future internet

MDPI

Article

A Lightweight Elliptic-Elgamal-Based Authentication Scheme for Secure Device-to-Device Communication

Adeel Abro, Zhongliang Deng and Kamran Ali Memon *

School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China; adeelabro@bupt.edu.cn (A.A.); dengzhl@bupt.edu.cn (Z.D.)

* Correspondence: ali.kamran77@gmail.com

Received: 13 March 2019; Accepted: 26 April 2019; Published: 7 May 2019

Abstract: Device-to-Device (D2D) is a major part of 5G that will facilitate deployments with extended coverage where devices can act as users or relays. These relays normally act as decode and forward relays (semi-intelligent devices) with limited computational and storage capabilities. However, introducing such a technology, where users can act as relays, presents a wide range of security threats, in particular, rogue relay devices or man in the middle attacks (M-I-T-M). Second, passing fewer control messages is always advisable when considering authenticity and secrecy. To mitigate M-I-T-M and to reduce communication costs, this paper presents a lightweight elliptic-ElGamal-based authentication scheme using PKI (FHEEP) in D2D communication. Pollard's rho and Baby Step, Giant Step (BSGS) methods are used to evaluate the authenticity and secrecy of our proposed scheme. The communication cost is calculated based on the comparative analysis indicating that our proposed scheme outperforms the baseline protocol. The proposed scheme can be used for any infrastructure architecture that will enhance the security of any D2D settings with better performance.

Keywords: 5G; D2D; lightweight authentication scheme; elliptic curve cryptography; ElGamal; Man-in-the-middle attack

1. Introduction

Device-to-Device communication (D2D) is one of the major technologies that 5G will present. It is based on LTE-B, also referred to as LTE REL-12. D2D is a novel and promising technology that allows communication between two or more devices in proximity with no need to go to the leading network. Thus, it will be a relief technology for areas that have very weak or no coverage. It has a high data rate and low latency, and thus it is amenable to low computation devices. IoT devices can communicate with each other and also with other devices such as smartphones. These smartphones can provide connectivity to nearby devices and can work as a small base station. This has led to new concepts such as Device-to-Device (D2D) communication where devices can provide connectivity to nearby devices, allowing higher bandwidth and extended coverage at no cost. As D2D is an integral part of 5G, it inherits the properties of 5G, such as high data rates and low latency. There are two forms of communication, as shown in Figure 1 [1]: (a) Inband and (b) Outband.

No cost indicates that installation of a new base station is not required and that it can be used between near-proximity devices. D2D is becoming the choice of all network operators as it allows higher bandwidth, more users, increased coverage and high data rates, particularly in sparse environments [2,3]. D2D comes in two settings: (a) With existing cellular infrastructure also referred as network assisted [4] (b) standalone settings where there is local connectivity in terms of time or geographic interests [4,5]. D2D offers several data sharing techniques that allow multiple technologies to provide connectivity.

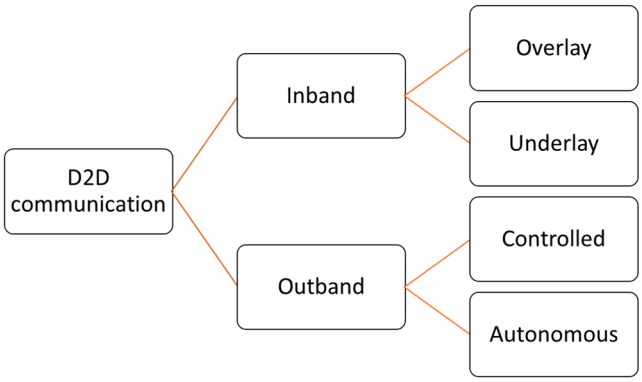

Figure 1. Types of D2D communication.

Some examples are Relay-By-Smartphone [5] and Data Spotting [6], and FlashlineQ [5] can provide connectivity up to 1 KM [6]. These devices can act as semi relays in a D2D network and can perform channel selection for underlying devices [1]. In D2D, these relays can act as transparent relays (TR) (where they just amplify and forward) or non-transparent relays (NTR) (i.e., decode and forward). A problem arises when one of the NTR become rouge, compromising the whole communication. Rouge Nontransparent relay NTR, as identified by the 3GPP security workgroup, can compromise credentials and launch user data attacks [7]. Attacks can occur on user privacy and resource availability as well as data integrity, authentication, and confidentiality [8]. Rouge NTR can result in interleaving attacks, one of the most successful types of attack that inflicts damage in cases of both success and failure. If an interleaving attack is successful, it is a Man in the Middle Attack (M-I-T-M). If it is a failure, it is a Denial of Service, aka DoS, attack [9]. Rouge NTR can result in other attacks such as replay attacks and authentication attacks, where an adversary can act as a legitimate user. To address these attacks, a security and authentication scheme is required that can handle M-I-T-M, Replay and Impersonation attacks. These devices have minimal computational and storage power and thus require the security algorithm to be computationally less expensive while exceeding the capabilities of existing algorithms. Stallings [10] required all security algorithms to handle important types of attacks, particularly the denial of service (DOS), eavesdropping, interleaving, masquerading, Man-in-the-Middle (M-I-T-M) Impersonation, Malware attacks, and trust manipulation attacks.

