a) analyzed the requirements and challenges for software's, protocols design and valid techniques for the emerging techniques Internet of Things (IoT). The authors reviewed several papers related to the research trends in IoT. Several secure authentication mechanisms and surveyed work for 4G/5G enabled IoT were also presented in (Kumari et al. 2018; Ni et al. 2018; Zikria et al. 2018b; Musaddiq et al. 2018).

To meet the above research demands mainly in the area of LTE/LTE-A, an improved authentication and security scheme for LTE/LTE-A networks has been proposed in this paper. The important contributions of this paper are summarized as follows:

1. The proposed system employs ECC, ECDH and stream cipher Salsa20 algorithm to mitigate the security weaknesses related to 4G wireless system.
2. This scheme adopts ECC and ECDH to protect the system from different security attacks and also improves the key exchange flow between UE and MME, which enhances the security level of the system.
3. The system employs Salsa20 stream cipher and modifies it for the purpose of the encryption and decryption of the plain text and cipher text, which makes the system more secure and faster.
4. The proposed scheme uses timestamp to protect the system from the replay attack and Hash based Message Authentication Code (HMAC) ensures the authenticity, integrity and certification of the transmission messages.
5. The proposed scheme also uses some sophisticated encryption functions to hide important parameters and achieve proper mutual authentication between UE and MME.
6. Security analysis of the proposed system has been carried out in detail to evaluate its performance with respect to LTE standard and some related existing work in terms of several security attributes, such as, replay attack, known key attack, impersonation attack, eavesdropping attack, DoS attack, many logged in user attack and perfect forward secrecy.
7. The effectiveness of the proposed system has been established by comparing the performance of our proposition with other related systems in terms of key generation time, encryption and decryption time, computational cost, total computational time, time complexity and storage overhead. The performance analysis establishes the supremacy of the proposed scheme over other existing schemes.

The rest of this paper is structured as follows. In Sect. 2, we have discussed the technical background relevant to this work. In Sect. 3, we have analyzed the methodology of the

proposed system. In Sect. 4, we have analyzed various security attributes related to the proposed system and compared its performance with LTE standard and some existing related work. In Sect. 5, the performance of the proposed system has been analyzed. Finally, some concluding remarks and outline for future work have been included in Sect. 6.

2 Theoretical background

2.1 Long term evaluation advance (LTE-A)

LTE-A is (Akyildiz et al. 2010) considered as a well-accepted standard for 4G wireless environments. The key objectives of LTE-A are to provide high data rate, wide scalable bandwidth, low latency and improved spectral efficiency (Cao et al. 2014).

The LTE-A comprises of following important components:

(a) *User Equipment (UE)* It is the user device, which consists of different mobile equipment's.
(b) *Evolved Node B (eNodeB or eNB)* eNB is a base station that controls the mobiles in different cells.
(c) *Mobility Management Entity (MME)* MME acts as a bridge between UE and HSS. It controls the high level operation of the mobile. It is also responsible for authentication and data transfer.
(d) *Home Subscriber Server (HSS)* A central data base that contains information about the entire serving subscriber. UE authentication is one of the major responsibilities of HSS.

The detailed analysis of the LTE-A architecture has been further discussed in (Akyildiz et al. 2010). This paper follows the communication flow of LTE-A architecture and considers the LTE-A as the physical network platform for analyzing end to end security and performance of the network.

2.2 Elliptic curve cryptography (ECC)

This cryptography technique was proposed by Miller and Koblitz in 1985 to design public key cryptosystem, which lies on the algebraic structure of elliptic curves over finite fields Z_q (Hankerson et al. 2004). It is currently used in various cryptographic systems to provide better security and computational efficiency. The security of the ECC is mainly dependent on the hardness involved in solving Elliptic Curve Discrete Logarithm Problem (ECDLP). Moreover, it uses smaller key bits to achieve equivalent level of security same as RSA (Hankerson et al. 2004) i.e. the 160-bit elliptic curve key provides the same level of security as 1024-bit RSA key (Mahto et al. 2016). Furthermore, the computational cost of elliptic curve point multiplication is less expensive as compared to the modular exponentiation which is involved in RSA (Chung et al. 2007). Hence, to achieve the better security and provide efficient performance, the proposed scheme adopts ECC over other cryptography techniques. An overview of ECC is presented below.

A set of elliptic curve points $E_q(a, b)$ over a finite field Z_q is the all pairs of integers (x, y) that satisfy the equation $y^2 \bmod q = x^3 + ax + b(\bmod q)$ together with O, called the point at infinity. Where, q is a large prime number and a and b are two constants such that $a, b \in Z_q$ and satisfies the condition of $4a^3 + 27b^3 \neq 0$. The additive cyclic group is defined by $E_g = \{(x, y) \in E_q(a, b)\} \cup \{O\}$. The point multiplication on the cyclic group is calculated by repeated addition. A point P is the public point of the elliptic curve group with order n such that $n \cdot P = 0$. The further details of the elliptic curve cryptosystems properties are described in (Hankerson et al. 2004).

The computational problems over the elliptic curve group which are normally used to design secure cryptographic systems have been analyzed below (Hankerson et al. 2004; Xu et al. 2018):

Elliptic curve discrete logarithm problem (ECDLP) Given P, $Q \in E_g$, hard to find an integer $m \in [1, n\text{-}1]$, such that $Q = m \cdot P$.

Computational Diffie–Hellman Problem (CDHP) For $a, b \in [1, n\text{-}1]$, given P, aP and bP, hard to compute abP.

Decisional Diffie–Hellman Problem (DDHP) For $a, b, c \in [1, n\text{-}1]$, given P, aP, bP and cP, difficult to decide whether $c = ab \bmod q$ or not.

2.3 Salsa20 algorithm

A stream cipher algorithm Salsa20 is one of the eSTREAM candidates proposed by Bernstein (2008). This is recommended for the design of cryptographic schemes where speed and security both are of prime importance. Salsa20 is also suggested for the quantum resistant algorithm (Cheng et al. 2017). The core of Salsa20 is a hash function having an input of 64-byte to produce an output of 64-byte. Mathematical operations such as addition, X-OR and constant distance rotation are used to construct the Salsa20 algorithm. Moreover, the keystream of 64-byte is obtained by mapping 32-byte secret key, 8-byte nonce, and 8-byte block number. Furthermore, it goes through several rounds to obtain 64-byte key stream, which is depicted in Fig. 1 (Afdhila et al. 2016; Bernstein 2005). Salsa20 encrypts a k-byte of plain text by performing X-OR operation with the first k-byte of keystream and discarding the remaining stream. Similarly, it decrypts the k-byte of cipher text by performing the X-OR operation with the first k-byte of the key stream to generate the plain text. To achieve secure and faster system, the

proposed scheme employs Salsa20 algorithm and modifies it for the purpose of encryption and decryption of the data. The Salsa20 encryption and decryption process is illustrated in Fig. 2 (Afdhila et al. 2016; Bernstein 2005).

3 Proposed model

In this section, a security system based on ECC and Salsa20 algorithm has been proposed to enhance the end to end security of 4G environment. The notations which are used in this study have been presented in Table 1 and the relevant functions have been defined in Sect. 3.1.

3.1 Functions

1. Encryption functions:
   a. $Enc\_fun(p, q) = p \oplus q$.
   b. $ECC\_Enc(a, b) = b * a$.

   Where, a represents the point function (a_x, a_y)
2. Decryption functions:
   a. $Dec\_fun(p, q) = p = e \oplus q$.

   where, $e = Enc\_fun(a, b)$
   b. $ECC\_Dec(a, b) = a = d * f$.

   where, $d = minv(b, n)$ and $f = ECC\_Enc(a, b)$

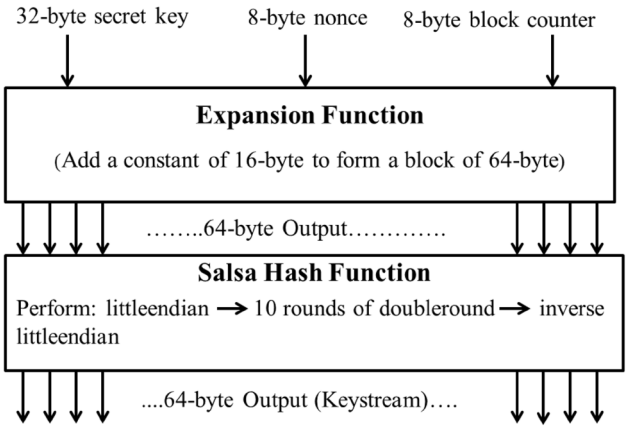


Fig. 1 Salsa20 keystream generation process

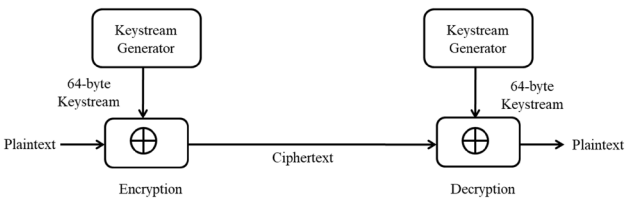


Fig. 2 Encryption and decryption process of Salsa20

3. HMAC (K) = A hash based message authentication code. The hash function performs both on secret key K and transmitted message to generate HMAC. It is used to ensure the authenticity, integrity and certification of the transmitting and receiving messages.

As for example, if a message which is transmitted from UE to MME is $(OP_{code}, T_{UE}, IMSI_A, Enc\_fun(E_U, K_A), ECC\_Enc(U_A, K_P), ECC\_Enc(P_{U_A}, A_R))$ then the authentication code generated by performing hash function on both the key $(K_P + K_A \oplus A_R)$ and the message $(OP_{code}, T_{UE}, IMSI_A, Enc\_fun(E_U, K_A), ECC\_Enc(U_A, K_P), ECC\_Enc(P_{U_A}, A_R))$ is found to be $HMAC(K_P + K_A \oplus A_R)$.

An important point of discussion related to our proposed model is that the keys which are generated by ECC are the pair of numbers i.e. point function. Whenever these keys are used as a session key for generating different keys or for traditional encryption, a single number is used which is generated by performing the XOR operation between the two numbers.

3.2 Communication steps

In this study, a distinctive Operation Code (OP_{code}) has been assigned to individual message to describe the function of each message. The various OP_{code} used in this model reduce the authentication time and operational complexity. The definitions of various OP_{code} have been described in Table 2.

