designs

Article

ML-AKA: An Authentication Protocol for Non-Standalone 5G-Based C-IoT Networks

Byomakesh Mahapatra, Vikash Singh *, Rituraj Bhattacharjee * and C. R. Srinivasan

Department of Instrumentation and Control Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Udupi 576104, India; byomakesh.mahapatra@manipal.edu (B.M.); cr.srinivasan@manipal.edu (C.R.S.)

* Correspondence: vikash.nepal@manipal.edu (V.S.); rituraj.b@manipal.edu (R.B.)

Abstract: When it comes to the development of 4G and 5G technologies, long-range IoT or machine-to-machine (M2M) communication can be achieved with the help of cellular infrastructure. In non-standalone (NSA) 5G infrastructure, cellular-IoT (C-IoT) devices are attached and authenticated by a 4G core network even if it is connected to a 5G base station. In an NSA-based 5G network, the presence of dual connectivity sometimes raises interoperability and authentication issues due to technological differences between LTE and 5G. An attacker explores these technological differences, introduces the threats, and performs various types of attacks like session hijacking at the interfaces and Man-in-the-Middle (MITM) attacks. With the introduction of these attacks, the attackers exploit the network resources and pinch out various critical information sources. To resolve this issue, the NSA-based C-IoT network must incorporate robust and seamless authentication and authorization mechanisms. This article presents the ML-AKA protocol that is used to enhance interoperability and trust between 4G and 5G networks by using a uniform key-sharing (UKS) mechanism. The proposed ML-AKA protocol is analyzed with the help of the AVISPA tool and validated with the use of Proverif. Further, the proposed protocol is compared with other existing protocols like EPS-AKA and UAKA-D2D, and the outcome shows that the proposed protocol significantly reduces the chances of MITM, DDOS and Spoofing attacks during the interoperability in the NSA-C-IoT network.

Keywords: cellular IoT; LTE; NSA-5G; ML-AKA

Citation: Mahapatra, B.; Singh, V.; Bhattacharjee, R.; Srinivasan, C.R. ML-AKA: An Authentication Protocol for Non-Standalone 5G-Based C-IoT Networks. *Designs* **2024**, *8*, 128. https://doi.org/10.3390/designs8060128

Academic Editor: Jun Feng

Received: 9 October 2024
Revised: 16 November 2024
Accepted: 21 November 2024
Published: 3 December 2024

1. Introduction

The Internet of Things (IoT) integrates sensors, processors, controllers, and network elements through a dedicated internet connection. With the development of smart infrastructure, IoT devices have become an intrinsic part of many things, such as smart agriculture, intelligent transportation, smart cities, etc. [1]. IoT platforms need to manage and process heterogeneous types of data generated by various interconnected devices attached to them. These attached devices need to implement distinct security protocols for end-to-end data delivery. The C-IoT concept evolves with the development of 4G and 5G cellular networks. Further, integration of new radio (NR) with the existing LTE infrastructures in non-standalone (NSA) 5G network architecture improves the connectivity and capacity of a network. As the NSA uses both 4G and 5G cellular infrastructure, it needs a much more flexible communication and security protocol for operation and management. In NSA-5G architecture, both the gNodeB and eNodeB are controlled by a common evolved packet core (EPC). This common EPC helps to provide seamless communication between 4G and 5G C-IoT devices. Assignment of both IoT and non-IoT devices (i.e., mobile and cellular, etc.) to the next-generation network needs more complex traffic and data privacy management strategies at the core network. In the C-IoT architecture, the information travels through different entities of the network, so it requires multi-layer authentication and a device verification protocol. As per the most recent CISCO report, there are 15 billion IoT devices connected via the cellular platform, and this is predicted to increase to

50 billion by 2025 [2]. NSA-based C-IoT architecture has multiple purposes that can be easily switched to the next-generation platform from the current LTE network with minimal changes at the base station platforms. Furthermore, NSA-5G can help to reduce the capital cost (CAPEX) and base station capacity. Thus, we can say that NSA-5G-based C-IoT is currently the fastest-developing wireless communication technology that can be used as a backbone for the IoT network. However, the integration of multiple technologies leads to many challenges in terms of the operator and service provider perspectives. Below, we have highlighted some of the network challenges and their probable solutions to support interoperability and dual connectivity in the NSA-based 5G C-IoT network.

- *Issue:* Associations between 5G and 4G lead to increases in network complexity and operational complexity.
- *Solution:* In the proposed NSA-based C-IoT architecture, entities in the LTE core network handle all the control and management functionalities. This will help to smooth the transition of C-IoT devices from 4G to 5G networks.
- *Issue:* The key generation and key management strategies of 4G and 5G are slightly different.
- *Solution:* The proposed architecture uses a multi-layer authentication mechanism to reduce computational complexity within the network elements and user devices.
- *Issue:* An increase in the heterogeneity among IoT devices requires a variety of security keys and key management strategies.
- *Solution:* In the proposed work, uniform key sharing (UKS) uses some encrypted keys that can be used at various levels of the C-IoT network.

To partially overcome the above challenges, some of the standard protocols such as UAKA-D2D and EPS-AKA have been employed in traditional LTE networks. But all these standard protocols provide limited data security and integrity [3], so to provide end-to-end security for 5G-NSA-based C-IoT architecture, we have introduced a novel multi-level security protocol.

To highlight the applicability of the proposed protocol, we have considered three end-to-end C-IoT communication scenarios. Communications scenarios involve a personal area network (PAN), a global area network (GAN) and a wide area network (WAN). In addition, the proposed protocol is validated through simulation and mathematical analysis. To overcome the multi-authentication problem and reduce the chances of spoofing attacks, we have developed a multilayer authentication and key agreement protocol for the next-generation key cellular network. The paper's major contributions are highlighted below.

1. We have propose an end-to-end NSA-based C-IoT architecture to identify and demonstrate the possible security holes.
2. We have developed a common key-based ML-AKA security protocol to establish secure communication between devices present across a multi-layer architecture.
3. We have proposed mathematical models for the generation, authentication, and verification of keys in a multi-tier architecture.

The other sections are arranged as follows: Section 2 describes previous work in this field that is relevant to C-IoT security. Section 3 describes the detailed architecture of an LTE-based C-IoT network. Section 4 highlights different security issues in C-IoT networks. Section 5 describes different previously available security protocols for C-IoT networks. In Section 6, the proposed authentication protocol and its key management strategies are presented. The security assessment, simulation result, and protocol performance analysis are listed in Section 7, and the concluding remarks are provided in Section 8.

2. Related Work

Previous researchers have carried out conceptual analyses of IoT security with regard to preventing different security attacks and threats, as well as authentication and privacy preservation. In this work, we focus on the aspects that have received less attention in previous work. To allow a 5G C-IoT system to provide privacy protection and increase the system's attack resistance, more secure and robust protocols have been developed. Furthermore, some of the authentication protocols developed for LTE technology are listed in Table 1. In [4], the authors proposed an authentication for a protocol for secure LTE-based device-to-device (D2D) and IoT communication. In this paper, they explained a Diffie–Hellman Key Exchange (DHKE) for analysis of session key agreement in D2D communication. They also described a Hash-based Message Authentication Code (HMAC) for authentication in D2D. However, these protocols involve more application-based algorithms and are less frequently used for data-sharing applications, as they have the least capability to overcome man-in-the-middle attacks (MITMs).

In [5], the authors proposed an authentication system based on a mutual commitment scheme for cellular devices. This proposed protocol cannot be used in many situations as the evaluation process and comparison are performed only during the last phase. This makes the protocol impractical in many situations.

The authors of [6] have proposed a key distribution algorithm for D2D communications in an LTE network. Their work uses the LTE framework for D2D communication to reduce communication costs. This algorithm considers the LTE core network as the key functional unit for the generation and distribution of the different session keys for IoT communication. They use an XOR-ed key for authentication purposes. The core network performs a key generation process by XOR operation and uses this XOR-ed key for device identification. In [7], a similar algorithm is proposed to establish a D2D communication key management framework. The authors of [8] have presented a solution for the privacy of user identity. They have proposed a security solution for LTE architecture using pseudonym tags, which helps to hide the actual identity of the device or users. This method provides more secure and efficient identity management compared to other techniques.