Elliptic curve cryptography (ECC) is a public key cryptography technique [11]. It is based on the algebraic structure that uses finite keys. It was firstly proposed in 1985 but has gained importance in current years due to its security strength and key size comparison to other public key security algorithms [12]. ElGamal is a public key cryptography technique proposed by Taher ElGamal and is based on Deffie Hellman Key Exchange (DHKE). There is a possibility of using ElGamal with ECC, several researches have successfully conducted, and results are promising in terms of security analysis. SHA is a hashing technique proposed firstly by the National Institute of Standards and Technology (NIST) [9]. There are several varieties of this hashing technique such as SHA-1 that was using a 160 bits hash. SHA2 goes from 224 to 512 bits hash, while SHA-3 also goes from 224 to 512 bits has. The only difference is base input that is 256 and 1600 respectively.

This paper presents a lightweight security scheme that is based on PKI called FHEEP. FHEEP uses the great ECC scheme to reduce key size and provide extended security based on the Discrete Logarithm Problem (DLP). It uses ECC over ElGamal as it can achieve semantic security [13]. FHEEP uses ECC for key selection based on Elliptic Curves approved by the National Institute of Standards and Technology (NIST). It is based on PKI infrastructure where the certificates have already been obtained from certification Authority CA.

2. Related Works

D2D networks [14–16] are key networks that will serve the communication, coverage, and data rate demands for any future applications. Currently, D2D has received much attention from academia and

industry due to its potential advantages and reliable Quality of Service (QoS). There are several services in mobile D2D networks that are time critical, and their time requirement depends on their requirement or purpose, limiting it to seconds [17,18]. The emerging mobile D2D heterogeneous network focuses on providing applications that will support all wireless communication ubiquitously, i.e., in all places and at all times [18,19]. Several approaches are used to mitigate the associated attacks on heterogeneous mobile networks. Kim et al. [19] present an asymmetric scheme for broadcasting in order to achieve low computation overhead and complexity. It used a broadcast encryption scheme and forward was performed in grouping for keys. It was also based on ECC. This technique proved more efficient than other key sharing schemes. Laiphrakpam et al. [20] presents a Deffie Hellman (DH) with ECC using a simple hash algorithm for secure transmission of images using Arnold's transform that allows encryption. The resultant algorithm provided confidentiality and proved resilient against Brute force attacks. Ni et.al [21] proposed an algorithm based on Deffie Hellman for IoT services in 5G that was used for groups. It allows a ticketing system for the relay for fog nodes. The fog nodes are responsible for network slicing and data forwarding. This scheme also allows anonymous authentication for preserving privacy. Sharma et al. [22] presented a protocol based on the Deffie-Hellman key exchange for the security of Xhanul links. The scheme includes handoffs between nodes. The scheme provides security and privacy, but not all security requirements are met. The approach is also tested against DoS attacks and appears secure against major security attacks. Karati et al. [23] proposed a certificateless signature scheme to solve the key-escrow problem for lightweight devices. Key-escrow or "fair" cryptosystem is an arrangement in which the keys decrypt encrypted data, are retained in escrow to allow only an authorized private entity to access those keys under certain conditions. Their scheme is based on two exponentials for the selection of keys both from signers and signatories. It uses the hash function to provide integrity, and all computation is performed in the cloud. The scheme is considered secure against Bilinear Deffie Hellman attacks. Gritti et al. [24] presented a solution for identification of device attestation and message authentication using the local declaration of devices by creating a subnet of things. They introduced the selection of keys on multiple cyclic groups using bipartite graphs. The authors established a bootstrap mechanism for identification and authentication. Fang et al. [25] Identified attacks in 5G and device-to-device communication such as denial of service, jamming and MITM attacks. The authors proposed different methods for various security issues such as cyclic redundancy check for authentication. Frequency hopping was used to determine availability. For authentication, the hash of identity of the message was used. The authors used Deffie-Hellman for key exchange and suggested that there should be a trust model for device-to-device communication. Wu et al. [26] used an interlock protocol to avoid MITM attacks. The authors used the Magpairing scheme to make the connection between two ad hoc devices. Second, the magnetic field is also taken into consideration to check the proximity. Cipher block chaining was used to secure the data, and the protocol was compared with Deffie Hellman; the novel scheme proved to be better than the compared algorithms. Sedidi et al. [27] employed a Diffie-Hellman key exchange algorithm using normal version; (b) using the ACK/NACK messages for delivery of packets to the base station; and (c) using Macro station to send a verification code to verify the reception of data and provide authentication. It was shown that the techniques were able to mitigate the Man-in-the-middle attack as well as the problem of key distribution. In [28], the authors implemented Elliptic Curve Cryptography to secure against blackhole attacks. The authors used the scheme using Adhoc on Demand Multipath Distance Vector as it is a more reliable scheme. The authors claim that packet delivery can be secured against attacks using a simulated environment. Dake et al. [29] presented an algorithm with ECC and Elgamal and compared it with the ElGamal scheme to prove that usage of ECC with Elgamal is a better option. The comparison was only performed in terms of space and time. The comparison showed that their proposed algorithm performed better on small devices. Khan et al. [30,31] used a scheme based on Elgamal for securing the microgrid communication and ensured that it is helpful in the protection of nonrepudiation and replay attacks. Shah et al. [32] also highlighted the issues that can occur in device-to-device communication, and it emphasizes the requirement of central control. It uses receive

signal strength and channel state information to design the key, as well as an algorithm based on DH to perform key management and sharing. It has proved to be a good defense against man-in-the-middle attacks. In [33], the authors used a Deffie-Hellman and ECC scheme together to ensure that the key exchange is performed securely and is not vulnerable to known attacks. Being a lightweight, ECC can be adapted in lightweight communication. They have used this scheme for image conversion as images are considered memory intensive, and the result shows that the scheme can be adapted for transmission of images securely.