The operational flow diagram of the proposed model has been presented in Fig. 3. To achieve an end to end secure communication, the proposed model has been categorized into three phases:

1. Registration phase
2. Authentication and key exchange phase
3. Data transmission phase.

In the registration phase, at first the user get registered himself in the server HSS with his own parameters and subsequently collects the server public key. Next, the server stores each legal user's parameters into a write protected file. Afterwards, the authentication and key exchange process starts.

In the authentication and key exchange phase, initially UE transmits an authentication request message to MME which contains encrypted keys with UE's identity. Upon receiving the authentication request message, MME sends a request to HSS for the encrypted private key and password verifier for user, based on the identity of the respective user. Subsequently, HSS sends those parameters to MME. After receiving; MME decrypts all the keys and authenticates the user by verifying the authentication parameter of UE. Then MME will validate the received message to check whether

Table 1 Notations used in the proposed system

Notations	Descriptions
E	An elliptic curve equation
$E_q(a, b)$	An elliptic curve
E_g	An elliptic curve group over E
P	Public point of the elliptic curve group with order n such that $n \cdot P = 0$
q, n	Large prime numbers
Z_q	A finite field over a large prime number q
$IMSI_A$	International Mobile Subscriber Identity for user A
PW_A	Password of user A
RS	Private key of the server HSS, select from [1, n-1]
P_S	Public key of the server, where $P_S = RS \cdot P$
V_A	Password verifier of user A, where $V_A = PW_A \cdot P$
K_P	Private key computed either using $K_P = PW_A \cdot P_S = (K_x, K_y) = K_x \oplus K_y$ or $K_P = RS \cdot V_A = (K_x, K_y) = K_x \oplus K_y$
RU_A	Private key of UE, select from [1, n-1]
RM	Private key of MME, select from [1, n-1]
P_{U_A}	Public key of UE, where $P_{U_A} = RU_A \cdot PW_A \cdot P_S = RU_A \cdot PW_A \cdot RS \cdot P$
P_M	Public key of MME, where $P_M = RM \cdot P_S = RM \cdot RS \cdot P$
A_R	User authentication random number select from [1, n-1]
E_U	Encrypted authentication random number of user A
U_A	Authentication parameter of user A
A_M	MME authentication random number select from [1, n-1]
E_M	Encrypted authentication random number of MME
M_A	Authentication parameter of MME
$A_{R,C}$	Computed authentication random number of MME
$U_{A,C}$	Computed MME key used to authenticate user A
$A_{M,C}$	Computed authentication random number of user A
$M_{A,C}$	Computed key of user A used to authenticate MME
K_S	Shared key individually generated by UE and MME
IK_i	Internal derived keys used by UE and MME themselves without sending them through wireless channel, $1 \leq i \leq 9$
TEK_i	Set of Traffic encryption keys, $1 \leq i \leq 81$
S_k	Secret key of Salsa20
S_n	Nonce of Salsa20
S_b	Block number of Salsa20
KS	Keystream of Salsa20

the message is valid or not. If not valid, MME terminates the message; otherwise, MME sends an authentication reply message to UE which contains encrypted keys. By receiving the authentication reply message, UE decrypts the keys and authenticates MME by verifying the authentication parameter of MME. If the received authentication parameter is proper, then mutual authentication is achieved. Next, it checks the correctness of the message. If the message is not a correct one, it discards the message; otherwise, the process goes to data transmission phase.

Table 2 Definitions of various OP_{code}

OP_{code}	Process	Description
1	Authentication request	Sent to MME by UE
2	Authentication reply	Sent to UE by MME
3	Authentication reject	Sent to UE by MME
4	Data transmission request	Sent to MME by UE
5	Data transmission reply	Sent to UE by MME
6	Data transmission reject	Sent to UE by MME
7	Data delivery	Sent to MME by UE
8	Data receiving	Sent to UE by MME
0, 9–15	Reserved	For future use

In the data transmission phase, UE sends a data transmission request to MME. On receiving the request message, MME checks whether the received message is valid or not. If it is not valid, the MME terminates it; else, MME sends a data transmission reply message to UE to make a confirmation that a secure communication can be established. On receiving the confirm message, UE starts delivering the encrypted data to MME in a secure communication channel. MME retrieves the plaintext by decrypting the data and afterwards the data exchange process continues. The complete process has been explained mathematically as follows:

3.2.1 Registration phase

At the initial stage of the network entry, the user registers himself to the server HSS with his own parameters such as identity of the user i.e. $IMSI_A$ and password verifier V_A and subsequently collects the server's public key P_S. Afterwards, the server stores each legal user's identity, password verifier and a status bit into a write protected file as presented in Table 3. Here, the status bit represents the present status of the user i.e. when the user is logged into the server, the status bit is set to one ('1'), else it is set to zero ('0').

3.2.2 Authentication and key exchange phase

In this phase, at first, UE transmits an authentication request message to MME that includes encrypted keys. After receiving the authentication request message, MME sends request to HSS for the encrypted private key K_P and password verifier V_A for user A based on $IMSI_A$. Subsequently, HSS delivers K_P and V_A to MME. MME decrypts the keys and authenticates the user by verifying the condition $U_{A,C} \stackrel{?}{=} U_A$. If this condition is not satisfied, MME terminates the session; else it authenticates UE and then verifies the correctness of the message by comparing the HMAC value i.e. $HMAC(K_P, K_A, A_R)_c \stackrel{?}{=} HMAC(K_P, K_A, A_R)_r$. Here, subscripts 'c' and 'r' are used to represent the calculated and retrieved

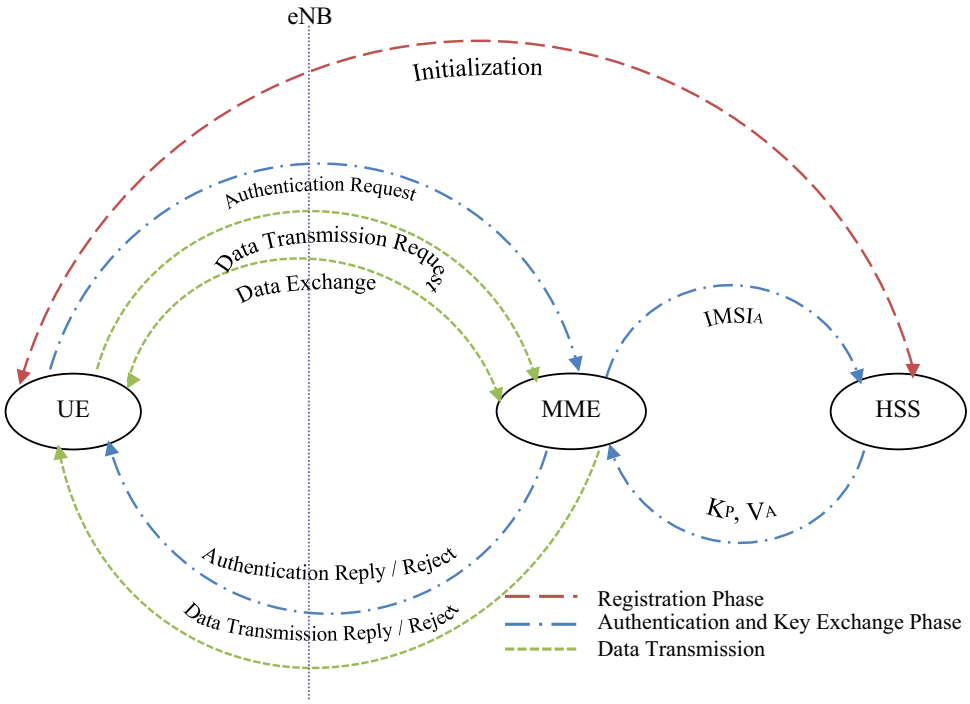


Fig. 3 Operational flow diagram of the proposed system

HMAC values respectively. If the above stated condition is not satisfied, MME discards the message; otherwise, it sends an authentication reply message to UE which contains encrypted keys. On receiving authentication reply message, UE decrypts the encrypted keys and authenticates MME by verifying the condition $M_{A,C} \stackrel{?}{=} M_A$. If the condition does not fulfill, UE discards the message; else, it authenticates the MME. Thus, the mutual authentication is achieved. Next, UE checks the correctness of the message by verifying the condition $HMAC(IK_1, IK_2, IK_3)_c \stackrel{?}{=} HMAC(IK_1, IK_2, IK_3)_r$. If the condition is not satisfied, the process is terminated; otherwise, it is forwarded to the data transmission phase.

3.2.3 Data transmission phase

In this phase, UE sends a data transmission request message to MME. On receiving the request message, MME checks the correctness of the data transmission request message by verifying the condition $HMAC(IK_4, IK_5, IK_6)_c \stackrel{?}{=} HMAC(IK_4, IK_5, IK_6)_r$. If this condition is not satisfied, MME discards the message; otherwise, it generates the dynamic keys such as, DK_{1-9},DX_{1-9} and TEK_{1-81}, and then sends a data transmission reply message to UE to confirm that a secure communication can be established. On receiving the confirmation, UE starts transmitting the encrypted data to MME in a secure communication channel. The data transmission process has been analyzed as follows: For example, assume that plaintext message $= m_0 m_1 m_2 m_3 \ldots ..m_{l-1}$ is divided into l number of blocks of the same size. Correspondingly, the ciphertext message $= C_0 C_1 C_2 C_3 \ldots ..C_{l-1}$ is generated and transmitted to MME. On receiving the ciphertext message, MME retrieves the plaintext by decrypting the data and subsequently data exchange process continues. The complete process is described in Sect. 3.2.4 in the form of algorithm.

