SAFE: Secure and Fast Key Establishment for Resource Constrained Devices in Device to Device Communications

Mahanya Kochhar[1](✉), Narendra S. Chaudhari[1], and Shubham Gupta[2]

[1] Indian Institute of Technology, Indore 453552, India
mahannya.kochhar@gmail.com
[2] SRM University, Amaravati 522240, Andhra Pradesh, India

Abstract. Device to Device Communications allows devices within certain proximity limits to communicate directly with or without cellular network infrastructure. This leads to faster data exchange and low latency delays. With ever-rising security threats that can jeopardize D2D communications, authentication of devices and initial key establishment for further message encryption is the need of the hour for secure D2D communication. In this paper, we first analyze authentication schemes for traditional Diffie-Hellman and their shortcomings in terms of performance and security for resource-constrained devices. We then propose a solution initially meant for wireless sensor networks for key issuing and establishment, which is ideal for devices with resource limitations. The principle of the protocol is based on Identity-based Key Issuing and a Key Generation Centre Model. It is observed that a secure session key is established between two devices with the above protocol. The proposed protocol eliminates the need for certificates that lead to storage, communication, and computation overheads. It is suitable in terms of computation and communication overhead with the existing literature. The proposed protocol with the proposed Key Generation Centre model can easily be integrated into devices enabled with Wi-Fi Direct further enhancing the security of D2D communications.

Keywords: Device to Device Communications · Wi-Fi Direct · Resource constrained devices · Key establishment

1 Introduction

Device to Device (D2D) Communications has been a major area of research in recent years. Two or more devices within certain proximity can communicate with each other with or without the involvement of existing cellular network infrastructure. With more devices being connected globally and ever-rising mobile subscribers, the need for fast and secure data exchange between devices is an urgent and pressing requirement.

D2D communications allow User-Equipments (UEs) within a certain proximity to communicate using a direct link without routing radio signal paths through the network infrastructure. This promises high data rates and ultra-low latency due to shorter

N. Khare et al. (Eds.): MIND 2022, CCIS 1763, pp. 293–307, 2022.
https://doi.org/10.1007/978-3-031-24367-7_29

signal paths. Till recent times, D2D communications did not appear financially feasible to network operators. However with more demanding resource-consuming applications and proximity-based services, D2D has been seen as a key complementary technology with the 5G infrastructure for peer-to-peer communication, proximity detection services, Machine to Machine Communication, coverage extensions, data and computation offloading, emergency communications and IoT enhancement (e.g., V2V communication).

5G technology promises higher bandwidth capabilities, low data rates and efficiency in the exchange of real-time data and hence is the preferred technology for IoT-based applications. The D2D Communications in 5G are possible via cellular services or Wi-Fi Direct. As cellular services are limited by partial or no coverage, Wi-Fi Direct can efficiently and reliably facilitate data exchange for the implementation of IoT-enabled smart city applications. Figure 1 illustrates the two typical D2D communication scenarios.

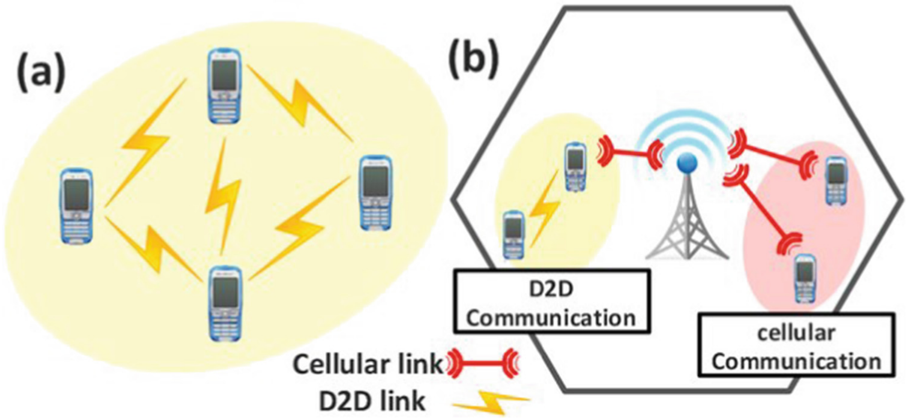

Fig. 1. D2D communications (a) without infrastructure (b) cellular network assisted D2D

With the development of emerging technologies, new security threats arise that can disrupt and jeopardize the whole communication setup. Typical D2D communication security threats include jamming, user emulation attacks, message modification and node impersonation. Man-in-the-middle (MITM) attack is an active eavesdropping attack when an attacker device disrupts the private communication between communicating devices and relays messages between those devices that still believe they are communicating with each other securely.