In [9], the authors create a Simple Password Exponential Key Exchange (SPEKE) protocol for the fourth-generation cellular network. In this protocol, the session keys are shared between the devices through EPC. During this transmission, there is a greater chance of session key hijacking, which could compromise the session IDs. Although all the discussed protocols can be used more efficiently for D2D and IoT communication, no one has proposed a multilevel security protocol that can incorporate multiple steps in order to secure different IoT layers.

Table 1. Overview of the contributions and limitations of security protocols for C-IoT and D2D applications.

Authors and Years	Method and Approach	Contribution (s)	Limitation(s)
Alam et al. [4] (2015)	DHKE method for LTE network	Secure from potential DDOS attack	Prone to MIMT attack
Jover et al. [8] (2015)	Multilevel security protocol for device-to-device communication	Secures IoT devices from attacks that occur at the network layer	Developed protocol provides protection against physical layer attacks
Yao et al. [10] (2016)	Group-based secure (GBS) secure key verification and validation method	User blog for each group uses a designated encoded header with each data packet.	Group header enhances required bandwidth and processing time
Raghothaman et al. [6] (2016)	Distributed key exchange (DKE) technique for IoT application	Reduces data injection attacks at the end user IoT devices	Key distribution is complex
Wang et al. [5] (2017)	Universal key exchange mechanism for cellular network	Secures the packet transmission over transmission channels	Increases communication overhead due to use of extra headers
Sun et al. [11] (2019)	Elliptical curve method for 5G cellular network	lightweight public key-based cryptosystem	Implementation complexity is high
Yan et al. [12] (2023)	EGHA protocol for cellular network	A temporary ID-based authentication protocols for 5G-based V2X.	Increases computational complexity with the added TID
Li et al. [13] (2023)	AGMA Code to reduce handover signalling in 5G-V2X	Inter- and intra-AMF handover secure solution	Other core elements are not considered
Ranaweera et al. [14] (2024)	RSA and ECC encryption method for migrated data	Service Migration Security Framework (SMSF) for 5G network	Not useful for heterogeneous network

3. Interfaces and Communication Links for NSA-Based C-IoT Networks

Providing efficient D2D or IoT communication over a cellular network platform requires various network entities like user equipment (UE), evolved NodeB (eNodeB), and core networks to be interconnected through various interfaces, either through wires or via a wireless medium. Figure 1 shows a block diagram of these entities and their interconnections [15,16]. Here, the UEs are interconnected via a Wireless LAN (WLAN) with the help of an access point (AP). These APs are interconnected to each other through a base station control by the Evolved Packet Core (EPC) network. The information between UEs and APs is exchanged through a dedicated UPlink/Downlink RF connection, whereas the interconnections between eNodeB and are created with the help of X2 interfaces. The core network performs all the policy, mobility, and authentication-related control and management functions with the use of network entities like Mobility Management Entity (MME) to provide a service gateway (S-GW), packet gateway (P-GW) and Home Subscriber Server (HSS), etc., as shown in Figure 1 [17,18].

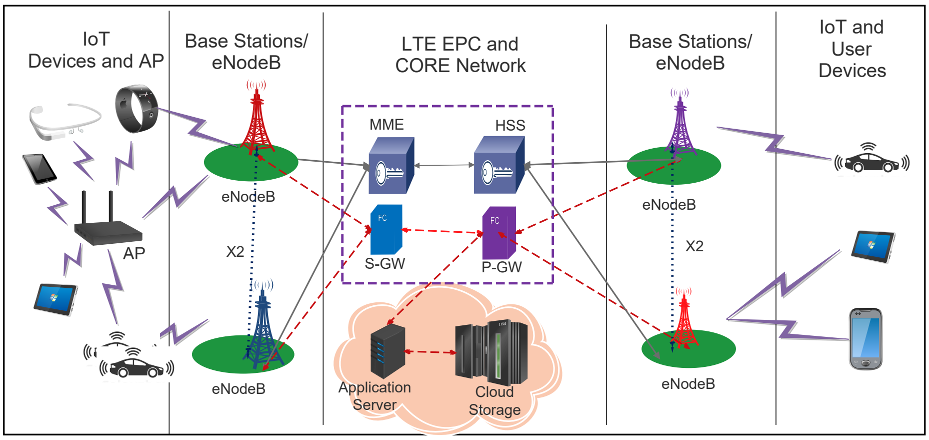


Figure 1. End-to-end device communication over a NSA-5G-based C-IoT network.

The NSA architecture consists of four different layers. Each layer has some key components and functionality for the end-to-end link establishment. The key network elements of different layers are as follows:

3.1. Device and Gateway

In the NSA C-IoT framework, a LTE infrastructure and EPC network are usually used for end-to-end data transfer and voice communication. The UE/DE is initially connected to a local AP or gateway through WiFi, Bluetooth, or any other short-range network like RFID, LoWPAN, etc. For long-range communication, these devices should be connected through a highly reliable cellular network. The access points (APs) are connected directly to the eNodeB of the LTE and LTE core network.

3.2. Access Network

In the C-IoT framework used for D2D communication between devices and the serving network i.e., the local area network connection is generally implemented over a wireless network platform. The network selection and authentication are based on the application requirement regarding the selection of the network type, e.g., local or global. The significance of the use of LTE is to allow latency, high bandwidth, and high-throughput communication to the end devices.

3.3. LTE Core and EPC Network

The LTE core parts accommodate the Universal Mobile Telecommunications System (UMTS) radio access by E-UTRAN. The LTE Core is also accompanied by the system architecture evolution (SAE), which includes the Evolved Packet Core (EPC) network in the LTE Core. By combining the concept of LTE and SAE, we create an entirely new system which is referred to as the Evolved Packet System (EPS). UEs that are present across the network are shared through these EPS.

3.4. Application and Connectivity Platform

Recent developments in cellular-based IoT communication technologies such as NB-IoT and LTE-M significantly reduce the bandwidth utilization of a cellular network and allow multiple nodes to share an allocated communication channel. However, these shared channels have some unique requirements when it comes to handling any type of diagnosis, such as SIM pre-provisioning, activation, and deactivation. To reduce their operational costs, there is a need for huge provisioning-driven mobile operators. The connectivity platform also contains a communication server which can store and forward messages, as well as a protocol translation. In essence, the connectivity platform gathers data from the cellular or IoT devices, processes them, and provides them for specific applications at the upper layer.

4. Important Security Issue in NSA-Based C-IoT Architecture

In the C-IoT architecture, a large number of devices are connected to a base station. Providing secure communication over a shared channel is a challenging task for network service providers, specifically in systems that include a huge number of devices. Because of the huge number of large-scale systems, the C-IoT network needs a real-time and secure communication platform in order to entertain all the connected devices with high throughput and data rates. After ensuring the security and privacy issues, the device authentication process is required to increase the quality of service (QoS) parameter [19]. If any unauthorized user gained access to the identity of any objects or devices, then the attacker could execute both impersonation attacks and man-in-the-middle attacks. Further, as the NSA architecture uses both the 4G and 5G architecture, there is a possibility that the object's identity or keys could be revealed over the LTE core network.

The heterogeneous traffic generated from the various types of networks must preserve the security and privacy of the data during the communication process. In the current scenario, the IoT services utilize the existing infrastructure and they use traditional encryption and the existing encryption processes to provide secrecy, which offers insufficient resistance against all the possible attacks over the different communication networks. In this research, we propose a security protocol that consists of cryptographic modules for confidentiality, integrity, authentication, and authorization for secure D2D communication [20]. As communication devices and IoT devices possess computational resources, they require lightweight authentication and authorization techniques to enable automated D2D communication by preserving traffic privacy and information security. Hence, the key generation and automated user authentication of UE/DE will be substantially more reliable [21].

The other important issue in a cellular-based IoT communication system is fake identification. Factors that cause fake identification in an IoT network include the following:

- The use of distinct unique IDs for different IoT devices or objects by user equipment can lead to confusion for mobile equipment and other IoT devices in terms of identification and authentication.
- The lack of availability of efficient algorithms for the IoT objects or device identification leads to security failures in the IoT system.