3. System Model

Figure 2 depicts the snapshot of the System model, where many devices are connected and participating in D2D communication. NTR is a non-transparent relay that provides extensive connectivity and coverage to nearby devices. MRBS is referred to as a master relay base station and is deployed by mobile network service providers; it is normally a high computational node. The certification authority (CA) has the responsibility of issuing the certificate to underlying nodes. Therefore, before any node provides communication, the facility must register with the CA that will publish the public key of all individuals in a whitelist maintained by the CA.

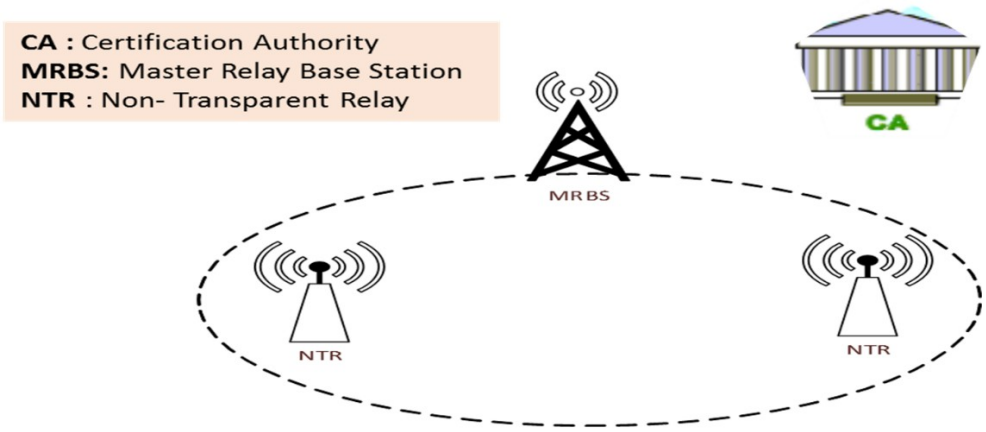

Figure 2. Showing system model diagram for the proposed solution.

Generation of public and private keys is achieved by the individual node to avoid the key-escrow problem. Thus, MRBS and NTR generate their public and private keys using Algorithm 1 with a huge key that can avoid DLP attacks.

- *Certification Authority (CA):* The Certification Authority (CA) is a globally recognized and trusted entity in which all nodes can trust. Its task is to verify the authenticity of nodes and publish the public key for each node on a secured publicly available Whitelist that can be accessed by each node ('node' indicates MRBS or NTR).
- *Master Relay Base Station (MRBS)* is a node that is already registered with the CA and provides connectivity of all underlying nodes. Cellular companies deploy MRBS
- *Non-Transparent Relay (NTR)* is a node that provides the facility of connection, communication, and coverage to underlying devices. These relays are decode and forward and thus have limited capabilities such as MRBS. NTR is very limited in power, storage and computation.

There are already two keys generated by MRBS and NTR with CA using Public Key Infrastructure (PKI). The keys are as follows:

$$\text{MRBS} = \{K_{MRBS},\ P_{MRBS}\}$$
$$\text{NTR} = \{K_{NTR},\ P_{NTR}\}$$

where K is the private key and P is the public key obtained through CA.

4. Our Cryptosystem

Our Cryptosystem is a dual security algorithm based on a combination of ECC with ElGamal using a SHA384 as the hashing algorithm and an embedded challenge response to avoid significant attacks. It is referred to as the dual security algorithm as for each session a new key is used for communication along with a stringent authentication mechanism. We built the algorithm on top of the base algorithm proposed in [34], but with improvements on the encryption and hashing process.

Initially, in Algorithm 1, the NTR wants to communicate with the MRBS, so the NTR needs to compute the points on Elliptic Curve EC that are broadcasted with basepoint B after a periodic interval. Step 1 in the algorithm says to compute K_{NTR} from Field F_N *where* $1 < K_{NTR} < N - 1$. It then computes public key P_{NTR} with a private key using Equation (1)

$$P_{NTR} = K_{NTR} \times B \tag{1}$$

where P_{NTR} K_{NTR} are keys for non-transparent relay.

Once P_{NTR} is computed, it will be shared with the MRBS, so it is sent to MRBS over the channel, as shown in Equation (2)

$$E[K_{NTR}(P_{NTR} + E[P_{MRBS}(CH)])] \tag{2}$$

Algorithm 1: Scheme for Encryption for Session transmission

AT NTR
Step 1: Select secret key K_{NTR}
Step 2: Compute Public Key $P_{NTR} = K_{NTR} \times B$
Step 3: Send temporal public key P_{NTR} to MRBS
$$E[K_{NTR}(P_{NTR} + E[P_{MRBS}(CH)])]$$
AT MRBS
Step 4: Decrypt to get P_{NTR}
Step 5: Select secret key K_{MB}
Step 6: Compute Public Key $P_{MB} = K_{MB} \times B$
Step 7: Compute the Secret key $SK = K_{MB} \times P_{NTR}$
Step 8: Encrypt the Message M to get the Cipher C_{MB}