Table 3 The verifier table with user status bit

User Identity	Password verifier	Status- bit
$IMSI_A$	$V_A = PW_A \cdot P$	0/1
$IMSI_B$	$V_B = PW_B \cdot P$	0/1
$IMSI_C$	$V_C = PW_C \cdot P$	0/1
–	–	–

3.2.4 Proposed algorithm

1. Registration_Phase ()

INPUT: n, P, $IMSI_A$

OUTPUT: V_A, P_S

Begin

Procedure ($n, P, IMSI_A, V_A, P_S$)

1. UE selects a password PW_A, $PW_A \leftarrow$ Random [1, n-1]
2. UE computes password verifier V_A, $V_A \leftarrow PW_A \cdot P$
3. HSS selects a random number RS, $RS \leftarrow$ Random [1, n-1]
4. HSS computes server public key P_S, $P_S \leftarrow RS \cdot P$
5. User gets registered to server HSS, HSS $\leftarrow$ UE ($IMSI_A, V_A$)
6. UE collects the server public key, UE $\leftarrow$ HSS (P_S)
7. If the user is logged into the server, then Status bit = '1'
   Else
   Status bit= '0'
   End If

End Procedure

End

2. Authentication _Phase ()

A. Authentication _Request _UE ()

$T_{UE} \leftarrow$ Time at which message 1 (M1) is sent from UE side

INPUT: n, P, P_S, V_A, PW_A, T_{UE}, $IMSI_A, OP_{code}$

OUTPUT: P_{U_A}, E_U, U_A

Begin

Procedure (n, P, P_S, V_A, PW_A, T_{UE}, $IMSI_A, OP_{code}$, P_{U_A}, E_U, U_A)

1. Select private key RU_A, $RU_A \leftarrow$ Random [1, n-1]
2. Compute public key P_{U_A}, $P_{U_A} \leftarrow RU_A \cdot PW_A \cdot P_S \leftarrow RU_A \cdot PW_A \cdot RS \cdot P$
3. Compute encrypted private key K_P, $K_P \leftarrow PW_A \cdot P_S \leftarrow PW_A \cdot RS \cdot P \leftarrow (K_x, K_y) \leftarrow K_x \oplus K_y$
4. Derive K_A from V_A, Where, $V_A \leftarrow PW_A \cdot P \leftarrow (V_{Ax}, V_{Ay})$, then $K_A \leftarrow V_{Ax} \oplus V_{Ay}$
5. Select an authentication random number A_R, $A_R \leftarrow$ Random [1, n-1]
6. Compute E_U and U_A, $E_U \leftarrow A_R \oplus K_P + K_A$ and $U_A \leftarrow A_R \cdot P$
7. Send authentication request message from UE to MME (M1)
   M1: $(OP_{code}, T_{UE}, IMSI_A, Enc\_fun(E_U, K_A), ECC\_Enc(U_A, K_P), ECC\_Enc(P_{U_A}, A_R), HMAC(K_P, K_A, A_R))$

End Procedure

End

B. Authentication _Reply _MME ()

1. $T_{R1} \leftarrow$ Time at which message 1 (M1) is received at MME
2. $\Delta T_1 \leftarrow$ A predefined threshold
3. $T_{MME} \leftarrow$ Time at which message 2 (M2) is sent from MME side

INPUT: n, P, $IMSI_A$, RS, V_A, ΔT_1, T_{R1}, T_{MME}, OP_{code}

OUTPUT: P_M, E_M, M_A, IK_3

Begin

Procedure (n, P, $IMSI_A$, RS, V_A, ΔT_1, T_{R1}, T_{MME}, OP_{code}, P_M, E_M, M_A, IK_{1-3})

If ($T_{R1} - T_{UE} \leq \Delta T_1$)

1. MME sends request to HSS for K_P and V_A based on $IMSI_A$
2. HSS computes K_P, $K_P \leftarrow RS \cdot V_A \leftarrow RS \cdot PW_A \cdot P \leftarrow (K_x, K_y) \leftarrow K_x \oplus K_y$
3. MME $\leftarrow$ HSS (K_P, V_A)
4. Decrypt U_A and E_U by using $ECC\_Dec()$ and $Dec\_fun()$
5. Calculate $A_{R,C}$ and $U_{A,C}$, $A_{R,C} \leftarrow (E_U - K_A) \oplus K_P$ and $U_{A,C} \leftarrow A_{R,C} \cdot P$
6. If ($U_{A,C} == U_A$)

   MME authenticates user and then decrypts P_{U_A} by using $ECC\_Dec()$

   Else

   MME terminates the session

   End If
7. If ($HMAC(K_P, K_A, A_R)_c == HMAC(K_P, K_A, A_R)_r$)
   a. MME selects RM, $RM \leftarrow$ Random [1, n-1]
   b. Calculate P_M, $P_M \leftarrow RM \cdot P_S \leftarrow RM \cdot RS \cdot P$
   c. Compute K_S, $K_S \leftarrow RM \cdot P_{U_A} \leftarrow RM \cdot RU_A \cdot PW_A \cdot RS \cdot P \leftarrow (K_{Sx}, K_{Sy}) \leftarrow (K_{Sx} \oplus K_{Sy})$
   d. Choose a random number A_M, $A_M \leftarrow$ Random [1, n-1]
   e. Compute E_M and M_A, $E_M \leftarrow A_M \oplus K_P + A_R$ and $M_A \leftarrow A_M \cdot P$
   f. Derive internal keys IK_1, IK_2 and IK_3,
      i. $IK_1 \leftarrow [(A_M \oplus K_P) + K_S] \oplus (K_S + A_R)$
      ii. $IK_2 \leftarrow [(A_M \oplus K_P) + IK_1] \oplus (IK_1 + A_R)$
      iii. $IK_3 \leftarrow [(A_M \oplus K_P) + IK_2] \oplus (IK_2 + A_R)$

   Else

   MME discards the message

   End If
8. MME sends authentication reply message (M2) to UE

   M2: $(OP_{code}, T_{MME}, Enc\_fun(E_M, K_P), ECC\_Enc(M_A, A_R), ECC\_Enc(P_M, A_M), HMAC(IK_1, IK_2, IK_3))$

Else

MME terminates the connection

End If

End Procedure

End

C. Reply _Check _UE ()

1. $T_{R2} \leftarrow$ Time at which message 2 (M2) is received at UE side
2. $\Delta T_2 \leftarrow$ Predefined threshold

INPUT: n, P, A_R, K_P, RU_A, PW_A

OUTPUT: IK_3

Begin

Procedure (n, P, A_R, K_P, RU_A, PW_A, IK_{1-3})

If ($T_{R2} - T_{MME} \leq \Delta T_2$)

1. Decrypt E_M and M_A by using $Dec\_fun()$ and $ECC\_Dec()$
2. Calculate $A_{M,C}$ and $M_{A,C}$, $A_{M,C} \leftarrow (E_M - A_R) \oplus K_P$ and $M_{A,C} \leftarrow A_{M,C} \cdot P$
3. If ($M_{A,C} == M_A$)
   UE authenticates MME and then mutual authentication is achieved
   Else
   UE terminates the connection
   End If
4. Decrypt P_M by using decrypt function $ECC\_Dec()$
5. Compute K_S, $K_S \leftarrow RU_A \cdot PW_A \cdot P_M \leftarrow RU_A \cdot PW_A \cdot RM \cdot RS \cdot P \leftarrow (K_{Sx}, K_{Sy}) \leftarrow K_{Sx} \oplus K_{Sy}$
6. Derive internal keys IK_1, IK_2 and IK_3,
   a. $IK_1 \leftarrow [(A_M \oplus K_P) + K_S] \oplus (K_S + A_R)$
   b. $IK_2 \leftarrow [(A_M \oplus K_P) + IK_1] \oplus (IK_1 + A_R)$
   c. $IK_3 \leftarrow [(A_M \oplus K_P) + IK_2] \oplus (IK_2 + A_R)$
7. If ($HMAC(IK_1, IK_2, IK_3)_c == HMAC(IK_1, IK_2, IK_3)_r$)
   Go to data transmission phase
   Else
   Discard the message
   End If

Else

Terminate the session

End If

End Procedure

End

3. Data _Transmission _Phase ()

A. Data _Transmission _Request _UE ()

INPUT: IK_{1-3}, K_S, P_{U_A}, P_M, OP_{code}

OUTPUT: IK_6

Begin

Procedure (IK_{1-3}, K_S, P_{U_A}, P_M, OP_{code}, IK_{4-6})

1. Generate internal derived keys IK_4, IK_5 and IK_6
   a. $IK_4 \leftarrow [(K_S \oplus IK_1) + IK_2] \oplus (IK_3 + K_S)$
   b. $IK_5 \leftarrow [(K_S \oplus IK_2) + IK_3] \oplus (IK_4 + K_S)$
   c. $IK_6 \leftarrow [(K_S \oplus IK_3) + IK_4] \oplus (IK_5 + K_S)$
2. UE sends a data transmission request message (M3) to MME
   M3: $(OP_{code}, HMAC(IK_4, IK_5, IK_6))$

End Procedure

End

B. Data _Transmission _Reply _MME ()

INPUT: $IK_{1–3}, K_S, P_{U_A}, P_M, OP_{code}$

OUTPUT: $TEK_{1–81}$

Begin

Procedure ($IK_{1–3}$, K_S, P_{U_A}, P_M, OP_{code}, $TEK_{1–81}$)

1. On receiving M3, MME generates internal derived keys IK_4, IK_5 and IK_6
   a. $IK_4 \leftarrow [(K_S \oplus IK_1) + IK_2] \oplus (IK_3 + K_S)$
   b. $IK_5 \leftarrow [(K_S \oplus IK_2) + IK_3] \oplus (IK_4 + K_S)$
   c. $IK_6 \leftarrow [(K_S \oplus IK_3) + IK_4] \oplus (IK_5 + K_S)$
2. If ($HMAC(IK_4, IK_5, IK_6)_c == HMAC(IK_4, IK_5, IK_6)_r$)
   a. Generate internal derived keys IK_7, IK_8 and IK_9
      i. $IK_7 \leftarrow [(IK_1 \oplus IK_4) + IK_5] \oplus (IK_6 + IK_1)$
      ii. $IK_8 \leftarrow [(IK_2 \oplus IK_5) + IK_6] \oplus (IK_7 + IK_2)$
      iii. $IK_9 \leftarrow [(IK_3 \oplus IK_6) + IK_7] \oplus (IK_8 + IK_3)$
   b. Compute dynamic keys $DK_{1–9}$, $DX_{1–9}$ and $TEK_{1–81}$
      i. $DK_i \leftarrow (IK_i \oplus P_{U_A}) + (K_S \oplus P_M)$, $1 \le i \le 9$
      ii. $DX_i \leftarrow (IK_i \oplus P_M) + (K_S \oplus P_{U_A})$, $1 \le i \le 9$
      iii. $TEK_{(i-1)\times 9+j} \leftarrow DK_i + (DX_j \oplus K_S)$, $1 \le i \le 9$ and $1 \le i \le 9$

   Else
      Discard the message and may wait for the valid UE
   End If
3. If (corresponding node (CN) == online)
      MME sends a data transmission reply message (M4) to UE with OP_{code} = 5
      M4: $(OP_{code}, HMAC(IK_7, IK_8, IK_9))$

   Else
      MME sends a data transmission reject message i.e. M4 with OP_{code} = 6
   End If

End Procedure

End

C. Data _Reply _Check _UE ()

INPUT: $IK_{1-3}, IK_{4-6}, K_S, P_{U_A}, P_M, OP_{code}$

OUTPUT: TEK_{1-81}

Begin

Procedure ($IK_{1-3}, IK_{4-6}, K_S, P_{U_A}, P_M, OP_{code}, TEK_{1-81}$)