5. D2D Authentication Process for Cellular IoT Network

In the studied cellular IoT network, the authentication methods authenticate the legitimacy of any subscriber through the mutual authentication process, which allows communicating parties to conform to the required standards. The LTE system architecture uses IP-based mobility control technology that includes the access system and the core servers. The access system is E-UTRAN and the core network is EPC [22,23]. When it comes to access network security, various factors that are taken into consideration are listed below:

- UE/DE and the core network should be mutually authenticated.
- UE, MME, abd eNodeB should preserve ciphering, integrity and replay protection.
- To prevent identity theft, the access network should use a temporary identity instead of the permanent identity of the user.

The above-mentioned factors are crucial for secure D2D communication in a C-IoT network. The first step in the key agreement protocol is to generate a certain key and perform a mutual authentication process through this key. Different standard authentication and key generation protocols used for LTE networks are re-evaluated in the following subsection [24,25].

5.1. Key Generation and Authentication Process in the LTE Framework

In an LTE network, the key generation process, authentication, ciphering, and integrity flow process in a core EPS network are shown in Figure 2. The network consists of one root key (K) and several intermediate keys (CK, IK), which are derived from root key K. The third leaf keys are generated by the intermediate key [25,26]. The different functionalities of the used keys and their authentication level in the authentication and key agreement (AKS) protocol for LTE-based radio networks are listed below:

- K includes a random number of bits and is used for certain users master keys, which are stored in USIM and AuC.
- The intermediate key (CK, IK) is a 128-bit key which is derived from the root key, K.
- The access security management entity key (K_{ASME}) is generated from the intermediate key (CK, IK) with the use of two additional parameters: the serving network ID and bitwise sum of two additional parameters. The K_{ASME} is distributed as the local master key.
- The key for evolved node B (K_{eNB}) is derived from K_{ASME} and the additional input counter. This additional input is required to ensure that each new key K_{eNB} differs from the previous key.
- Keys K_{RRCenc}, K_{RRCint} and K_{UPenc} are used to authenticate the integrity of RRC and subscribers. The key generation attains the key separation and prevents associated key attacks. Once any key is changed, only the dependent keys are unaffected.

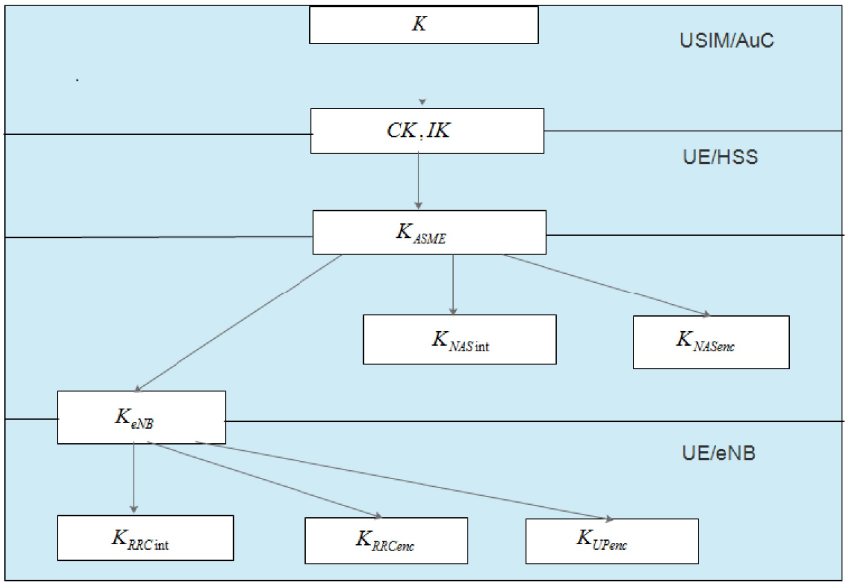


Figure 2. Key generation and authentication procedure in LTE network.

Figure 2 describes the key generation, confidentiality, integrity, ciphering and authentication process in an LTE framework for RRC communication, non-access stratum communication, and user equipment authentication. The session key generated from the root key is K, and this is used to store the address of the user service identity module (USIM) and home subscriber server (HSS), which produces a cipher key (CK), a integrity (IK), and a new session key (K_{ASME}). This K_{ASME} then acquires another cipher and IK for every communication channel [18,27].

5.2. Authentication Vector Generation Process in the LTE Framework

For the generation of AV, the serving network requests access to the home network. The valid request sent by the serving network contains the IMSI, the serving network id (SNID). This SNID is used for the calculation of K_{ASME} in the home network. By obtaining the AV request, the home network calculates it or retrieve the pre-calculated data.

In each AV, the AMF is involved in the AUTN. Computation of the AuC occurs as follows, when AuC accepts the request from home subscriber.

- Message Auth. Code = fun1$(K\|sqn\|rand\|AMF)$.
- Calculated reply XRES = fun2$(K\|rand)$.
- Cipher Key, CK = fun3$(K\|rand)$.
- Nobility Key, NK = fun4$(K\|rand)$.
- Obscurity Key, OK = fun5$(K\|rand)$.
- Auth. Token,AUTN =(sqn XOR AK)$\|AMF\|mac$.

Here, fun1, fun2, fun3, fun4, and fun5 are the cryptographic functions; fun1 is known as the message authentication function, fun2 is known as the truncated message authentication function, and fun3, fun4 and fun5 are known as key generation functions. For explanation of acronyms and symbols explanation refer Table 2.

Table 2. Acronyms and symbols.

Acronyms and Symbols	
Acronyms	**Explanation**
D_i/DID_i	Device i and its identity
$AP_i/APID_i$	AP number and its unique ID
N_i	Secret key shared between D_i and eNodeB
KDF	Cryptographic function (Key Derivation Function)
H_i	Session key generated with the help of random nonce values a and b
$HMAC_i$	Hash function based Message Authentication Code
eNodeB	Evolved node B
SQN	Sequence number
SNID	Serving network identity
AK/XAK	Anonymity key
CK/XCK	Cipher key
IK/XIK	Integrity key
SK/K	Secret key shared between devices (D_i) and eNodeBs
KSI_{ASME}	Key set identifier for each K_{ASME}
ACK	Acknowledgement
N_{asme}^{i}	Roaming key
${N_{D2D}}^{i}$	D2D fun key
r_i	Random nonce value
$rand_i$	Random value chosen by eNodeB
R_k	Shared random secret key
S_{ID}	Session identity
K_{CM}	Common secret key
K_{DS}	D2D session key
$eKSI$	Evolved key set identifier

5.3. *The Standard EPS-AKA Authentication and Key Agreement Protocol*

Various types of authentication protocols are used for IoT networks, but the most common security framework considered for a C-IoT is the evolve packet system authentication and key agreement (EPS-AKA) protocol. The EPS-AKA is based on the cryptography challenge–response protocol, which is similar to the UMTS-AKA protocol, except that it involves the generation of a set identifier (eKSI) and requires a separate indicator process for user equipment (UE) validation [28,29]. Different secret keys are generated in the LTE framework for RRC communication, non-access stratum communication, and user equipment. The mechanism of the EPS-AKA protocol is described in Figure 3, and the standard steps for this protocol are as follows:

- The UE sends service requests to the core network.
- In order to generate the EPS authentication vectors, an MME request is sent to HSS and dispensed back to MME.
- MME and HSS mutually authenticate each other and share keys between them.
- The UE dispenses secret data to the serving networks by combining the session key with the integrity key.