The message is sent along with challenge CH, which is encrypted with the public key of MRBS P_{MRBS}, and then the key is encrypted with the private key of NTR K_{NTR}. Once the message arrives at the MRBS, it has to decrypt the message using the public key of the NTR P_{NTR} and then send the challenge out using its own private key K_{MRBS}, as shown in Equation (3):

$$\begin{gathered} D[P_{NTR}[K_{NTR}(P_{NTR} + E[P_{MRBS}(CH)]] = [(P_{NR} + \\ E[P_{MRBS}(CH)]P_{NTR}],\ D[K_{MRBS}[P_{MRBS}(CH)]]P_{NTR},\ CH \end{gathered} \tag{3}$$

Now, from P_{NTR}, the Secret key SK is calculated by performing point multiplication with the private key of MRBS, so step 4 addresses the selection of computer K_{MB} from Field F_N *where* $1 < K_{MB} < N - 1$. Equation (2) P_{MB} can be calculated using $K_{MB} \times B$. Now, the SK can be calculated using Equation (4):

$$SK = K_{MB} \times P_{NR} = (X_S, Y_S) \tag{4}$$

SK is considered a secret point of communication for both the MRBS and NTR in the on-going session. Now, this SK can be used to encrypt the Message M to obtain the Cipher C_{MB}. The Cipher is calculated using the point multiplication of SK and M, as shown in Equation (5).

$$C_{MB} = (C_{MB-X}, C_{MB-Y}) C_{MB-X} = (X_S \times MX_{original}) mod\ B C_{MB-Y} = (Y_S \times MY_{original}) mod\ B \tag{5}$$

The Cipher C_{MB} is then used to calculate the Hash H using SHA384 or SHA512, the second family of SHA using Equation (6):

$$H = Hash\ (C_{MB}) \tag{6}$$

Once the Hash is calculated, the communication incorporates Challenge CH into the solution by solving it using a simple challenge solver that has already been announced by the MRBS using Equation (7):

$$CH_1 = Solve(CH)Solve(CH) = Answer\ CH + New\ Challenge \tag{7}$$

For data authentication, the following procedures can be adapted. The new challenge CH_1 contains a solution to the old challenge, and the new challenge to be solved by the NTR is made. The challenges are added in to ensure that communication has occurred successfully, and the new challenge is sent with the same purpose of achieving surety; this message is received at the end.

To achieve authentication that the MRBS is the one sending the message, the signature S is calculated by encrypting the challenge CH_1 and Hash H with its main private key K_{MRBS}, as shown in Equation (8).

$$S = E[K_{MRBS}(H + CH_1)] \tag{8}$$

Once the signature S is obtained, the message can be sent to the NTR that will contain the cipher, hash and challenge along with Time stamp TS to maintain the freshness of the message, as shown in Equation (9).

$$E[P_{NR}(C_{MB}, \mathrm{S},\ P_{MB}, \mathrm{TS})] \tag{9}$$

Equation (9) includes the signature S, the session public key of MRBS P_{MB}, the time stamp TS to maintain the freshness of the message, and they are encrypted using the temporal public key of NTR P_{NTR}, which can be opened only by the private key of NTR K_{NTR}, as shown in Equation (10).

$$DK_{NTR}[E[P_{NTR}(C_{MB}, \mathrm{S},\ P_{MB},\ TS)]]C_{MB}, \mathrm{S},\ P_{MB},\ TS \tag{10}$$

Here, D is used for decryption, and E is used for encryption, as they both cancel each other, so the message can be obtained.

Once the message is out, time stamp TS is used to check the freshness of the message; if the time is more than the (Time to Live) TTL of the message, the message is discarded, and there is no further processing. If the TS matches the requirement, the NTR gets the public key from the CA using a whitelist maintained by the CA. This public key of MRBS P_{MRBS} will be used to decrypt the signature, as shown in Equation (11).

$$D[P_{MRBS}E[K_{MRBS}(H+CH_1)]] = H+CH_1 \tag{11}$$

The Hash H and Challenge CH_1 is obtained, and the challenge answer is verified by matching it with the CH that was sent by the NTR. If it matches, it verifies that the old message was received; else, it might have been lost or held by an attacker. It also solves CH_1 to obtain CH_1 if a new communication has to occur; else, this calculation is not required. Once the challenge test is passed, the hash of cipher text C_{MB} is taken again using Equation (6); if both the hashes are equal, it will verify that the message was not modified and is received, as shown in Equation (12):

$$H' = Hash\ (C_{MB}) \tag{12}$$

If $H' = H$, the test is passed; else, it is failed.

If all the requirements are met, the secret key is calculated using modified Equation (5) as the NTR uses its private key K_{NR} and the session public key of MRBS P_{MB} to get SK, i.e., $K_{NR} \times P_{MB}$. That will be used to decrypt the cipher text C_{MB} to obtain message M.

5. Analysis and Discussion

FHEEP is based on the security requirement of the D2D network; thus, it is necessary to analyze it in two aspects, security and the computational aspects.

5.1. Security Aspects

There are several security attacks that can be tested on our proposed cryptosystem; the first trial can be a brute force attack.

- Brute Force Attack

The adversary can know the primary public key of MRBS P_{MRBS} and NTR P_{NTR}, while it will also know the base point B and Elliptic curve EC. Now, the challenging task is to determine the private key K_{MRBS} or K_{NTR}; if one of them can be found, the entire communication can be compromised; for example, if the MRBS is compromised, the entire underlying system is as well, and if the NTR is compromised, the small cell network that the NTR is making will be compromised. Our proposed algorithm uses ECC as the key selection, while the size of the key is ≥ 384, making it more secure against modern attacks. ECC is based on DLP, which states that

$$a^x = M \tag{13}$$

where M is a known Multiplicative Group, a is a generator term from M, x is an element such that $x \in M$.