1. On receiving message, UE checks if ($OP_{code} == 5$)
   a. Generate IK_7, IK_8 and IK_9
      i. $IK_7 \leftarrow [(IK_1 \oplus IK_4) + IK_5] \oplus (IK_6 + IK_1)$
      ii. $IK_8 \leftarrow [(IK_2 \oplus IK_5) + IK_6] \oplus (IK_7 + IK_2)$
      iii. $IK_9 \leftarrow [(IK_3 \oplus IK_6) + IK_7] \oplus (IK_8 + IK_3)$
   b. If ($HMAC(IK_4, IK_5, IK_6)_c == HMAC(IK_4, IK_5, IK_6)_r$)
      i. Generate DK_{1-9}, DX_{1-9} and TEK_{1-81}
         a) $DK_i \leftarrow (IK_i \oplus P_{U_A}) + (K_S \oplus P_M),\ 1 \le i \le 9$
         b) $DX_i \leftarrow (IK_i \oplus P_M) + (K_S \oplus P_{U_A}),\ 1 \le i \le 9$
         c) $TEK_{(i-1)\times 9+j} \leftarrow DK_i + (DX_j \oplus K_S),\ 1 \le i \le 9$ and $1 \le i \le 9$

      Else
         UE discards the fake message and may wait for the correct one
      End If

   Else
      UE discards the fake message and may wait for the correct one
   End If

End Procedure

End

D. Data _Message _Encryption _UE ()

INPUT: $S_k, S_n, S_b, \mathrm{m}_i, K_S, IK_9, TEK_{1-81}, OP_{code}$

OUTPUT: C_i

Begin

Procedure ($S_k, S_n, S_b, \mathrm{m}_i, K_S, IK_9, TEK_{1-81}, OP_{code}, C_i$)

1. Generate Key stream of Salsa20 KS by using Salsa20 algorithm.
2. Generate cipher text message C_i,

   $C_i \leftarrow \mathrm{m}_i \oplus KS_{i \bmod 64} + \mathrm{TEK}_j$, where $0 \le i \le l-1$, $j = (i+k) \bmod 81 + 1, 0 \le k \le 81$,

   and l is the number of message blocks of the same size.
3. UE sends a data message (M5) to MME

   M5: $(OP_{code}, Enc\_fun(k, K_S), Enc\_fun(S_n, k), C_i, HMAC(IK_9, TEK_k, K_S))$

End Procedure

End

Data _Message _Decryption _MME ()

INPUT: S_k, S_b, K_S, IK_9, TEK_{1-81}

OUTPUT: m_i

Begin

Procedure (S_k, S_b, K_S, IK_9, TEK_{1-81}, m_i)

1. Decrypt k by using the decrypt function $Dec\_fun()$
2. Decrypt the nonce of Salsa20 (S_n) by using the decrypt function $Dec\_fun()$
3. Generate Key stream of Salsa20 KS by using Salsa20 algorithm
4. If ($HMAC(IK_9, TEK_k, K_S)_c == HMAC(IK_9, TEK_k, K_S)_r$)

MME obtains the plaintext message m_i by decrypting the cipher text C_i

$$\mathrm{m}_i \leftarrow \begin{cases} (C_i - \mathrm{TEK}_j) \oplus KS_{i \bmod 64}, f \geq \mathrm{TEK}_j \\ (C_i + \overline{\mathrm{TEK}_j} + 1) \oplus KS_{i \bmod 64}, f < \mathrm{TEK}_j \end{cases}$$

Where, $0 \leq i \leq l-1, j = (i+k) \bmod 81 + 1, 0 \leq k \leq 81$

Else

MME discards the message and waits for the correct message delivered by UE

End If

End Procedure

End

4 Security analysis

In this section, we have analyzed different Security Attributes (SA) related to the proposed system and compared them with LTE standard and other related existing systems.

SA1: mutual authentication During authentication and key exchange process of the proposed protocol, UE and MME authenticate each other by verifying the authentication parameters U_A and M_A with the computed authentication parameter $U_{A,C}$ and $M_{A,C}$ of UE and MME respectively. In the authentication reply step, MME authenticates UE by verifying the equality condition between $U_{A,C}$ and U_A. UE derives E_U and U_A by using different parameters K_P, K_A and A_R and then sends them to MME. During the authentication reply step, MME decrypts E_U to generate $A_{R,C}$ by using K_P and K_A, and then computes $U_{A,C}$ which is equal to the received U_A. In the authentication reply check step, UE authenticates MME by verifying the equality condition between $M_{A,C}$ and M_A. By following above similar process, it is found that $M_{A,C}$ and M_A both provide equal computed value. Thus the proposed system achieves mutual authentication. In this context, it can be identified that the process of proper mutual authentication between UE and MME has not been followed in existing literatures such as SPDiD (Huang et al. 2013), Se4GE (Huang et al. 2014), Kanani et al. (2014), MEPS-AKA (Abdrabou et al. 2015) and Hamandi et al. (2017) (Singh and Shrimankar 2018).

SA2: replay attack The proposed system uses time stamp T_{UE} and $HMAC(K_P, K_A, A_R)$ to defend replay attacks. For the purpose of illustration, let us assume that the authentication request message (M1) is duplicated by hackers and sent. In this case, the condition $T_{R1} - T_{UE} \leq \Delta T_1$ will not be satisfied because of the fact that time stamp T_{UE} is set at that time when M1 is sent from UE and ΔT_1 is the predefined threshold. Consequently, the message is discarded by MME. If the hackers modify the time stamp T_{UE} to the present time through some means, then also the computation of $HMAC(K_P + K_A \oplus A_R)$ by the hackers will be incorrect. Therefore, the computed $HMAC(K_P, K_A, A_R)_c$ will not be equal to the received $HMAC(K_P, K_A, A_R)_r$. Similar conclusion can be drawn for M2 as well. All other transmission messages contain HMAC (K) function, whose secret key K is only known to UE and MME. Hence the proposed scheme can defend the replay attack in an efficient manner. However, the scheme Hamandi et al. (2017) does not prevent the replay attack (Singh and Shrimankar 2018).

SA3: impersonation attack Impersonate attack (Xiehua and Yongjun 2011) occurs when the hackers access the security parameters of the users stored in the server. In the proposed scheme, the server may compromise V_A of user. However, with this knowledge of V_A, the hackers cannot decrypt all the keys of authentication request message, which requires the knowledge of K_P, K_A and the random number A_R. Computation of K_P and K_A requires a password PW_A. The hackers may try to extract PW_A from V_A but they fail to do so as it is hard to solve the ECDLP (Hankerson et al. 2004). Therefore, the proposed scheme is more immune to impersonation attack. In contrast, in existing schemes like SPDiD (Huang et al. 2013), Se4GE (Huang et al. 2014) and Kanani et al. (2014) if the server compromises user identity and Data Connection Core (DCC) then the hackers can decrypt all the keys involved in the authentication request message. Thus these schemes can get affected by impersonation attack.

SA4: known key attack When the session ephemeral private keys are accidently exposed to attackers through any means, the attackers can avail all the keys of the system resulting in known key attack (Hamandi et al. 2017). In the proposed scheme, both UE and MME compute a shared key $K_S = RM \cdot RU_A \cdot PW_A \cdot RS \cdot P$. If it is assumed that the ephemeral private keys RM and RU_A are exposed to an attacker then also it is difficult to derive the shared key K_S as it is not easy to extract the knowledge of $PW_A \cdot RS \cdot P$. This is because of the fact that the computation of $PW_A \cdot RS \cdot P$ from the pair $(V_A, P_S) = (PW_A \cdot P, RS \cdot P)$ is equivalent to solving the CDHP, which is difficult to achieve. Hence the proposed system can prevent the known key attack. In the existing scheme Se4GE (Huang et al. 2014), the attacker can easily compute the common secret key $CSK = P_{BR}^{MR} \bmod p = g^{BR \cdot MR} \bmod p$ or $CSK = P_{MR}^{BR} \bmod p = g^{MR \cdot BR} \bmod p$ with the knowledge of the private keys MR and BR corresponding to UE and MME respectively. Above process shows that if the private keys are exposed, the attackers may also compute common secret key CSK in the existing schemes SPDID (Huang et al. 2013) and Kanani et al. (2014). Thus all these existing schemes are not capable enough to prevent the known key attack.

SA5: DoS attack DoS attack (Panda and Chattopadhyay 2019) occurs when the attacker sends the illegal messages to reduce the performance of the network and also makes the resources inaccessible from the intended users. The DOS attack can be avoided by protecting the messages using encryption mechanisms and hashing. The proposed scheme encrypts all the transmitted keys by using eminent encrypted functions to protect all the management messages. Another important feature is that HMAC function is used to validate all the messages. Therefore, any of the illegal messages cannot pass to UE and MME for validation. Thus the system can defend DoS attack successfully. However, in the existing schemes MEPS-AKA (Abdrabou et al. 2015), Hamandi et al. (2017) and Kumari et al. (2018) some of the transmitted keys have not been encrypted and also some of the messages have not been protected by any authentication mechanism. Hence, the schemes MEPS-AKA, Hamandi et al. (2017) and Kumari et al. (2018) can get affected by DoS attack (Singh and Shrimankar 2018). Another scheme namely Improved EPS-AKA (Abdeljebbar and Kouch 2018) was also vulnerable to DoS attack as it did not use any authentication mechanism to protect the transmitted messages.

SA6: Eavesdropping attack In the proposed system, the hackers can get only the transmitted keys from the different encrypted functions such as *Enc_fun*() and *ECC_Enc*() which are adopted in different messages. However, to decrypt the public keys

P_{U_A} and P_M, it is required to compute two authentication random numbers A_R and A_M. Moreover, P_{U_A} is associated with password PW_A which is unavailable to hackers. Similarly, other keys and security parameters which are involved in this communication process are also well protected. Even though the hackers capture the messages from the network, it is not possible to extract the user's keys. Hence the proposed system is able to defend the eavesdropping attack (Panda and Chattopadhyay 2019).