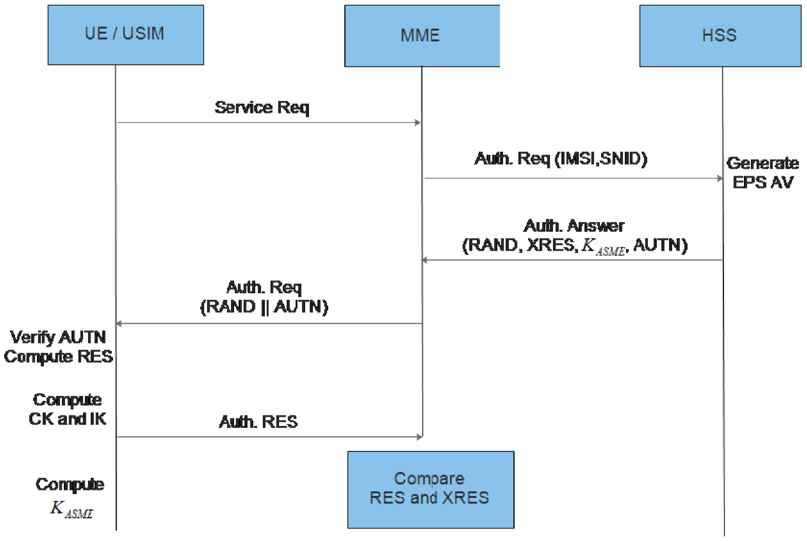


Figure 3. EPS-AKA protocol of the NSA-based C-IoT network.

6. The Novel ML-AKA Protocol for the 5G-NSA Based C-IoT Network

The following proposed protocol shows the challenge–response key agreement to provide authentic communication and security for the different UE/DE and core network devices present in the LTE-based C-IoT network. In the existing EPS-AKA algorithm, there are some potential vulnerabilities, i.e., DDoS attacks, identity catcher attacks, etc. In the ML-AKA protocol, we tackle these attacks by preserving key parameters like bandwidth and throughput. In the C-IoT network, three levels of authentication are required for secure communication from D2D or from the device to the core network. The different layers of communication are listed below:

- *Personal Area Authentication (PAA):* Authentication of the local devices for end-to-end secure communication establishment, dealing with local communication and personal devices, as shown in Figure 4.
- *Local Area Authentication (LAA):* The link between the devices and eNodeB is secured by using LAA. This deals with the authentication and secure data transfer between end-to-end devices through an eNodeB, as shown in Figure 5.
- *Global Area Authentication (GAA):* The communication between eNodeB and the core network, along with the local component, is secured by GAA. The GAA includes EPC security, physical channel security, etc., as shown in Figure 6.

The above three layers required mutual authentication and a key agreement protocol for each DE/UE at its service layer.

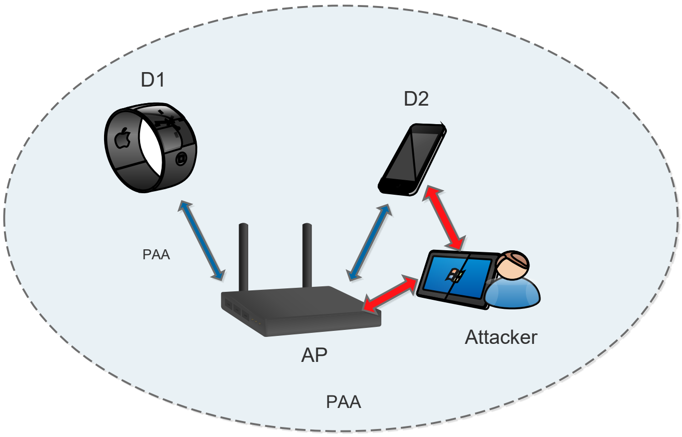


Figure 4. A C-IoT communication framework in a personal area network.

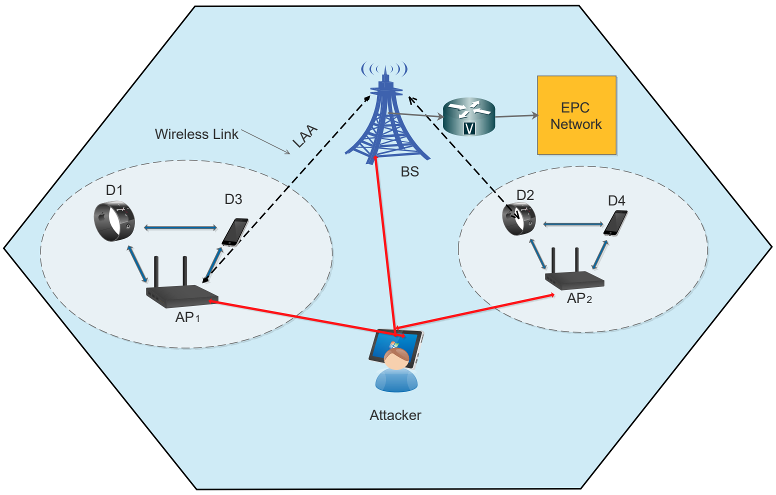


Figure 5. A C-IoT communication framework in a local area network.

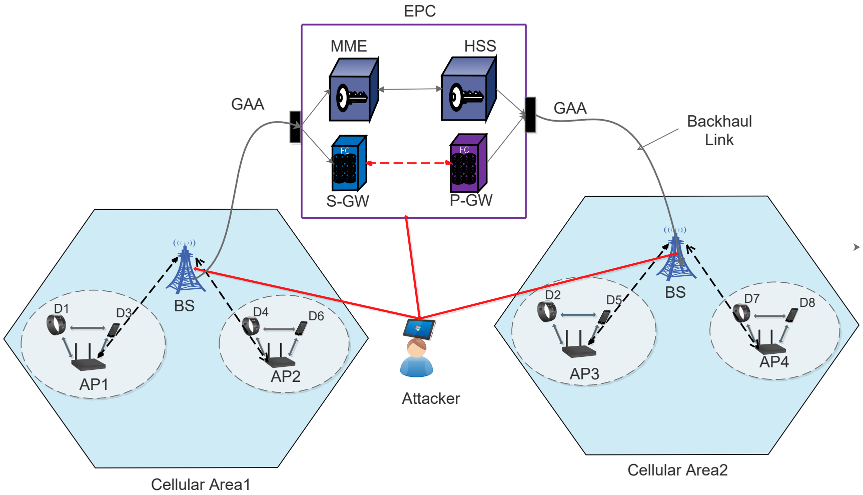


Figure 6. A C-IoT communication framework in global area network scenario.

The end-to-end authentication of the proposed ML-AKA protocol is carried out in four different steps, as shown in Figure 7. The steps are listed below:

- *System model setup:* In the system model setup stage, different devices generate some common parameters and create key agreements for D2D communication.
- *Roaming registration of devices:* See below for explanations of the roaming registration of devices, which will allow us to achieve authentic D2D communication.
  (1) Devices (D_i where i = 1,2) deal with the registration to the AP_i in order to acquire local area resources to take care of local area authentication (LAA). The device identities (DID_1, DID_2) are used for device registration requests that devices send to their AP.
  (2) After the registration request from D_i to the AP_i has been accepted, the authentication request js sent to the eNodeB; each request has its own identity ($APID_i$) for verification purposes.
  (3) After accepting the $APID_i$ authentication request by the eNodeB, the server checks the legitimacy of the access points through the $APID_i$. If the authentication request is not pinged by the legitimate AP, eNodeB simply rejects the authentication request and halts the process. When the authentication request comes from the authorized devices (D_i) and access point ($APID_i$), the eNodeB generates authentication information that contains a roaming key Ni_{asme}. The roaming key is calculated with the use of a cryptographic derivative function (KDF).

  $$Ni_{asme} = KDF(N_i, APID_i, rand_i) \tag{1}$$

  where N_i is the key shared in between device (D_i) and its nearest eNodeB with an $rand_i$.
  (4) After obtaining the authentication information (Ni_{asme}), the AP_i and D_i mutually authenticate each other. Along with the authentication and verifying the legitimacy, both the D_i and AP_i derive the D2D function key, which is Ni_{D2D} with the use of roaming key Ni_{asme}.

  $$Ni_{D2D} = KDF(Ni_{asme}, fun_{id}, rand_i) \tag{2}$$

  where fun_{id} is the D2D function entity identity and $rand_i$ represents random keys chosen by AP_i
  The D2D session key is generated by the D2D fun key Ni_{D2D}.
- *D2D connection establishment:*
  In the D2D connection establishment phase, two nearest devices (D_1 and D_2) discover each other and share a random secret key (R_k). If the shared random secret key (R_k) matches, a D2D connection is established.
- *Session key generation for D2D:*
  After the secure D2D or D2X connection has been established, the session key generation process begins, which is explained in the following steps:
  (1) There is one device (D_1) ping for the generation of session key process through sending the device to device session request to their access points (AP_1). Device identities keys like DID_1 and DID_2 are used for D2D session request.
  (2) After receiving the session request, AP_1 verifies the legitimacy of the respective devices. Session requests are only accepted for legitimate devices. Only one session identity is used, and this is denoted as SID_1. After that, AP_1 picks the random nonce value r_1 and sends a key agreement request, which consists of DID_1, DID_2, r_1 and SID_1.
  (3) Once the AP_2 has sent the key agreement request, it confirms the legacy of AP_1 and approves the D2D services. After selecting a new random nonce value r_2, AP_1 delivers the device identity (DID_1, DID_2) to AP_1.
  (4) With the sharing of their IDs, AP_1 and AP_2 generate a random pre-shared key (K_{ps}) with the use of XOR functions. After the key, K_{ps}, has been generated, it is sent to both the devices (D_1, D_2) attached to the eNodeB to establish a session. During this session, the management and key exchange request sent from AP_1 to D_1 contains D_2's identity

DID_2, K_{ps} and session identity SID_1, and the request sent from AP_2 to D_2 contains DID_1, (K_{ps}) and SID_1.