For secure and reliable exchange of information between nodes with protection from the security threats mentioned, key establishment via cryptographic primitives is required that authenticates the devices before any exchange of information. The security requirements of Confidentiality, Integrity and Authentication (CIA triad) are a must for any secure key establishment protocol along with robustness, privacy, non-repudiation and availability and dependability of the network.

The authors of this work propose SAFE: a secure and fast key establishment for Resource Constrained Devices in D2D communication scenarios with an identity-based key issuing model.

1.1 Technical Contribution

This article proposes a lightweight cryptographic protocol based on research work conducted for key establishment in typical D2D devices. The authors of this work use the idea of an Identity-based Diffie-Hellman Key Establishment with the latest and fastest Elliptic Curve Cryptography primitives discussed subsequently for mutual authentication and key establishment in D2D scenarios. The important contributions of this paper include:

- Analysis of the Station to Station Protocol (STS) as a possible solution to the MITM attack in traditional unauthenticated Diffie-Hellman key exchange.
- We present an underlying security issue in STS and propose a modification to the same to improve its security.
- We find issues with our proposed approach in terms of performance and suitability for resource-constrained devices with costly cryptographic primitives involved.
- To achieve a robust, lightweight authenticated key agreement we propose an identity-based solution initially meant for wireless sensor networks for our D2D communication scenarios.
- We propose a new identity-based key issuing model to the above solution to improve security and robustness for our D2D communication scenarios. We further evaluate SAFE's computation and communication overheads and compare them with existing literature works. We further analyze the latest Elliptic Curve Cryptography (ECC) primitives for fast scalar multiplication for use in our key generation and key establishment phases.

The rest of the paper is organized as follows: Sect. 2 provides a literature overview and discusses conventional and current research work for secure D2D communications. Section 3 describes a new protocol proposed by this work's authors and its performance limitations. Section 4 defines the notations and security goals for secure D2D communication. Section 5 identifies a solution initially for sensor networks and proposes the same for the key issuing and establishment phase. Section 6 discusses the results of SAFE. Section 7 analyzes the latest Elliptic Curve Cryptography primitives for performance improvements. Finally, Sect. 8 draws the conclusions.

2 Background

D2D bypasses the cellular network infrastructure or base stations enhancing spectral efficiency and reducing latency. Such enhancements to the existing infrastructure speed up the data exchange between devices. For the adoption and deployment of D2D services, security and privacy are fundamental aspects to be addressed [1].

Wi-Fi Direct has emerged as a suitable technology for D2D communications to save data exchange and communication costs. D2D communication establishment via Wi-Fi Direct is a four-stage process: discovery, GO negotiation, WPS and address configuration. D2D communications with Wi-Fi Direct are susceptible to various security threats and challenges [2]. Wi-Fi Direct relies on Wi-Fi Protected Setup (WPS) to connect two devices securely. The limitations of this setup in terms of security enable an attacker to perform a brute force attack against the WPS Pin solution [3]. Short Authentication Scheme (SAS) based Protocols [4] by S Pasini et al. require a safe and secure Out of Band (OOB) channel for string authentication and in reality, no channel can be considered to be secure in wireless communications. Hence such protocols are subject to eavesdropping and MITM which leads to a compromise in the security of the system.

Diffie-Hellman Key Exchange is based on the computational hardness of the Discrete Logarithm problem [5]. Traditionally, Diffie-Hellman Key Exchange is unauthenticated and subject to the famous MITM attack [6]. Whitfield Diffie and others introduced a protocol referred to as the Station to Station (STS) Protocol for authenticated Diffie-Hellman key exchange [7]. The STS protocol consists of traditional Diffie-Hellman key establishment along with an exchange of authentication signatures with the help of certificates issued via a trusted authority. In practice, the STS Protocol uses certificates to facilitate the distribution of users' public keys and user-specific Diffie-Hellman parameters.