SA7: many logged in user's/device's attack The many logged in user's attack occurs when the identity and password of the legal users/devices are leaked by some means to many hackers, as a result of which they can simultaneously access the accounts of the legitimate users/devices in a remote server. In the proposed system, only single hacker, having the knowledge of proper user identity and password can access the account although many others try to do so. This is because of the fact that whenever a single hacker logs in by using proper user identity and password, the server sets the status bit to '1'. Meantime, if any other hacker tries to log into the server with the same user identity and password, the status-bit indicates that someone is already logged in and the server rejects rest of the attempts. Thus the proposed scheme is safe from many logged in user's attack. As far as the existing schemes SPDiD (Huang et al. 2013), Se4GE (Huang et al. 2014), Kanani et al. (2014), MEPS-AKA (Abdrabou et al. 2015), Enhanced-AKA (Degefa et al. 2016), Hamandi et al. (2017), Kumari et al. (2018), SEGB (Parne et al. 2018), DGBES (Gupta et al. 2018), Improved EPS-AKA (Abdeljebbar and Kouch 2018) and EAKA-EPS (Singh and Shrimankar 2018) are concerned, they are not safe from many logged in users/devices attack as they do not incorporate any concept of setting the login status of the logged user/device.

SA8: perfect forward secrecy Perfect forward secrecy (Alezabi et al. 2014) implies that if the password of the user and secret key of the server are exposed then also the secrecy of the other computed keys should not be affected. As for

example, if the hacker has the knowledge of user password PW_A and server private key RS then it is possible to compute V_A and P_S. Moreover, the hacker may get information about the public keys P_{U_A} and P_M which are decrypted from the messages M1 and M2 respectively. However, it is difficult to compute the shared key $K_S = RM \cdot RU_A \cdot PW_A \cdot RS \cdot P$ as it requires two private keys RU_A and RM which are two random numbers. If someone tries to extract them from the pair $(P_{U_A}, P_M) = (RU_A \cdot PW_A \cdot RS \cdot P, RM \cdot RS \cdot P)$, it is not easy to solve due to hard of CDHP. Hence it can be said that the proposed scheme offers perfect forward secrecy. In contrast, the existing schemes MEPS-AKA (Abdrabou et al. 2015), Enhanced-AKA (Degefa et al. 2016) and Hamandi et al. (2017) do not provide perfect forward secrecy (Chien 2018; Singh and Shrimankar 2018).

Security comparison of the proposed system with the other related systems has been presented in Table 4. Here, different security attributes such as Replay attack, Known key attack, Impersonation attack, Eavesdropping attack, DoS attack, Many logged in user's attack and Perfect forward secrecy have been intensified by "Yes" and "No". Moreover, the degree of Mutual authentication has been indicated by "Partial" and "Full".

5 Performance analysis

In this section, performance analysis of the proposed system and some other related systems has been analyzed and compared. Simulation has been performed using MATLAB 2015a platform. The simulation parameters have been presented in Table 5. Generation of different keys and related functions has been analyzed to examine various security issues related to wireless communication systems. The logical key reasoning for evaluating the time consumptions of several keys and functions have been explained below:

In this work, the time generating different keys and related functions on both UE and MME sides has been evaluated for different key lengths such as 112-bit, 128-bit and 160-bit which provide the security levels of 56-bit, 64-bit and 80-bit respectively (Mahto et al. 2016; Barker 2016). Moreover, the performance of the proposed system has been compared with other existing systems with respect to key generation time on both UE and MME sides. Here, the key generation time has been taken as the sum of the time required to generate the following keys: public keys (P_{U_A} and P_M), shared keys (K_S) and encryption keys (TEK_{1-81} and KS) for both UE and MME sides, for the purpose of comparison. Furthermore, the encryption and decryption time, the computational cost, the total computational time, the time complexity and the storage overhead have been calculated and compared with other existing systems. The respective results are listed below:

5.1 Key generation time

The time elapsed for the generation of different keys and different functions on UE side and MME side has been summarized in Table 6 and Table 7 respectively.

The proposed system has been compared with some related existing systems such as SPDiD (Huang et al. 2013), Se4GE (Huang et al. 2014) and Kanani et al. (2014) based

Table 4 Security comparison of the proposed scheme with other related schemes

Security attributes (SA)/reference schemes	SA1	SA2	SA3	SA4	SA5	SA6	SA7	SA8
SPDiD (Huang et al. 2013)	Partial	Yes	No	No	Yes	Yes	No	Yes
Se4GE (Huang et al. 2014)	Partial	Yes	No	No	Yes	Yes	No	Yes
Kanani et al. (2014)	Partial	Yes	No	No	Yes	Yes	No	Yes
MEPS-AKA (Abdrabou et al. 2015)	Partial	Yes	Yes	–	No	Yes	No	No
Enhanced-AKA (Degefa et al. 2016)	Full	Yes	Yes	Yes	Yes	Yes	No	No
Hamandi et al. (2017)	Partial	No	Yes	–	No	Yes	No	No
Improved EPS-AKA (Abdeljebbar and Kouch 2018)	Full	Yes	Yes	–	No	Yes	No	Yes
SEGB (Parne et al. 2018)	Full	Yes	Yes	–	Yes	Yes	No	Yes
DGBES (Gupta et al. 2018)	Full	Yes	Yes	–	Yes	Yes	No	Yes
EAKA-EPS (Singh and Shrimankar 2018)	Full	Yes	Yes	–	Yes	Yes	No	Yes
Kumari et al. (2018)	Full	Yes	Yes	Yes	No	Yes	No	Yes
LTE-A (Abdrabou et al. 2015; Alezabi et al. 2014; Degefa et al. 2016; Hamandi et al. 2017; Hou et al. 2010 ; Køien 2011; Xiehua and Yongjun 2011)	Partial	No	No	No	No	No	No	No
Proposed	Full	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Yes defends the attack, *No* unable to defend the attack, – not applicable

Table 5 Simulation parameters

Parameter	Value
Physical network	LTE-A
Number of eNBs	One
Number of UEs	One
Number of MMEs	One
Processor	Intel i5- 4590 3.30 GHz
RAM	16 GB
Operating system	Windows 10 64-bit

Table 6 Consumption of time for the generation of different keys and related functions on UE side

Key/function	Key length (bits)		
	112 (s)	128 (s)	160 (s)
V_A	0.0964	0.0982	0.1115
PU_A	0.1021	0.1048	0.1362
U_A	0.0535	0.0588	0.0717
K_S	0.1252	0.332	0.3593
IK1-3	0.0006367	0.0006448	0.0006513
IK4-6	0.0000439	0.0000482	0.0000545
IK7-9	0.0000491	0.0000529	0.0000622
DK1-9	0.0003042	0.0003123	0.0003623
DX1-9	0.0005514	0.0005661	0.0005872
TEK1-81	0.0011	0.0013	0.0016
KS	0.0151	0.0157	0.0163
HMAC()	0.223	0.2246	0.2277
ECC_Enc()	0.0968	0.1001	0.1308
ECC_Dec()	0.0951	0.0994	0.1279
Enc_Fun()	0.0000084	0.0000093	0.0000191
Dec_Fun()	0.0646	0.0767	0.0985

on key generation time on both UE and MME sides as presented in Fig. 4a, b respectively.

Figure 4a shows that the time consumed to generate the keys on UE side for the proposed system is 0.513 s as compared to the existing systems such as SPDiD, Se4GE and Kanani et al. (2014) which take 1.827 s, 3.495 s and 4.296 s respectively for a security level of 80-bit resulting in a percentage improvement of 71.92, 85.32 and 88.05% for the proposed system over SPDiD, Se4GE and Kanani et al. (2014) respectively. Similarly, Fig. 4(b) shows that the time required to generate the keys on MME side for the proposed system is 0.591 s in contrast to SPDiD, Se4GE and Kanani et al. (2014) which take 1.819 s, 3.502 s and 3.558 s respectively for a security level of 80-bit providing a percentage improvement of 67.51, 83.12 and 83.39% for the proposed system over SPDiD, Se4GE and Kanani et al. (2014) respectively. These results indicate that the proposed

Table 7 Consumption of time for the generation of different keys and related functions on MME side

Key/function	Key length (bits)		
	112 (s)	128 (s)	160 (s)
PM	0.0952	0.0993	0.1277
M_A	0.001	0.0601	0.0713
K_S	0.2069	0.3371	0.4571
IK1-3	0.0018	0.0018	0.0018
IK4-6	0.0000445	0.0000488	0.0000541
IK7-9	0.0000392	0.0000613	0.0000815
DK1-9	0.0002992	0.0003064	0.0003284
DX1-9	0.0007573	0.0007707	0.0008367
TEK1-81	0.0013	0.0013	0.0014
KS	0.0047	0.0047	0.0048
HMAC()	0.1744	0.175	0.1771
ECC_Enc()	0.0541	0.0642	0.0716
ECC_Dec()	0.1286	0.15	0.1716
Enc_Fun()	0.0058	0.006	0.0072
Dec_Fun()	0.0064	0.0066	0.0076

system is more efficient than SPDiD, Se4GE and Kanani et al. (2014) as far as key generation time is concerned.

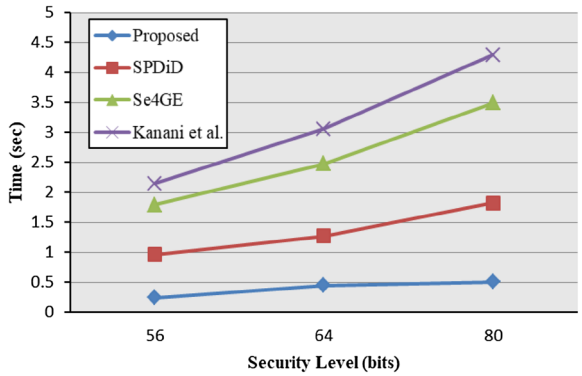
5.2 Encryption and decryption time

Comparison of the time spent for encrypting the plaintext on UE side and decrypting the cipher text on MME side of the proposed system and the other existing systems such as SPDiD, Se4GE and Kanani et al. (2014) has been presented in Fig. 5a, b respectively for different sizes of the plain text such as 256, 768, 512 and 1024 bits

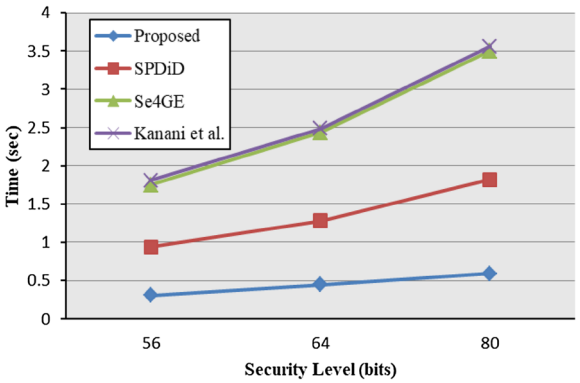
From Fig. 5a it is found that the time required for encrypting the plain text of 1024-bit in length takes 1.096 s, 2.067 s, 2.822 s and 2.92 s for the proposed system, SPDID, Se4GE and Kanani et al. (2014) system respectively. As far as decryption time is concerned, these values are found to be 2.34 s, 4.063 s, 36.909 s and 38.669 s for the proposed system, SPDID, Se4GE and Kanani et al. (2014) system respectively as evident from Fig. 5b considering a text size of 1024-bit. In both the cases, it is found that the proposed system provides considerable improvement over SPDID, Se4GE and Kanani et al. (2014) system.