Therefore, the problem is to find the key with the following DLP equations of the following type:

$$P_{MRBS} = K_{MRBS} \times B \tag{14}$$

Where adversaries must find the K time point multiplication with base point B such that P is achieved, while the Elliptic curve works on an elliptic equation that effects the rotation with K, so the complexity becomes greater when the key is rotated M times.

Second, if this communication is cracked, our proposed algorithm uses session-based encryption using the same algorithm so that cracking occurs for each session. Thus, complexity will accumulate in each session. The opposite is true if one of the sessions is hacked (which is not possible), so only the session communication may become compromised and not the remaining sessions.

- Pollard's rho method [35]

This is an intelligent way to attack that is also lightweight in nature. It requires parallelization and random walk. It has been observed that it makes it possible to find the secret key by reducing to the square root of the attack. It has been theoretically shown in a few studies that with 10^4 computers, a key can be cracked in less than two years [36]; similarly, Bos et al [37] presented a scheme for solving 112 bits primes using a games console; all are based on the theoretical assumption of collusion. Our algorithm uses a key size of greater than 512, which is equal to 15,360 bits of RSA [9,37]. Thus, for the first key, even if the square root is taken, it will be fairly impossible to crack the problem. Figure 3 shows that the key size for the ECC does not increase considerably, but the key size for the RSA increases dramatically.

- Replay Attack

Replay attacks allow the adversary to store the message and then send the message at some other interval. This can result in various issues that lead to loss of assets, tracks, and other information. To avoid this attack, as can be seen in Equation (9), the time stamp is introduced. Depending on the

channel condition of each session, the TTL is determined, so if the TTL is more than the current time difference with the message time, the message is considered fresh, as shown in the following equation:

$$If\ (Current_{time} - Message_{time}) > TTL \tag{15}$$

Then, the message is fresh; else, message M is not fresh.

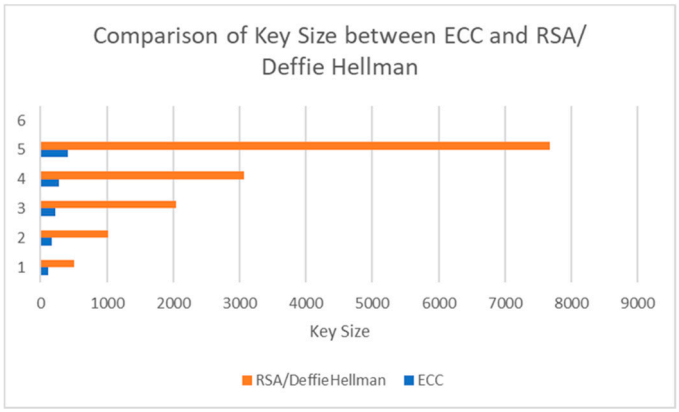

Figure 3. Key comparison of ECC concerning RSA and Deffie Hellman.

Thus, the replay attack cannot be performed on our proposed algorithm. There are several types of attack, such as TCPReplay, but as we have introduced the time stamp and nonce, the attack will be caught.

- M-I-T-M Attack

For a M-I-T-M attack that can result in interleaving or a DoS attack, we consider the Needam-schroder protocol [38] and let NTR be a rouge relay:

$$\text{MRBS} \rightarrow \text{NTR MRBS} \rightarrow \text{NTR}\ E[P_{NR}(C_{MB},\ \text{S},\ P_{MB}, \text{TS})]$$

Here, instead of the NTR, if some other adversary receives the message, it will not be able to open the message or even gain access to the cipher text. Thus, this attack will fail. We now consider another attack where a rogue NTR sends its public key to MRBS $E[K_{NTR}(\text{P}_{\text{NR}} + \text{E}[\text{P}_{\text{MRBS}}(\text{CH})])]$. This message can only be generated by a real NTR and not the other rouge NTR, as a rouge NTR message will be

$$E[K_{Rouge_{NTR}}(P_{NR} + E[P_{MRBS}(CH)])$$

This is not the intended party of communication; thus, the MRBS can ignore the message. It is seen that the main public key is obtained by the CA and not by the nodes; thus, changes to the M-I-T-M have failed, and the issue is solved.

- Baby Step, Giant Step (BSGS) Method

BSGS [35] method by shank is an impressive method for solving the DLP problem and focuses on collisions. It reduces complexity by $\sqrt{N}$ times, that is, half the size. If we are using a 192-bit curve, after taking $\sqrt{N}$, we have 10^{16} points; thus, it will require 10^{21} Zeta bytes to store the hash, but this is not a large figure compared to the recommendation of the proposed algorithm, which is on the order of 10^{156} Octillion attempts to make it fairly impossible to crack the key.

- Impersonation attack

The proposed algorithm is robust against authentication attacks where the adversary may pose as a legitimate node. The problem of rouge relay is becoming more common due to the introduction of

non-transparent relays [36]. To avoid these attacks, our proposed algorithm uses a two-way approach; first, it encrypts the message using its private key, as shown in the equation below.

$$[K_{NTR-p}(P_{NR} + E[P_{MRBS}(CH)])]$$

The MRBS will check the Whitelist maintained by the CA and obtain public key P_{NTR}, which will not open the message, and decryption will fail, as shown in the following equation.

$$D[P_{NTR}E[K_{NTR-p}(P_{NR} + E[P_{MRBS}(CH)])]] = Fail$$

Thus, the proposed algorithm is secure against impersonation attacks.