(5) Devices D_1 and D_2 choose random nonce values a and b, which are used to generate the session key H_1 and H_2, respectively. Before the session keys are exchanged, the message authentication code MAC_1 and MAC_2 are computed for H_1 and H_2 with the use of hashed MAC (HMAC) and the common secret key K_{cm}.

$$MAC_1 = HMAC_{K_{cm}}\ (H_1, t_1) \tag{3}$$

$$MAC_2 = HMAC_{K_{cm}}\ (H_2, t_2) \tag{4}$$

where K_{cm} = R_{AP} XOR K_{ps}, t_1 and t_2 are timestamps from devices' (D_1, D_2) local clocks.

(6) After obtaining the H_1 and H_2, the devices (D_1, D_2) verify the HMAC by computing their common secret key K_{cm}. If the result of the verification is correct then a D2D session key K_{DS} = $(H_1)^b$ = $(H_2)^a$ is generated.

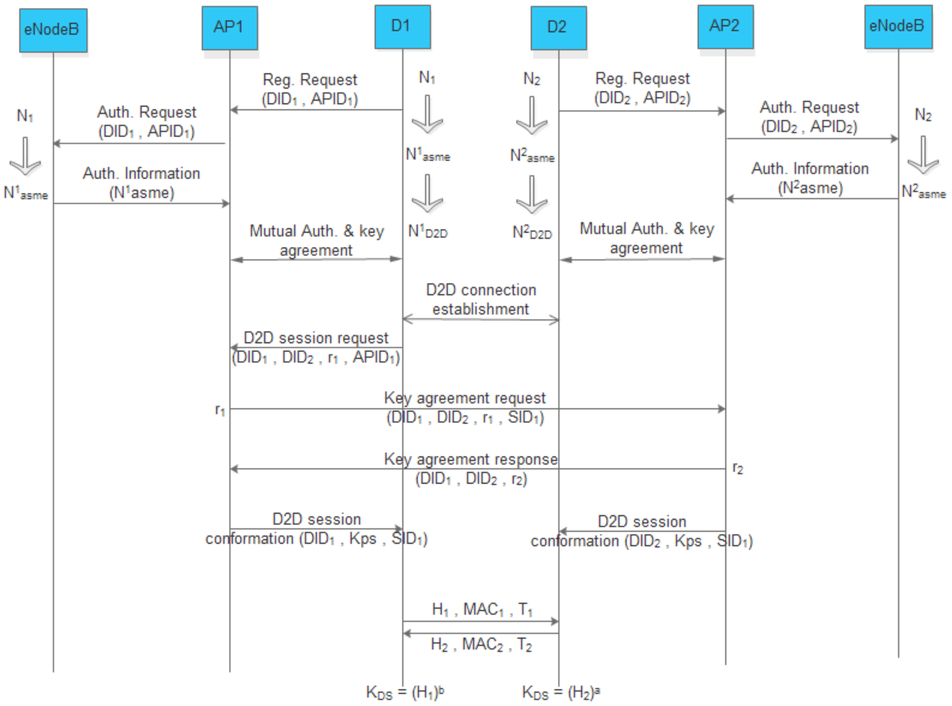

eNodeB
AP1
D1
D2
AP2
eNodeB
Reg. Request (DID₁ , APID₁)
N₁
N₂
Reg. Request (DID₂ , APID₂)
N₁
Auth. Request (DID₁ , APID₁)
Auth. Request (DID₂ , APID₂)
N₂
N¹asme
N²asme
N¹asme
Auth. Information (N¹asme)
Auth. Information (N²asme)
N²asme
N¹D2D
N²D2D
Mutual Auth. & key agreement
Mutual Auth. & key agreement
D2D connection establishment
D2D session request (DID₁ , DID₂ , r₁ , APID₁)
r₁
Key agreement request (DID₁ , DID₂ , r₁ , SID₁)
Key agreement response (DID₁ , DID₂ , r₂)
r₂
D2D session conformation (DID₁ , Kps , SID₁)
D2D session conformation (DID₂ , Kps , SID₁)
H₁ , MAC₁ , T₁
H₂ , MAC₂ , T₂
KDS = (H₁)ᵇ
KDS = (H₂)ᵃ

Figure 7. Authentication, registration, and acknowledgment flow diagram for ML-AKA protocol.

6.1. Accuracy of the ML-AKA Protocol

$$N^i_{asme} = KDF(N_i, eNodeB_{id}, rand_i) \tag{5}$$

$$N^i_{D2D} = KDF(N^i_{asme}, func_{id}, rand_i) \tag{6}$$

$$K_{ps} = r_1 XOR r_2 \tag{7}$$

$$K_{cm} = R_{AP} XOR K_{ps} \tag{8}$$

$$H_1 = (g)^a, H_2 = (g)^b \tag{9}$$

$$MAC_1 = HMAC_{K_{cm}}\ (H_1, t_1) \tag{10}$$

$$MAC_2 = HMAC_{K_{cm}}\ (H_2, t_2) \tag{11}$$

$$K_{DS} = (H_1)^b = (H_2)^a \tag{12}$$

6.2. Security Assessment

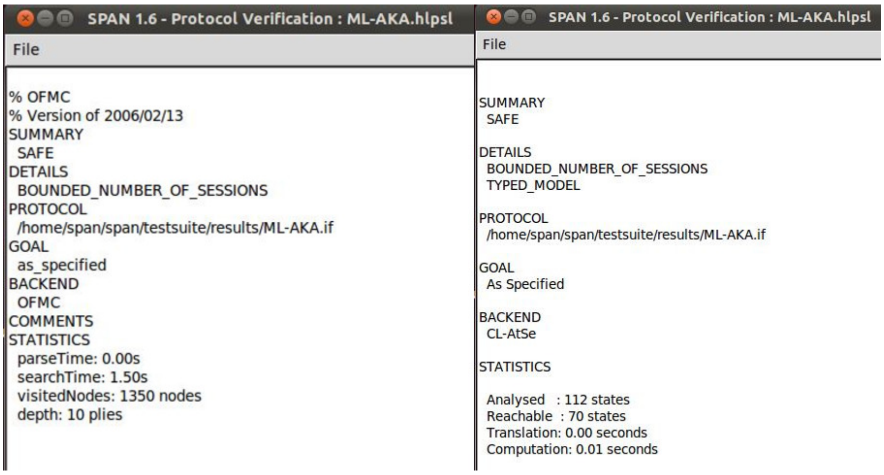
This section highlights the methodology and parameters required to analyze the proposed protocol. To analyze and validate our proposed ML-AKA protocol, we have used some formal simulation and validation tools. The objective details of the analysis are mentioned below:

- *Mutual authentication among the devices present in the network:* The ML-AKA protocol follows the security framework of EPS architecture and inherits the security features for ML-AKA device-to-device communication and the mutual authentication between the devices and APs. The ML-AKA protocol uses HMAC functions and some secret authentication keys like R_{AP} and K_{ps} to perform authentication between devices. In order to compute the common secret key K_{cm} and achieve the MAC verification, the shared secret R_{AP} and K_{ps} are used. The common secret K_{cm} is used to generate MAC_1 and MAC_2 for H_1 and H_2, respectively, and is also used to authenticate the identity of devices which are communicating with each other.
- *Manage session with the use of session keys:* Devices communicate with each other for a particular session prior to the exchange of actual information. This session is created by generating and exchanging some session keys between the devices. In ML-AKA protocols, the generated session keys are exchanged between entities that are present across various levels.
- *Enhance session key privacy:* In cellular-based IoT communication, heterogeneous types of devices are connected to each other by sharing secret keys through eNodeB and the core network. The primary objective of core network entities like MME and HSS is to maintain this privacy and preserve the identity of the shared secret keys. However, in some cases, if the LTE core network entities are compromised, data exchanged between the IoT devices can be easily evaluated by the third party. To avoid this, the ML-AKA protocol disallows outside attacks, meaning that attackers are unable to access the plaintext.
- *Reduce the probability of security attacks at the network junction:* In the traditional C-IoT network, the channels between the APs are secured with the help of a standard EPS-AKA protocol under the 3GPP standard. But communication channels between APs and IoT devices are secured by the standard Wi-Fi Protected Access (WPA) protocol under the IEEE 802.11 standard. Due to this difference in the protocols and authentication mechanism, unauthorized malicious attackers may target the transition points. To reduce the likelihood of this occurring, the ML-AKA protocol uses a common key that can be used interoperatively in all the forms of communication passing through the C-IoT architecture.

7. Simulation and Protocol Performance Analysis

We simulate the proposed protocol using Automated Validation of Internet Security Protocols and Applications (AVISPA v 1.1) and ProVerif tools in a Linux OS environment on a system with ACPI X86 based on an intel-i7 processor with 8GB RAM. By keeping the simulation environment constant, we have simulated three different protocols that are commonly used for IoT environments and are closely related to the proposed ML-AKA protocol. The test parameters, including search time (Ts) and parse time (Tp), are evaluated by increasing the number of test nodes.

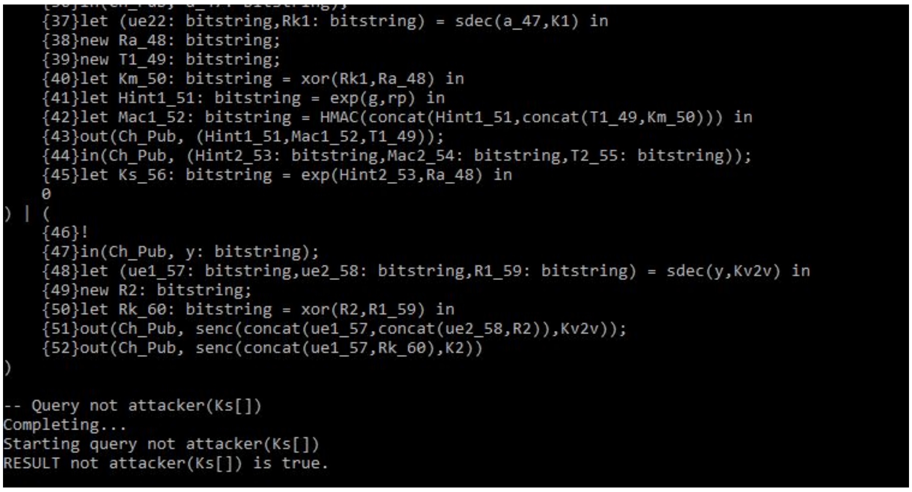
During the evaluation, different protocols performance metrics, such as communication cost and response time, are considered. In order to keep the comparison fair, the same testing environment and hardware are utilized. The simulation results, as shown in Table 3 and Figure 8, show that the Ps and Ts time is reduced by a significant amount by using the proposed ML-AKA protocol. It also shows that the security level's depth can be increased by using the ML-AKA protocol. For software analysis, the Python programming language and the Python Cryptography Toolkit (PyCrypto) are utilized, and the verification result is shown in Figure 9.


SPAN 1.6 - Protocol Verification : ML-AKA.hlpsl
File
SPAN 1.6 - Protocol Verification : ML-AKA.hlpsl
File

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/ML-AKA.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 1.50s
  visitedNodes: 1350 nodes
  depth: 10 plies
```

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/ML-AKA.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 112 states
  Reachable  : 70 states
  Translation: 0.00 seconds
  Computation: 0.01 seconds
```

Figure 8. Simulation result of ML-AKA protocol using AVISPA tools.



```
    {36}in(Ch_Pub, a_47: bitstring);
    {37}let (ue22: bitstring,Rk1: bitstring) = sdec(a_47,K1) in
    {38}new Ra_48: bitstring;
    {39}new T1_49: bitstring;
    {40}let Km_50: bitstring = xor(Rk1,Ra_48) in
    {41}let Hint1_51: bitstring = exp(g,rp) in
    {42}let Mac1_52: bitstring = HMAC(concat(Hint1_51,concat(T1_49,Km_50))) in
    {43}out(Ch_Pub, (Hint1_51,Mac1_52,T1_49));
    {44}in(Ch_Pub, (Hint2_53: bitstring,Mac2_54: bitstring,T2_55: bitstring));
    {45}let Ks_56: bitstring = exp(Hint2_53,Ra_48) in
    0
) | (
    {46}!
    {47}in(Ch_Pub, y: bitstring);
    {48}let (ue1_57: bitstring,ue2_58: bitstring,R1_59: bitstring) = sdec(y,Kv2v) in
    {49}new R2: bitstring;
    {50}let Rk_60: bitstring = xor(R2,R1_59) in
    {51}out(Ch_Pub, senc(concat(ue1_57,concat(ue2_58,R2)),Kv2v));
    {52}out(Ch_Pub, senc(concat(ue1_57,Rk_60),K2))
)

-- Query not attacker(Ks[])
Completing...
Starting query not attacker(Ks[])
RESULT not attacker(Ks[]) is true.
```

Figure 9. Outcome of protocol variation with the help of ProVerif tool.

Table 3. Comparison between different protocols.

Protocol Name	Parse Time (Tp)	Search Time (Ts)	Visited Nodes	Depth
EPS-AKA	0.031 s	2.22 s	1360 nodes	10 plies
UAKA-D2D	0.043 s	1.73 s	1360 nodes	10 plies
ML-AKA	0.01 s	1.40 s	1360 nodes	10 plies

7.1. Protocol Performance Analysis

The proposed ML-AKA protocol is well matched with the existing LTE core network EPS authentication and key agreement protocol for a secure device-to-device communication. Afterward, the comparison was made based on the existing protocol with respect to communication overhead.

7.2. ML-AKA Protocol Computational Cost

End-to-end communication involves three kinds of system entities:

- Devices (D_i).
- Access Points (AP_i).
- eNodeB.

In the system setup model, the different networks arrange the common secret parameters for device-to-device communication, and the computation cost of ML-AKA protocol depends on the size of the security parameters, e.g., len_p.

In the roaming registration stage of devices, eNodeB calculates the roaming key and all the authentication information, while AP calculates and generates the device-to-device (D2D) fun key. We have taken the authentication information generation time cost as T_{AI}, while T_r represents the random operation generation, T_{Kasme} is the roaming key generation, and T_{KD2D} is the D2D fun key.

While generating the session key for D2D, the AP uses the rand function and computes the N_i which uses an XOR function to compute the pre-shared secret N_i. Then, each device (D_i) conducts the following exponentiation operation and two HMAC operations:

$$T_{Ks} = T_r + T_{XOR} + 2T_{modExp} + 2T_{HMAC} \tag{13}$$

where one XOR fun is used to calculate K_m, two modular exponentiation processes are implemented for the session key and hint calculation, and two HAMC fun operations are used for the validation and mac generation process.