5.3 Computational cost

The logic behind the calculation of the computational cost of the proposed scheme and related existing schemes has been analyzed as follows: While computing the computational cost, different simpler operations such as addition, subtraction, X-OR etc. as stated in Sect. 3 (Sect. 3.2.4) has not been included due to their minimal contribution as compared to

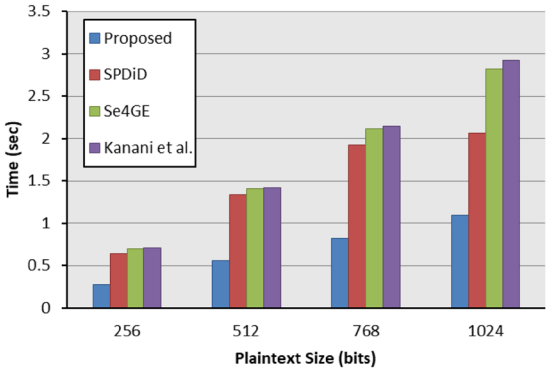


(a) The Key generation time of the proposed system and existing systems on UE side

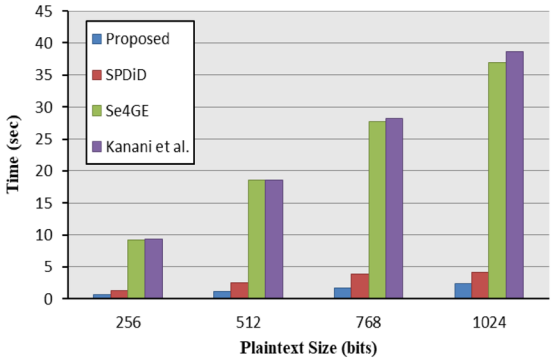


(b) The Key generation time of the proposed system and existing systems on MME side

Fig. 4 Comparison of key generation time of the proposed system with existing systems on UE side and MME side. **a** The Key generation time of the proposed system and existing systems on UE side. **b** The Key generation time of the proposed system and existing systems on MME side



(a) The encryption time of proposed system and existing systems



(b) The decryption time of proposed system and existing systems

Fig. 5 Comparison of the encryption time and decryption time of the proposed system with existing systems. **a** The encryption time of proposed system and existing systems. **b** The decryption time of proposed system and existing systems

other operations. For computing the computational cost different notations are used which are defined below:

1. T_{ME}: the time for executing a modular exponentiation operation
2. T_{EPM}: the time for computing an elliptic curve point multiplication
3. T_{RE}: the time for computing a RSA encryption operation
4. T_{RD}: the time for computing a RSA decryption operation
5. T_H: the time for executing a HMAC operation
6. T_{SY_ENC}: the time for computing symmetric encryption/decryption operation
7. T_{ASY_ENC}: the time for computing asymmetric encryption/decryption operation
8. T_{AES}: the time for computing AES encryption/decryption operation
9. T_{KDF}: the time for executing KDF operation
10. T_Q: the time for computing quantum key operation
11. T_{MAC}: the time for executing MAC operation

Here, computational cost is evaluated individually for all the phases such as Registration phase, Authentication and key exchange phase and Data transmission phase by considering the computational time of the security functions which are mentioned above. The detailed analyses have been presented below:

5.3.1 Computational cost of the SPDiD scheme (Huang et al. 2013)

Computational costs corresponding to the several operations executed on the different phases are as follows:

Registration phase Not applicable

Authentication and key exchange phase $T_{ME} + 2\,T_H + 2\,T_{ME} + 2\,T_H = 3\,T_{ME} + 4\,T_H$

Data transmission phase $T_{ME} + 8\,T_H$

The overall computational cost of the SPDiD scheme is: $4\,T_{ME} + 12\,T_H$

5.3.2 Computational cost of the Se4GE scheme (Huang et al. 2014)

Computational costs corresponding to the several operations executed on the different phases are as follows:

Registration phase Not applicable

Authentication and key exchange phase $2\,T_{ME}+T_{RE}+T_H+T_{RD}+T_H+4\,T_{ME}+T_H+2\,T_{ME}+T_H = 8\,T_{ME}+T_{RE}+T_{RD}+4\,T_H$

Data transmission phase: $4\,T_H$

The overall computational cost of the Se4GE scheme is: $8\,T_{ME}+T_{RE}+T_{RD}+8\,T_H$

5.3.3 Computational cost of the Kanani et al. (2014) scheme

Computational costs corresponding to the several operations executed on the different phases are as follows:

Registration phase Not applicable

Authentication and key exchange phase $2\,T_{ME}+2\,T_{RE}+T_H+2\,T_{RD}+T_H+3\,T_{ME}+2\,T_H+T_{ME}+T_{RE}+T_H+T_{RD}+T_H+T_{ME}+2\,T_H = 7\,T_{ME}+3\,T_{RE}+3\,T_{RD}+8\,T_H$

Data transmission phase Not applicable

The overall computational cost of the Kanani et al. (2014) scheme is: $7\,T_{ME}+3\,T_{RE}+3\,T_{RD}+8\,T_H$

5.3.4 Computational cost of the MEPS-AKA scheme (Abdrabou et al. 2015)

Computational costs corresponding to the several operations executed on the different phases are as follows:

Registration phase Not applicable

Authentication and key exchange phase $T_H + T_{ME} + T_{SY\_ENC} + T_{ME} + 7\ T_{SY\_ENC} = 2\ T_{ME} + T_H + 8\ T_{SY\_ENC}$

Data transmission phase Not applicable

The overall computational cost of the MEPS-AKA scheme is: $2\,T_{ME}+T_H+8\,T_{SY\_ENC}$

5.3.5 Computational cost of the Enhanced-AKA scheme (Degefa et al. 2016)

Computational costs corresponding to the several operations executed on the different phases are as follows:

Registration phase Not applicable

Table 8 Computational cost of the proposed system and different existing systems

Performance properties/reference schemes	Registration phase	Authentication phase	Data transmission phase	Total computational cost
SPDiD (Huang et al. 2013)	–	$3\,T_{ME}+4\,T_H$	$T_{ME}+8\,T_H$	$4\,T_{ME}+12\,T_H$
Se4GE (Huang et al. 2014)	–	$8\,T_{ME}+T_{RE}+T_{RD}+4\,T_H$	$4\,T_H$	$8\,T_{ME}+T_{RE}+T_{RD}+8\,T_H$
Kanani et al. (2014)	–	$7\,T_{ME}+3\,T_{RE}+3\,T_{RD}+8\,T_H$	–	$7\,T_{ME}+3\,T_{RE}+3\,T_{RD}+8\,T_H$
MEPS-AKA (Abdrabou et al. 2015)	–	$2\,T_{ME}+T_H+8\,T_{SY\_ENC}$	–	$2\,T_{ME}+T_H+8\,T_{SY\_ENC}$
Enhanced-AKA (Degefa et al. 2016)	–	$3\,T_{KDF}+4\,T_{SY\_ENC}$	–	$3\,T_{KDF}+4\,T_{SY\_ENC}$
Hamandi et al. (2017)	–	$T_{MAC}+3\,T_{KDF}+2\,T_{SY_ENC}+(3\,T_{KDF})$x	–	$T_{MAC}+3\,T_{KDF}+2\,T_{SY_ENC}+(3\,T_{KDF})$x
Improved EPS-AKA (Abdeljebbar and Kouch 2018)	–	$3\,T_{ME}+4\,T_H+T_{MAC}+2\,T_{KDF}+11\,T_{ASY\_ENC}$	–	$3\,T_{ME}+4\,T_H+T_{MAC}+2\,T_{KDF}+11\,T_{ASY\_ENC}$
DGBES (Gupta et al. 2018)	–	$7\,(T_H)*n+(4T_H+2T_{AES})*m$	–	$7\,(T_H)*n+(4T_H+2T_{AES})*m$
SEGB (Parne et al. 2018)	–	$(7T_H+4T_{AES})*n+(4T_H)*m$	–	$(7T_H+4T_{AES})*n+(4T_H)*m$
EAKA-EPS (Singh and Shrimankar 2018)	–	$3\,T_{MAC}+4\,T_{KDF}+6\,T_{ASY\_ENC}$	–	$3\,T_{MAC}+4\,T_{KDF}+6\,T_{ASY\_ENC}$
Kumari et al. (2018)	$6\,T_{EPM}+T_H$	$13\,T_{EPM}+7\,T_H+2\,T_Q$	–	$19\,T_{EPM}+8\,T_H+2\,T_Q$
Proposed	$2\,T_{EPM}$	$20\,T_{EPM}+4\,T_H$	$6\,T_H$	$22\,T_{EPM}+10\,T_H$

Authentication and key exchange phase $T_{SY\_ENC} + T_{KDF} + T_{SY\_ENC} + T_{KDF} + 2\ T_{SY\_ENC} + T_{KDF} = 3\ T_{KDF} + 4\ T_{SY\_ENC}$

Data transmission phase Not applicable

The overall computational cost of the MEPS-AKA scheme is: $3\ T_{KDF} + 4\ T_{SY\_ENC}$

5.3.6 Computational cost of the Hamandi et al. (2017) scheme

Computational costs corresponding to the several operations executed on the different phases are as follows:

Registration phase Not applicable

Authentication and key exchange phase $T_{SY\_ENC} + T_{MAC} + 3\ T_{KDF} + T_{SY\_ENC} + (3\ T_{KDF})^*\ x = T_{MAC} + 3\ T_{KDF} + 2\ T_{SY\_ENC} + (3\ T_{KDF})^*\ x$

Data transmission phase Not applicable

The overall computational cost of the MEPS-AKA scheme is: $T_{MAC} + 3\ T_{KDF} + 2\ T_{SY\_ENC} + (3\ T_{KDF})^*\ x$

Where, x is represented as the required numbers of authentication vectors/UEs.