5.2. Computational Aspects

The proposed algorithm is better in terms of computation cost and memory consumption. The proposed algorithm uses ECC. ECC arithmetic is based on an operation that is performed in a finite field forming an Abelian group [39]. It uses point addition and point multiplication that is different from normal multiplication; it is faster than typical multiplication. ECC keys are very short, providing similar security compared to typical techniques such as RSA or DH, as shown in Table 1; they are extracted from [40–42]. It shows that usage of FHEEP will result in less computational overhead and handling of keys as the required bits are x number of times less than those of RSA and DH. It also shows that memory consumption and traffic over the network will be reduced due to fewer bits being set.

Table 1. Showing a key comparison of ECC, RSA, and DH.

FHEEP	RSA/DH
112	512
224	2048
571	15,360

5.3. Authentication Overhead

A normal authentication message is a four-step process where each party exchanges certain information to be authenticated to communicate, as explained in [34]. In authentication requests, a node may send its information or a challenge, while in response, the node may solve the challenge. Then, a secret key is shared that has to be verified by the second party. Considering this technique to be secure, our communication only occurs in one step out of the whole; even if the communication request considered in Figure 2 is ignored, the proposed algorithm will take two (2) steps as well as deliver the cipher text, and authentication is also performed, as shown in Figure 4. It is seen that the communication response is reduced by 4:1 to 5:1, including the communication request. Figure 4 shows that the communication cost is reduced compared to that of the main authentication protocol, and both results are achieved.

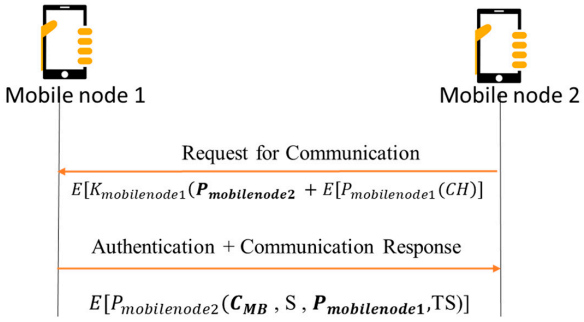

Figure 4. Showing the decreased communication cost between two nodes.

If we consider each communication step to be X joule [43], the total number of steps for traditional communication will be $4X$ joule, and the total session cost will be

$$C_{CostT} = \sum_{i=0}^{n} i(4X)$$

where i is the session number and n are total session number, while for our scheme the communication cost will be only X joules; thus, the total sessions cost will be

$$C_{CostP} = \sum_{i=0}^{n} i(X)$$

Thus, there is clearly a difference of 4:1 between the two techniques.

The first baseline protocol ULMAP [44] is only based on NFC and RFID tag authentication for a small number of devices, and the second baseline scheme Improved SIP [45] is only based on ECC and relays of several messages transferred between them.

As shown in Figure 5, the FHEEP outperforms the other protocols in terms of a number of message exchanges with similar security and trust.

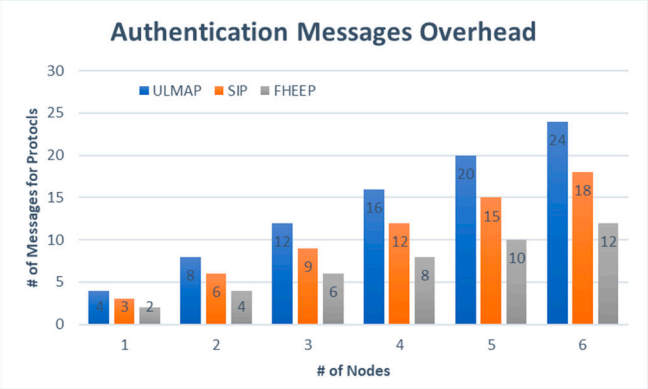

Figure 5. Showing the Messages overhead for no of nodes.

6. Conclusions

We presented a modified algorithm from previous works as a merger of two algorithms, where ECC is used to select the key pair, and then Elgamal is used for exchanging the secret key. Communication then occurs normally with secret key encryption, which is considered faster and the most secure. Moreover, fewer messages form the main key will reduce the probability of guess attacks. ECC is proven to be light; as shown in the results, the 512-bit key is equal to a 15,360 bit key in RSA, which is quite a lot bigger. We also tested the FHEEP against major well-known attacks and found it to be secure against these attacks mathematically. We also used the second level of hashing to ensure that the integrity of the message is met and that there are no replay attacks, as we are also considering the nonce in authentication messages. The algorithm is recommended for light devices that have low computation power but that possess decision capabilities. This algorithm also addresses a new type of attack called the interleaving attack, which uses a Third party or Certification authority to allow devices to act as relays, and their trust is established by the CA. FHEEP can also handle new types of attacks and will be a computationally less expensive algorithm for D2D communication in 5G. In the future, authors plan to test the algorithms over various setting such as 5G installations and do the security testing of the proposed algorithm. We also plan to do the formal analysis of the proposed algorithm to validate the security of the algorithm.

Author Contributions: Conceptualization, methodology, original draft preparation and formal analysis was done by A.A along with K.A.M; validation, supervision, and funding of the project by Z.D.