The communication cost of the ML-AKA protocol is analyzed in Table 4 and the calculated result of security system parameters generation and session key generation operations is shown in Figure 10. The results show that when the size of the security key length increases, the cost in terms of time also increases for all operations. Exceeding a length of 128 bits results in a linear increase in time cost. The validation result of ML-AKA also shows that, despite the larger size of len_p, the session key generation time and MAC verification time should be approximately constant. Furthermore, to assess the performance of the proposed ML-AKA, it is compared with various existing protocols in terms of attack mitigation techniques. Table 5 outlines the ability of different protocols to mitigate various types of possible attacks in C-IoT networks. In the table, a ✓ indicates capability, while a × denotes incapability. From the table, it was observed that ML-AKA can able to mitigate DDoS, spoofing, MiTM, and reply attacks. Whereas other considered protocols are only able to mitigate some selective attacks.

Table 4. Communication cost computation of ML-AKA protocol.

Device Name	Phase	Time
Devices (D_i)	Phase 2	$2T_{KDF}$
	Phase 3	-
	Phase 4	$T_r + T_{xor} + 2T_{modExp} + 2T_{HMAC}$
Access Points (AP_i)	Phase 2	$T_r + T_{KDF}$
	Phase 3	-
	Phase 4	$T_r + T_{xor}$
eNodeB	Phase 2	$T_r + T_{xor}$
	Phase 2	-
	Phase 4	-

Table 5. Comparison of different protocols in terms of attack mitigation.

	EPS-AKA	UAKA-D2D	ML-AKA
DDoS Mitigation	✓	✓	✓
Spoofing Attack	×	✓	✓
MIMT Attack	×	×	✓
Reply Attack	✓	✓	✓

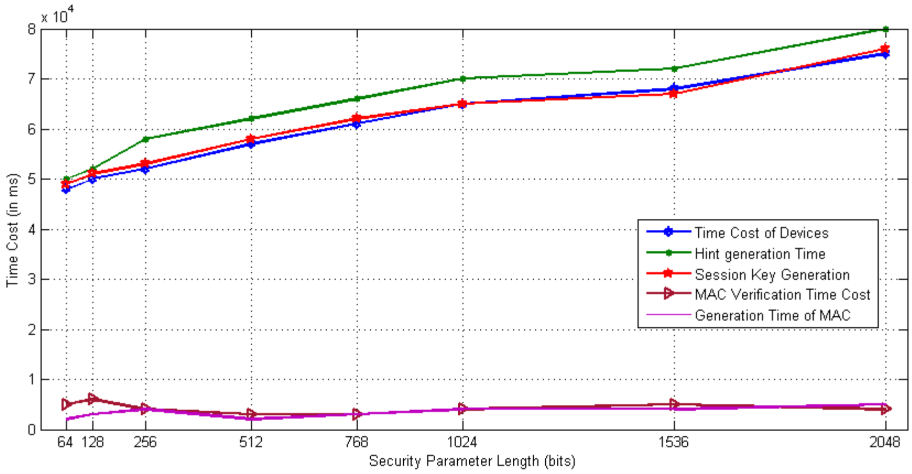


Figure 10. Time cost of Phase IV operations.

7.3. Communication Overhead

After the analysis of the ML-AKA protocol total communication overhead was calculated and compared with other existing protocols. We initially fixed the size of each security system variable to calculate the communication overhead. Table 6 listed all the security parameters used to calculate the communication overhead. The communication overhead of all the entities (Devices, APs, and eNodeBs) was calculated during the device key agreement and session establishment phase. In [4], the EPS-AKA communication overhead was calculated based on the security key length, which was $708 + 608 * a$ bits, where a is the authentication vector count. In the session key generation phase, the communication overhead of the device-to-device session request from devices (D_i) to access points (AP_i) is $(2 * 128 + 64 = 320)$ bits; D1 and D2 represent the first part of the data overhead and APID represents the second part of the data overhead.

Table 6. Length of the security parameters (bits).

Length of the Security Parameters (Bits)	
Parameters	**Size (Bits)**
DID	128
eNodeB/APID	64
SID	64
hint	len_p
r_i, R_k	len_p
MAC_i	256

Figure 11, shows that the maximum communication overhead is an exponential operation of key H_i and session key generation. Figure 12 compares the communication overhead for the key agreement phase for the EPS-AKA, UAKA-D2D and ML-AKA protocols. The outcome shows that the communication overhead is constant for EPS-AKS for all input bit lengths, whereas for UAKA-D2D and ML-AKA protocols, it increases linearly with reference to the increase in the bits length of each key. The outcome also shows that the ML-AKA is 5–10% and 20–50% more efficient than UAKA-D2D and EPS-AKA, respectively, under the same simulation conditions and different bit lengths. Similarly, Figure 13, compares the communication overhead required for a different protocol for the session establishment step. The figure illustrates a steady-state relationship between communication overhead and security parameter bit length for EPS-AKA, whereas for UAKA-D2D and the proposed ML-AKA, the communication overhead increases with the increase in bit length.

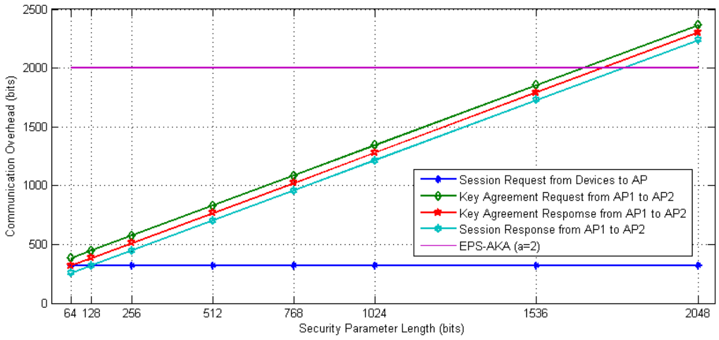


Figure 11. Communication overhead comparison for key agreement and session establishment using the ML-AKA protocol.

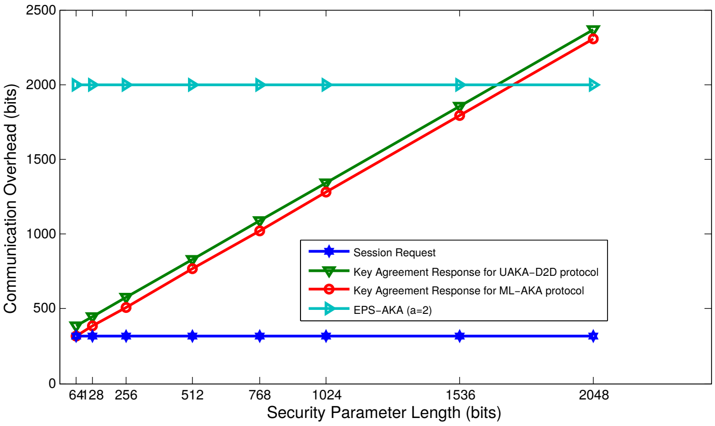


Figure 12. Comparison of communication overhead for key agreement step.

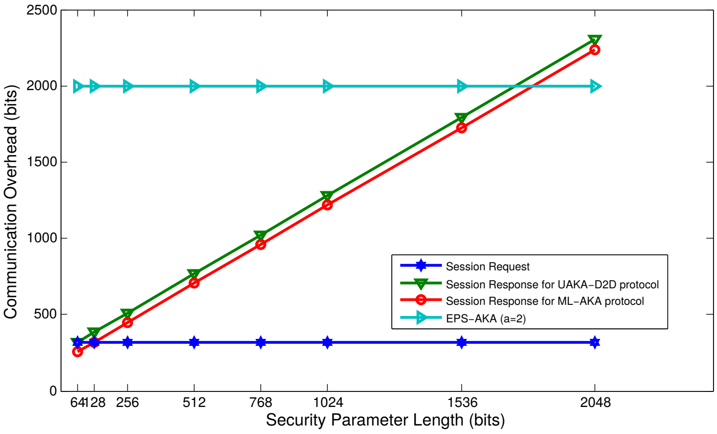


Figure 13. Comparison of communication overhead for session establishment step.

8. Conclusions and Future Scope

The advancement of IoT technology, coupled with mobile and wireless systems for long-distance data transmission, has introduced increasingly complex security challenges. The LTE-based C-IoT represents a new approach that has been developed for long-range IoT applications. In this research, we present a novel ML-AKA security protocol aimed at providing three-layer mutual authentication in a C-IoT architecture. The ML-AKA protocol incorporates an improved approach to D2D authentication. We employ an enhanced symmetric key cryptography method that provides greater flexibility for D2D authentication and validation in a C-IoT environment. The simulation results, presented in Section 7, indicate that the proposed ML-AKA protocol achieves higher efficiency in terms of execution time and authentication delay. Additionally, the ML-AKA protocol exhibits lower communication overhead compared to existing protocols.