5.3.7 Computational cost of the Improved EPS-AKA scheme (Abdeljebbar and Kouch 2018)

Computational costs corresponding to the several operations executed on the different phases are as follows:

Registration phase Not applicable

Authentication and key exchange phase $T_H + T_{ME} + T_{ASY\_ENC} + T_{ME} + 3\ T_{ASY\_ENC} + 2\ T_H + T_{ME} + T_{ASY\_ENC} + T_{MAC} + T_{KDF} + T_{ME} + 4\ T_{ASY\_ENC} + T_H + T_{KDF} + 2\ T_{ASY\_ENC} = 3\ T_{ME} + 4\ T_H + T_{MAC} + 2\ T_{KDF} + 11\ T_{ASY\_ENC}$

Data transmission phase Not applicable

The overall computational cost of the MEPS-AKA scheme is: $3\ T_{ME} + 4\ T_H + T_{MAC} + 2\ T_{KDF} + 11\ T_{ASY\_ENC}$

5.3.8 Computational cost of the DGBES scheme (Gupta et al. 2018)

Computational costs corresponding to the several operations executed on the different phases are as follows:

Registration phase Not applicable

Authentication and key exchange phase

(a) The computational cost of MTC devices is: $(4\ T_H)$ * n + $(2\ T_H + T_{AES})$ * m
(b) The computational cost of network is: $(3\ T_H)$ * n + $(2\ T_H + T_{AES})$ * m

Total computational cost: $(7\ T_H)$ * n + $(4\ T_H + 2\ T_{AES})$ * m

Data transmission phase: Not applicable

The overall computational cost of the MEPS-AKA scheme is: $(7\ T_H)$ * n + $(4\ T_H + 2\ T_{AES})$ * m

Where, n and m are represented as the number of MTCDS and number of group formed for n number of MTCDS respectively.

5.3.9 Computational cost of the SEGB scheme (Parne et al. 2018)

Computational costs corresponding to the several operations executed on the different phases are as follows:

Registration phase Not applicable

Authentication and key exchange phase

(a) The computational cost of MTC devices is: $(4\ T_H + 2\ T_{AES})$ * n + $(2\ T_H)$ * m
(b) The computational cost of network is: $(3\ T_H + 2\ T_{AES})$ * n + $(2\ T_H)$ * m

Total computational cost: $(7\ T_H + 4\ T_{AES})$ * n + $(4\ T_H)$ * m

Data transmission phase Not applicable

The overall computational cost of the MEPS-AKA scheme is: $(7\ T_H + 4\ T_{AES})$ * n + $(4\ T_H)$ * m

5.3.10 Computational cost of the EAKA-EPS scheme (Singh and Shrimankar 2018)

Computational costs corresponding to the several operations executed on the different phases are as follows:

Registration phase Not applicable

Authentication and key exchange phase $T_{ASY\_ENC} + T_{MAC} + 2\ T_{KDF} + 2\ T_{ASY\_ENC} + T_{MAC} + T_{ASY\_ENC} + T_{KDF} + T_{ASY\_ENC} + T_{MAC} + T_{ASY\_ENC} + T_{KDF} = 3\ T_{MAC} + 4\ T_{KDF} + 6\ T_{ASY\_ENC}$

Data transmission phase Not applicable

The overall computational cost of the MEPS-AKA scheme is: $3\ T_{MAC} + 4\ T_{KDF} + 6\ T_{ASY\_ENC}$

5.3.11 Computational cost of the Kumari et al. (2018) scheme

Computational costs corresponding to the several operations executed on the different phases are as follows:

Registration phase $2\ T_{EPM} + T_H + 4\ T_{EPM} = 6\ T_{EPM} + T_H$

Authentication and key exchange phase $12\ T_{EPM} + T_H + T_{EPM} + 4\ T_H + 2\ T_Q + 2\ T_H = 13\ T_{EPM} + 7\ T_H + 2\ T_Q$

Data transmission phase Not applicable

The overall computational cost of the MEPS-AKA scheme is: $19\ T_{EPM} + 8\ T_H + 2\ T_Q$

5.3.12 Computational cost of the proposed scheme

Computational costs corresponding to the several operations executed on the different phases are as follows:

Registration phase $T_{EPM}+T_{EPM}=2\,T_{EPM}$

Authentication and key exchange phase $6\,T_{EPM}+T_H+4\,T_{EPM}+T_H+6\,T_{EPM}+T_H+4\,T_{EPM}+T_H=20\,T_{EPM}+4\,T_H$

Data transmission phase $6\,T_H$

The overall computational cost of the proposed scheme is $22\,T_{EPM}+10\,T_H$.

The computational cost of the proposed system and other related systems has been listed in Table 8.

From Table 8, it is noticed that the proposed system achieves lower computational cost as compared to other existing systems such as SPDiD, Se4GE, Kanani et al. (2014), MEPS-AKA, Improved EPS-AKA and EAKA-EPS system. This is because of the fact that the proposed system uses elliptic curve point multiplication whereas the systems SPDiD, MEPS-AKA, Improved EPS-AKA and EAKA-EPS includes modular exponentiation operation and the systems Se4GE and Kanani et al. (2014) use RSA encryption and decryption operation and modular exponentiation operation (Chung et al. 2007). Moreover, the computational cost of the proposed scheme is little higher than some of the related existing schemes due to the fact that the proposed scheme uses more ECC functions to protect the keys which helps to prevent the system from several security attacks. Furthermore, it achieves perfect forward secrecy. Although some of the proposed schemes achieves better computational cost than the proposed scheme, many of them are vulnerable to several security attacks such as DoS attack, replay attack and many logged in user's/device's attack and also do not provide perfect forward secrecy. Thus, it can be said that the proposed protocol provides better security than the existing schemes with competitive computational cost.

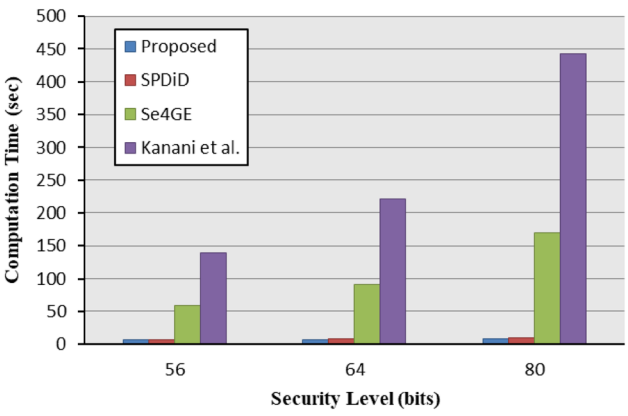


Fig. 6 Comparison of the total computational time of the proposed system and existing systems

5.4 Computational time

Comparison of total computation time of the proposed system with existing systems has been presented in Fig. 6.

From Fig. 6 it is found that the total computation time of the proposed system is only 8.724 s in contrast to SPDiD, Se4GE and Kanani et al. (2014) system which consumes a time of 10.21 s, 170.593 s and 442.895 s respectively for a security level of 80-bit, resulting in a percentage improvement of 14.55, 94.89 and 98.03% over the systems SPDiD, Se4GE and Kanani et al. (2014) respectively.

5.5 Time complexity

The time complexity of the proposed scheme and the related schemes has been evaluated based on the following logic: The security of the proposed system is based on the hard of solving ECDLP. Hence, the time complexity of the proposed system is $O(\sqrt{p})$ (Soram and Khomdram 2009; Panda and Chattopadhyay 2019), where p is the largest prime divisor of the order n. The security of the other related systems SPDiD, Se4GE and Kanani et al. (2014) are depends on the difficulty of solving discrete logarithm problem. Therefore, the time complexity of the systems SPDiD, Se4GE and Kanani et al. (2014) is $O(\exp\sqrt{cm\ln m})$ (Panda and Chattopadhyay 2019; Elgamal 1985), where, c = 0.69 and m is the length of the public key. The time complexity of the proposed system and other related systems are listed in Table 9.

Table 9 Time complexity of proposed system and different existing systems

Reference schemes	Time complexity
Proposed	$O(\sqrt{p})$
SPDiD	$O(\exp\sqrt{cm\ln m})$
Se4GE	$O(\exp\sqrt{cm\ln m})$
Kanani et al. (2014)	$O(\exp\sqrt{cm\ln m})$

Table 10 Setting of parameters

Parameters	Size (bits)
IMSI/IMSI$_A$	128
Time Stamp	64
Random number/PRN	128
HMAC	64
DH/ECDH	192
Authentication Key	128

Table 11 Comparison of storage overhead of the proposed system with existing systems

Reference schemes	Storage overhead (bits)
Proposed scheme	705
SPDiD	1536
Se4GE	3352
Kanani et al. (2014)	3352

5.6 Storage overhead

In this section, we present the storage overhead of the proposed scheme and some existing schemes. The list of parameters with the standard size for the evaluation of storage overhead has been listed in Table 10 (Saxena et al. 2015). The logical key reasoning for evaluating the storage overhead of the proposed and related existing schemes has been analyzed as follows:

In the proposed scheme, at the initial stage of network entry when the user gets registered with the server, user's identity ($IMSI_A$, $IMSI_B$ ….), password verifier (V_A) and a status bit into a write protected file also get stored in the server. Subsequently, the server sends its public key P_S to the users. Moreover, by the request of MME for the authentication purpose, the server sends K_P and V_A to MME. Hence, the storage space requirement for the proposed scheme is found to be 705 bits which has been calculated by adding the individual storage space of all the above mentioned parameters. In SPDiD, the server stores the DCC (K_i and K_f) and IMSI and also sends a message $PRN_{1-6}|CSK|P'_{hs}|HMAC(P_{hs} + PRN_5 \oplus PRN_6)$ to MME for the purpose of authentication. Therefore, the storage space for SPDiD is calculated as 1536 bits. Similarly, in both the schemes such as Se4GE and Kanani et al. (2014), the server stores the DCC which contains IMSI, RSA triple keys such as public key e_i, private key d_i and the modulus N_i and authentication key K_i. Thus the storage space requirement for both the schemes Se4GE and Kanani et al. (2014) is found to be 3352 bits for 1024 bits RSA system (Soram and Khomdram 2009; Panda and Chattopadhyay 2019). The storage overhead of the proposed system and existing systems has been presented in Table 11.