Funding: This research work has been carried out in State Key Laboratory Intelligent Communication, Navigation and Micro-Nano System, School of Electronic Engineering, Beijing University of Posts and Telecommunications (BUPT), Beijing, China. The research work received financial supported by National High Technology 863 Program of China (No. 2015AA124103) and National Key R&D program no 2016YFB0502001. The authors are thankful for the financial support and acknowledge guidance and support provided by State Key Laboratory Intelligent Communication, Navigation and Micro-Nano System, BUPT.

Acknowledgments: The authors appreciate and acknowledge anonymous reviewers for their reviews and guidance.

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

References

1. Kato, N. On device-to-device (D2D) communication [Editor's note]. *IEEE Netw.* **2016**, *30*, 2. [CrossRef]
2. Wang, M.; Yan, Z. A Survey on Security in D2D Communications. *Mob. Netw. Appl.* **2017**, *22*, 195–208. [CrossRef]
3. Alkurd, R.; Shubair, R.M.; Abualhaol, I. Survey on device-to-device communications: Challenges and design issues. In Proceedings of the IEEE 12th International New Circuits and Systems Conference (NEWCAS), Trois-Rivieres, QC, Canada, 22–25 June 2014; pp. 361–364.
4. Asadi, A.; Wang, Q.; Mancuso, V. A survey on device-to-device communication in cellular networks. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1801–1819. [CrossRef]
5. Liu, J.; Kato, N.; Ma, J.; Kadowaki, N. Device-to-Device Communication in LTE-Advanced Networks: A Survey. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 1923–1940. [CrossRef]
6. Theobald, L.M.; Wu, D.; Wang, J.; Hu, R.Q.; Cai, Y.; Zhou, L.; Laya, A.; Wang, K.K.; Widaa, A.A.; Alonso-Zarate, J.; et al. Device-to-device discovery for proximity-based service in LTE-advanced system. *IEEE Commun. Mag.* **2014**, *16*, 60–64. [CrossRef]
7. 3GPP. *Feasibility Study on Remote Management of USIM Application on M2M Equipment*; 3GPP: Sophia Antipolis, France, 2009.
8. Shiu, Y.-S.; Chang, S.-Y.; Wu, H.-C.; Huang, S.C.-H.; Chen, H.-H. Physical Layer Security in Wireless Networks: A Tutorial. *IEEE Wirel. Commun.* **2011**, *18*, 66–74. [CrossRef]
9. Khan, A.S.; Javed, Y.; Abdullah, J.; Nazim, J.M.; Khan, N. Security issues in 5G device to device communication. *Int. J. Comput. Sci. Netw. Secur.* **2017**, *17*, 366–375.
10. Stallings, W. *Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud*; Addison-Wesley Professional: Boston, MA, USA, 2015.
11. Rahman, H.; Azad, S. Elliptic curve cryptography. In *Practical Cryptography: Algorithms and Implementations Using C++*; CRC Press: Boca Raton, FL, USA, 2014; pp. 147–181.
12. Lazrag, H.; Chaibi, H.; Rachid, S.; Rahmani, M.D. An Optimal and Secure Routing Protocol for Wireless Sensor Networks. In Proceedings of the 2018 6th International Conference on Multimedia Computing and Systems (ICMCS), Rabat, Morocco, 10–12 May 2018; p. 13.
13. Van der Meulen, R. Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016. *Gart. Press Release* **2017**, *2016*, 1. [CrossRef]
14. Alenezi, M.; Almustafa, K.; Hussein, M. On Virtualization and Security-Awareness Performance Analysis in 5G Cellular Networks. *J. Eng. Sci. Technol. Rev.* **2018**, *11*, 199–207. [CrossRef]
15. Al-Turjman, F. Information-centric framework for the Internet of Things (IoT): Traffic modeling & optimization. *Future Gener. Comput. Syst.* **2018**, *80*, 63–75. [CrossRef]
16. Farris, I.; Orsino, A.; Militano, L.; Iera, A.; Araniti, G. Federated IoT services leveraging 5G technologies at the edge. *Ad Hoc Networks* **2018**, *68*, 58–69. [CrossRef]
17. Habiba, U.; Hossain, E. Auction Mechanisms for Virtualization in 5G Cellular Networks: Basics, Trends, and Open Challenges. *IEEE Commun. Surv. Tutor.* **2018**. [CrossRef]
18. Ahmad, I.; Kumar, T.; Liyanage, M.; Okwuibe, J.; Ylianttila, M.; Gurtov, A. Overview of 5G Security Challenges and Solutions. *IEEE Commun. Stand. Mag.* **2018**, *2*, 36–43. [CrossRef]
19. Kim, J.Y.; Hu, W.; Shafagh, H.; Jha, S. SEDA: Secure Over-The-Air CodeDissemination Protocol for the Internet of Things. *IEEE Trans. Dependable Secur. Comput.* **2016**, *15*, 1041–1054. [CrossRef]