The proposed protocol is tested over a small homogeneous network environment and achieves a significant improvement in the reduction of spoofing and DDoS attacks. However, establishing a secure authentication mechanism for a large network with heterogeneous types of devices requires a complex key distribution mechanism instead of the use of uniform keys distribution. Future work on this topic will consist of the design and development of a complex key distribution and authentication mechanism that can be used for next-generation cellular and IoT networks.

Author Contributions: Conceptualization, B.M.; methodology, B.M.; software, B.M.; validation, B.M. and V.S.; formal analysis, B.M., R.B., C.R.S. and V.S.; Investigation, B.M., R.B., C.R.S. and V.S.; resources B.M., R.B., C.R.S. and V.S.; data curation, B.M.; writing—original, B.M.; writing—review and editing, B.M., R.B., C.R.S. and V.S.; visualization, B.M., R.B., C.R.S. and V.S.; supervision, V.S.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the articlel. Further inquiries can be directed to the corresponding author(s).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Atzori, L.; Iera, A.; Morabito, G. The internet of things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805. [CrossRef]
2. Evans, D. The Internet of Things How the Next Evolution of the Internet Is Changing Everything. Available online: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf (accessed on 10 April 2011).
3. Saxena, N.; Grijalva, S.; Chaudhari, N.S. Authentication protocol for an IoT-enabled LTE network. *ACM Trans. Internet Technol. (TOIT)* **2016**, *16*, 25. [CrossRef]
4. Alam, M.; Yang, D.; Rodriguez, J.; Abd-alhameed, R. Secure device-to-device communication in LTE-A. *IEEE Commun. Mag.* **2014**, *52*, 66–73. [CrossRef]
5. Wang, M.; Yan, Z.; Niemi, V. UAKA-D2D: Universal Authentication and Key Agreement Protocol in D2D Communications. *Mob. Netw. Appl.* **2017**, *22*, 510–525. [CrossRef]
6. Raghothaman, B.; Deng, E.; Pragada, R.; Sternberg, G.; Deng, T.; Vanganuru, K. Architecture and protocols for LTE-based device to device communication. In Proceedings of the Computing, Networking and Communications (ICNC), San Diego, CA, USA, 28–31 January 2013; pp. 895–899.
7. Forsberg, D.; Horn, G.; Moeller, W.D.; Niemi, V. *LTE Security*; John Wiley & Sons: Hoboken, NJ, USA, 2012.
8. Jover, R.P. Security and impact of the IoT on LTE mobile networks. In *Security and Privacy in Internet of Things (IoTs): Model, Algorithms, Implementations*; CRC Press: Boca Raton, FL, USA, 2015; Volume 6.
9. Hu, F. *Security and Privacy in Internet of Things (IoTs): Models, Algorithms, and Implementations*; CRC Press: Boca Raton, FL, USA, 2016.
10. Yao, J.; Wang, T.; Chen, M.; Wang, L.; Chen, G. GBS-AKA: Group-based secure authentication and key agreement for M2M in 4G network. In Proceedings of the 2016 International Conference on Cloud Computing Research and Innovations (ICCCRI), Singapore, 4–5 May 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 42–48.
11. Sun, Y.; Cao, J.; Ma, M.; Li, H.; Niu, B.; Li, F. Privacy-preserving device discovery and authentication scheme for D2D communication in 3GPP 5G HetNet. In Proceedings of the 2019 International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 18–21 February 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 425–431.
12. Yan, X.; Ma, M.; Su, R. Efficient Group Handover Authentication for Secure 5G-Based Communications in Platoons. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 3104–3116. [CrossRef]
13. Li, G.; Lai, C. Platoon handover authentication in 5G-V2X: IEEE CNS 20 poster. In Proceedings of the 2020 IEEE Conference on Communications and Network Security (CNS), Virtually, 29 June–1 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–2.
14. Ranaweera, P.; Yadav, A.K.; Liyanage, M.; Jurcut, A.D. A Novel Authentication Protocol for 5G gNodeBs in Service Migration Scenarios of MEC. *IEEE Trans. Dependable Secur. Comput.* **2024**, *21*, 2930–2948. [CrossRef]
15. Wang, M.; Yan, Z. A survey on security in D2D communications. *Mob. Netw. Appl.* **2017**, *22*, 195–208. [CrossRef]
16. Doppler, K.; Rinne, M.; Wijting, C.; Ribeiro, C.B.; Hugl, K. Device-to-device communication as an underlay to LTE-advanced networks. *IEEE Commun. Mag.* **2009**, *47*. [CrossRef]
17. Køien, G.M. Mutual entity authentication for LTE. In Proceedings of the Wireless Communications and Mobile Computing Conference (IWCMC), Istanbul, Turkey, 4–8 July 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 689–694.
18. Zhang, M.; Fang, Y. Security analysis and enhancements of 3GPP authentication and key agreement protocol. *IEEE Trans. Wirel. Commun.* **2005**, *4*, 734–742. [CrossRef]

19. Liu, Y.; Xu, Y.; Li, D.; Wang, W. Device-to-device communication in LTE-A cellular networks: Standardization, architecture, and challenge. In Proceedings of the Vehicular Technology Conference (VTC Spring), Seoul, Republic of Korea, 18–21 May 2014; pp. 1–5.
20. Wang, M.; Yan, Z. Security in D2D communications: A review. In Proceedings of the Trustcom/BigDataSE/ISPA, Helsinki, Finland, 20–22 August 2015; Volume 1, pp. 1199–1204.
21. Muthana, A.A.; Saeed, M.M. Analysis of user identity privacy in LTE and proposed solution. *Int. J. Comput. Netw. Inf. Secur.* **2017**, *9*, 54. [CrossRef]
22. Alezabi, K.A.; Hashim, F.; Hashim, S.J.; Ali, B.M. An efficient authentication and key agreement protocol for 4G (LTE) networks. In Proceedings of the Region 10 Symposium, Kuala Lumpur, Malaysia, 14–16 April 2014; pp. 502–507.
23. Jumaa, N.K. Implementation of Enhanced AKA in LTE Network. *Int. J. Comput. Sci. Mob. Comput.* **2015**, *4*, 1124–1132.
24. Armando, A.; Basin, D.; Boichut, Y.; Chevalier, Y.; Compagna, L.; Cuéllar, J.; Drielsma, P.H.; Héam, P.C.; Kouchnarenko, O.; Mantovani, J.; et al. The AVISPA tool for the automated validation of internet security protocols and applications. In Proceedings of the International Conference on Computer Aided Verification, Edinburgh, UK, 6–10 July 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 281–285.
25. Ramadan, M.; Li, F.; Xu, C.; Mohamed, A.; Abdalla, H.; Ali, A.A. User-to-User Mutual Authentication and Key Agreement Scheme for LTE Cellular System. *IJ Netw. Secur.* **2016**, *18*, 769–781.
26. Aiash, M.; Mapp, G.; Lasebae, A.; Phan, R. Providing security in 4G systems: Unveiling the challenges. In Proceedings of the 2010 Sixth Advanced International Conference on Telecommunications, Barcelona, Spain, 9–15 May 2010; pp. 439–444.
27. Bikos, A.N.; Sklavos, N. LTE/SAE security issues on 4G wireless networks. *IEEE Secur. Priv.* **2013**, *11*, 55–62. [CrossRef]
28. He, D.; Wang, J.; Zheng, Y. User authentication scheme based on self-certified public-key for next generation wireless network. In Proceedings of the 2008 International Symposium on Biometrics and Security Technologies, Islamabad, Pakistan, 23–24 April 2008; pp. 1–8.
29. Yue, J.; Ma, C.; Yu, H.; Zhou, W. Secrecy-based access control for device-to-device communication underlaying cellular networks. *IEEE Commun. Lett.* **2013**, *17*, 2068–2071. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.