5.7 Discussion

The outcomes of the above analysis have been summarized below:

1. The proposed system attains better percentage of improvement over other existing systems SPDiD, Se4GE and Kanani et al. (2014) with respect to key generation time.
2. The proposed system modifies the Salsa20 stream cipher technique at the time of the process of encryption and decryption and uses it for the same purpose. Hence in contrast to other existing systems SPDiD, Se4GE and Kanani et al. (2014), the proposed system acquires faster encryption of plain text and decryption of cipher text.
3. The proposed system offers proper mutual authentication where some other related existing systems SPDiD, Se4GE, Kanani et al. (2014), MEPS-AKA and Hamandi et al. (2017) provide partial mutual authentication.
4. The proposed system attains greater security than the standard LTE and the related existing systems.
5. The computational cost of the proposed system decreases from the values of the other related systems SPDiD, Se4GE, Kanani et al. (2014), MEPS-AKA, Improved EPS-AKA and EAKA-EPS. This is due to the fact that the proposed system includes ECC and ECDH in contrast to the systems SPDiD, Se4GE, Kanani et al. (2014), MEPS-AKA, Improved EPS-AKA and EAKA-EPS which employ RSA and DH-PKDS in it. This achievement has occurred because of the fact that the computational cost of elliptic curve point multiplication is much less than that of modular exponentiation used in RSA and DH-PKDS (Chung et al. 2007). Moreover, the proposed system achieves better percentage of improvement on total computation time over the existing systems SPDiD, Se4GE and Kanani et al. (2014).
6. The performance analysis shows that the storage overhead of the proposed system is also reduced as compared to the existing systems SPDiD, Se4GE and Kanani et al. (2014).
7. Hence it can be concluded that the proposed scheme outperforms the related existing systems in all respect.

6 Conclusions and future work

In this paper, an improved authentication and security scheme has been proposed for LTE/LTE-A networks by employing ECC, ECDH and Salsa20 stream cipher algorithm to enhance the end to end security and speedy data transmission. The proposed work protects the transmission messages, prevents the system from several security attacks and offers proper mutual authentication by incorporating a number of propositions such as timestamp, different encrypted functions, authentication parameters, HMAC and user password verifier. From the security analysis of the system it is found that the proposed system attains better security as compared to LTE standard and some related existing work. Furthermore, the performance analysis of the proposed system shows the following outcomes: Firstly, the

key generation time of the proposed scheme is much less than other related systems SPDiD, Se4GE and Kanani et al. (2014). Secondly, the encryption and decryption speed are faster than the systems SPDiD, Se4GE and Kanani et al. (2014). Thirdly, the computational cost and the total computation time of the proposed system are much lower as compared to other existing systems. Finally, the storage overhead of the proposed system is also significantly decreased. Hence it can be concluded that our proposed system is more efficient, secure, and reliable as compared to the existing security schemes.

The above discussion shows that our proposition is capable to provide lower computation cost. However, enhancing the performance of the system without sacrificing its security by employing a reduced number of ECC point multiplication can be considered an important area of research in future. Extension of this work to emerging technology like IoT can also be considered as another scope for future work.

References

Abdeljebbar M, Kouch RE (2018) Security improvements of EPS-AKA Protocol. Inter J Netw Secur 20(4):636–644

Abdrabou MA, Elbayoumy ADE, Wanis EAE (2015) LTE authentication protocol (EPS-AKA) weaknesses solution. In: Proceedings of the Seventh IEEE international conference on intelligent computing and information systems, pp 434–441

Afdhila D, Nasution SM, Azmi F (2016) Implementation of stream cipher salsa20 algorithm to secure voice on push to talk application. In: Proceedings of the IEEE Asia pacific conference on wireless and mobile, pp 137–141

Akyildiz IF, David M, Estevez G, Reyes EC (2010) The evolution to 4G cellular systems: LTE-advanced. Phys Commun 3:217–244

Alezabi KA, Hashim F, Hashim SJ, Ali BM (2014) An efficient authentication and key agreement protocol for 4G (LTE) networks. In: Proceedings of the IEEE region 10 symposium, pp 502–507

Barker E (2016) Recommendation for key management part 1: General (revision 4). NIST Spec Publ 800(57):1–147

Bernstein DJ (2005) Snuffle 2005: the Salsa20 encryption function. http://cr.yp.to/snuffle.html#xsalsa

Bernstein DJ (2008) The Salsa20 family of stream ciphers. In: LNCS 4986, Springer, pp 84–97

Cao J, Li H, Ma M, Zhang Y, Lai C (2012) A simple and robust handover authentication between henb and enb in LTE networks. Comput Netw 56(8):2119–2131

Cao J, Ma M, Li H, Zhang Y, Luo Z (2014) A survey on security aspects for LTE and LTE-A networks. IEEE Commun Surv Tutorial 16:283–302

Cheng C, Lu R, Petzoldt A, Takagi T (2017) Securing the internet of things in a quantum world. IEEE Commun Mag 55:116–120

Chien H-Y (2018) An effective approach to solving large communication overhead issue and strengthening the securities of AKA protocols. Int J Commun Syst 31:e3381

Chung YF, Huang KH, Lai F, Chen TS (2007) ID-based digital signature scheme on the elliptic curve cryptosystem. Comput Stand Interfaces 29(6):601–604

Degefa FB, Lee D, Kim J, Choi Y, Won D (2016) Performance and security enhanced authentication and key agreement protocol for SAE/LTE network. Comput Netw 94(1):145–163

Elgamal T (1985) A public key cryptosystem and signature scheme based on discrete logarithms. IEEE Trans Inf Theory 31:469–472

Ferrag MA, Maglaras L, Argyriou A, Kosmanos D, Janicke H (2018) Security for 4G and 5G cellular networks: A survey of existing authentication and privacy-preserving schemes. J Netw Comput Appl 101:55–82

Gupta S, Parne BL, Chaudhari NS (2018) DGBES: Dynamic Group Based Efficient and Secure authentication and key agreement protocol for MTC in LTE/LTE-A networks. Wirel Pers Commun 98:2867–2899

Hamandi K, Abdo JB, Elhajj IH, Kayssi A, Chehab A (2017) A privacy-enhanced computationally-efficient and comprehensive LTE-AKA. Comput Commun 98:20–30

Hankerson D, Menezes A, Vanstone S (2004) Guide to elliptic curve cryptography. Springer-Verlag, New York

Hou M, Xu Q, Shanqing G, Jiang H (2010) Cryptanalysis of identity-based authenticated key agreement protocols from parings. J Netw 5(7):826–855

Huang YL, Leu FY, Wei KC (2012) A secure communication over wireless environments by using a data connection core. Math Comput Model 58(5):1459–1474

Huang YL, Leu FY, Liu JC, Lo LJ, Chu WCC (2013) A secure wireless communication system integrating PRNG and Diffie-Hellman PKDS by using a Data Connection Core. In: Proceedings of the IEEE International conference on broadband wireless computing, communication and applications, pp 360- 365

Huang YL, Leu FY, You I, Sun YK, Chu CC (2014) A secure wireless communication system integrating RSA, Diffie-Hellman PKDS, intelligent protection-key chains and a data connection core in a 4G environment. J Supercomput 67(3):635–652

Hwang JJ, Yeh TC (2002) Improvement on Peyravian–Zunic's password authentication schemes. IEICE Trans Commun E85-B(4):823–825

Islam SKH, Biswas GP (2013) Design of improved password authentication and update scheme based on elliptic curve cryptography. Math Comput Model 57(11–12):2703–2717

Jablon D (2013) The SPEKE Password-based key agreement methods. IETF draftjablon-speke-02.txt

Kanani PI, Kaue V, Shah K (2014) Hybrid PKDS in 4G using secured DCC. In: Proceedings of the IEEE international conference on signal propagation and computer technology, pp 1-6

Køien GM (2011) Mutual entity authentication for LTE. In: Proceedings of the 7th IEEE international conference on wireless communications and mobile computing conference, pp 689–694

Kumari KA, Sadasivam GS, Gowri SS, Akash SA, Radhika EG (2018) An approach for End-to-End (E2E) security of 5G applications. In: Proceedings of the 4th IEEE international conference on big data security on cloud, pp 133–138

Lai C, Li H, Lu R, Shen XS (2013) SE-AKA: A secure and efficient group authentication and key agreement protocol for LTE networks. Comput Netw 57(17):3492–3510

Li C-T (2013) A new password authentication and user anonymity scheme based on elliptic curve cryptography and smart card. IET Inf Secur 7(1):3–10

Lin CL, Hwang T (2003) A password authentication scheme with secure password updating. Comput Secur 22(1):68–72

Mahto D, Khan DA, Yadav DK (2016) Security analysis of elliptic curve cryptography and RSA. In: The world congress on engineering, pp 1-4

Musaddiq A, Zikria YB, Hahm O, Yu H, Bashir AK, Kim SW (2018) A survey on resource management in IoT operating systems. IEEE Access 6:8459–8482

Ni J, Lin X, Shen XS (2018) Efficient and secure service-oriented authentication supporting network slicing for 5G-enabled IoT. IEEE J Sel Areas Commun 36(3):644–657

Panda PK, Chattopadhyay S (2019) A modified PKM environment for the security enhancement of IEEE 802.16e. Comput Stand Interfaces 61:107–120

Parne BL, Gupta S, Chaudhari NS (2018) SEGB: security enhanced group based AKA protocol for M2 M communication in an IoT enabled LTE/LTE-A network. IEEE Access 6:3668–3684

Peyravian M, Zunic N (2000) Methods for protecting password transmission. Comput Secur 19(5):466–469

Saxena N, Thomas J, Chaudhari NS (2015) ES-AKA: an efficient and secure authentication and key agreement protocol for UMTS networks. Wirel Pers Commun 84:1981–2012

Singh G, Shrimankar D (2018) A privacy preserving authentication protocol with secure handovers for the LTE/LTE-A networks. Sadhana 43:128. https://doi.org/10.1007/s12046-018-0891-1

Soram R, Khomdram M (2009) Juxtaposition of RSA and elliptic curve cryptosystem. Int J Comput Sci Netw Secur 9(9):11–21

Xiehua L, Yongjun W (2011) Security enhanced authentication and key agreement protocol for LTE/SAE network. In: Proceedings of the 7th IEEE international conference on wireless communications, networking and mobile computing (WiCOM), pp 1–4

Xu L, Wu F (2015) An improved and provable remote user authentication scheme based on elliptic curve cryptosystem with user anonymity. Secure Commun Netw 8(2):245–260

Xu D, Chen J, Liu Q (2018) Provably secure anonymous three-factor authentication scheme for multi-server environments. J Ambient Intell Hum Comput. https://doi.org/10.1007/s12652-018-0710-x

Zhu L, Yu S, Zhang X (2008) Improvement up on mutual password authentication scheme. In: Proceedings of the IEEE international seminar on business and information management, pp 400–403

Zikira YB, Yu H, Afzal MK, Rehmani MH, Hahm O (2018a) Internet of things (IoT): Operating system, applications and protocols design and validation techniques. Future Gener Comput Syst 88:699–706

Zikira YB, Afzal MK, Ishmanov F, Kim SW, Yu H (2018b) A survey on routing protocols supported by the contiki Internet of Things operating system. Future Gener Comput Syst 82:200–219