20. Laiphrakpam, D.S.; Khumanthem, M.S. A robust image encryption scheme based on chaotic system and elliptic curve over finite field. *Multimed. Tools Appl.* **2017**, 1–24. [CrossRef]
21. Ni, J.; Lin, X.; Shen, X. Efficient and Secure Service-oriented Authentication Supporting Network Slicing for 5G-enabled IoT. *IEEE J. Sel. Areas Commun.* **2018**. [CrossRef]
22. Sharma, V.; You, I.; Leu, F.Y.; Atiquzzaman, M. Secure and efficient protocol for fast handover in 5G mobile Xhaul networks. *J. Netw. Comput. Appl.* **2018**, *102*, 38–57. [CrossRef]
23. Karati, A.; Islam, S.H.; Karuppiah, M. Provably Secure and Lightweight Certificateless Signature Scheme for IIoT Environments. *IEEE Trans. Ind. Inf.* **2018**. [CrossRef]
24. Gritti, C.; Molva, R.; Önen, M. Lightweight Secure Bootstrap and Message Attestation in the Internet of Things. In Proceedings of the SAC 2018, 33rd ACM/SIGAPP Symposium On Applied Computing, Pau, France, 9–13 April 2018.
25. Fang, D.; Qian, Y.; Hu, R.Q. Security for 5G Mobile Wireless Networks. *IEEE Access* **2017**, *PP*, 1. [CrossRef]
26. Wu, Y.; Chen, B.; Weng, J.; Zhao, Z.; Cheng, Y. Attack and Countermeasure on Interlock-based Device Pairing Schemes. *IEEE Trans. Inf. Forensics Secur.* **2017**. [CrossRef]
27. Sedidi, R.; Kumar, A. Key exchange protocols for secure Device-to-Device (D2D) communication in 5G. In Proceedings of the 2016 Wireless Days (WD), Toulouse, France, 23–25 March 2016. [CrossRef]
28. Ahmed, T. Securing AOMDV Protocol in Mobile Adhoc Network with Elliptic Curve Cryptography. In Proceedings of the 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox's Bazar, Bangladesh, 16–18 February 2017; pp. 539–543.
29. Dake, S.S.; Ighare, R.U. A proposed ECC algorithm for smart cards cell phones and wireless networks. In Proceedings of the 2017 International Conference on Nascent Technologies in Engineering (ICNTE), Navi Mumbai, India, 27–28 January 2017.
30. Khan, S.; Khan, R. Elgamal elliptic curve based secure communication architecture for microgrids. *Energies* **2018**, *11*, 759. [CrossRef]
31. Javed, Y.; Khan, A.S.; Abbasi, M.A.K. Key Security Attacks and Their Remedies in D2D Communication. *UBICC* **2019**, 13.
32. Shah, S.T.; Hasan, S.F.; Seet, B.C.; Chong, P.H.J.; Chung, M.Y. Device-to-Device Communications: A Contemporary Survey. *Wirel. Pers. Commun.* **2018**, *98*, 1247–1284. [CrossRef]
33. Kumar, M.; Gupta, P. A Novel and Secure Multiparty Key Exchange Scheme Using Trilinear Pairing Map Based on Elliptic Curve Cryptography. *Adv. Intell. Syst. Comput.* **2018**, *583*, 37–50. [CrossRef]
34. Javed, Y.; Khan, A.S.; Qahar, A.; Abdullah, J. EEoP: A lightweight security scheme over PKI in D2D cellular networks. *J. Telecommun. Electron. Comput. Eng.* **2017**, *9*, 99–105.
35. Bach, E. Toward a theory of Pollard's rho method. *Inf. Comput.* **1991**, *90*, 139–155. [CrossRef]
36. Bos, J.; Kaihara, M.; Kleinjung, T. On the Security of 1024-bit RSA and 160-bit Elliptic Curve Cryptography. *Iacr Cryptol. Eprint* **2009**, *57*, 1–19.
37. Bos, J.W.; Kaihara, M.E.; Kleinjung, T.; Lenstra, A.K.; Montgomery, P.L. Solving a 112-bit prime elliptic curve discrete logarithm problem on game consoles using sloppy reduction. *Int. J. Appl. Cryptogr.* **2012**, *2*, 212. [CrossRef]
38. Meadows, C. Analyzing the Needham-Schroeder public key protocol: A comparison of two approaches. *Comput. Secur.—Esorics 96* **1996**, *1146*, 351–364. [CrossRef]
39. Atiyah, M.F.; Macdonald, I.G. Introduction to commutative algebra. *Am. Math. Mon.* **1969**, *77*, ix+128. [CrossRef]
40. Hankerson, D.; Menezes, A.J.; Vanstone, S. *Guide to Elliptic Curve Cryptography*; Springer: New York, NY, USA, 2004.
41. Vanstone, S.A. Next generation security for wireless: Elliptic curve cryptography. *Comput. Secur.* **2003**, *22*, 412–415. [CrossRef]
42. Wang, Y.; Ramamurthy, B.; Zou, X. The Performance of Elliptic Curve Based Group Diffie-Hellman Protocols for Secure Group Communication over Ad Hoc Networks. In Proceedings of the 2006 IEEE International Conference on Communications, Istanbul, Turkey, 11–15 June 2006; pp. 2243–2248.
43. Stajano, F.; Anderson, R. The Resurrecting Duckling: Security Issues for Ubiquitous Computing. *Computer* **2002**, *35*, supl22–supl26. [CrossRef]

44. Fan, K.; Song, P.; Yang, Y. ULMAP: Ultralightweight NFC Mutual Authentication Protocol with Pseudonyms in the Tag for IoT in 5G. *Mob. Inf. Syst.* **2017**, *2017*. [CrossRef]
45. Chaudhry, S.A.; Naqvi, H.; Sher, M.; Farash, M.S.; Hassan, M.U. An improved and provably secure privacy preserving authentication protocol for SIP. *Peer Peer Netw. Appl.* **2015**, 1–15. [CrossRef]