Since these traditional authentication mechanisms are either costly in terms of processing speed, and resource utilization or are vulnerable to attacks, it is evident that a strong, lightweight mutual authentication scheme is required in the Wi-Fi Direct Protocol to enhance the security of D2D communications.

The authors of [8] propose a solution for secure D2D communication with Elliptic Curve Cryptography (ECC) and the lightweight AEAD cipher for efficiency. Maode Ma et al. [9] proposed an LTE-AKA scheme based on [8] for 5G D2D networks. However, both the above solutions are designed for 5G D2D networks, require the necessary 5G infrastructure, and are not suitable for Wi-Fi Direct D2D scenarios. Based on the computational hardness of the ECC Discrete Logarithm Problem, the authors of [10] propose an authenticated certificateless key agreement protocol that uses International Mobile Subscriber Identity (IMSI) as identity information.

To improve the security of D2D communications via Wi-Fi Direct, the authors of [11] proposed an authentication approach called Secure Key Exchange with QR code (SeKeQ) to enable devices to establish a shared key over public channels. Besides having large computation and communication overheads, SeKeQ requires the D2D device to scan a QR code for string authentication purposes. We typically want to avoid any OOB authentication in an authentication protocol for security reasons.

We now discuss and analyze the MAKE scheme [12] that aims to enhance the security capabilities of the Wi-Fi Direct Protocol.

2.1 Intelligent Device Filtering and Mutual Authentication and Key Establishment (MAKE)

The authors of [12] propose an intelligent device filtering mechanism and mutual authentication and key establishment scheme for preventing DOS attacks in the discovery phase and MITM attacks in the key agreement phase of the Wi-Fi Direct Protocol.

It is assumed by the authors of [12] that nodes are stationary and have unique MAC Addresses and Received Signal Strength Indicator (RSSI) values. These addresses are used to verify the legitimacy of the probe requests received at a node for further communication. With such a probe request, the authors of [12] aim to prevent bogus requests received at legitimate devices that drain battery resources and processing capabilities.

The mutual authentication and key establishment scheme (MAKE) aims to enhance the security capabilities of the SAS-based key agreement [4] with timestamps added and message authentication codes for mutual authentication and secret key establishment. The authors aim to eliminate the use of OOB channels and plaintext communication of short authentication strings.

3 Cryptanalysis of Protocols

3.1 Cryptanalysis of Intelligent Device Filtering and MAKE Scheme

While the protocol proposed by [12] enhances the security of the Wi-Fi Direct Protocol and protects devices from DOS and MITM attacks, it has some severe limitations in terms of strict assumptions.

The assumption by the authors that nodes are stationary is not feasible for D2D communications as mobile devices are at the heart of D2D communications and their RSSI values cannot be held constant ever as RSSI itself is determined by weather conditions, temperature and obstacles in the path of communication [13]. Hence the assumption that the RSSI of devices is held constant is impractical in real-life applications. Thus, the intelligent device filtering for malicious nodes to prevent DOS will fail in real-time scenarios.

The MAKE scheme employs the use of timestamps and lifetime values for authentication purposes. Timestamps put forward a huge challenge of maintaining local clocks that are periodically synchronized securely with reliable sources of time that lead to delays and tradeoffs in performance and security and are hence not recommended by the authors of [7] for any use in authentication protocols.

Cryptanalysis of STS Protocol

We have critically analyzed the STS protocol and found a vulnerability in the same. The attacker can compromise any communication between two devices. Such an attack is illustrated in Fig. 2.

1. If a powerful adversary can issue certificates in someone else's name through trusted authority or compromise the trusted authority, a successful MITM attack can be established where an adversary can impersonate one of the devices, making it appear as if a normal exchange of information is underway.
2. Let's assume a D2D communication scenario where one device (say Device$_1$) wants to communicate with another D2D device (Device$_2$) but somehow attacker device (Device$_3$) intervenes. Device$_3$ pretends to be Device$_2$ to Device$_1$ and receives the STS protocol's message (1). Device$_3$ cannot determine x (secret key of Device$_1$), from α^x. However, Device$_3$ sends α, p and $\alpha^{x'}$ to Device$_2$ where x' is Device$_3$'s

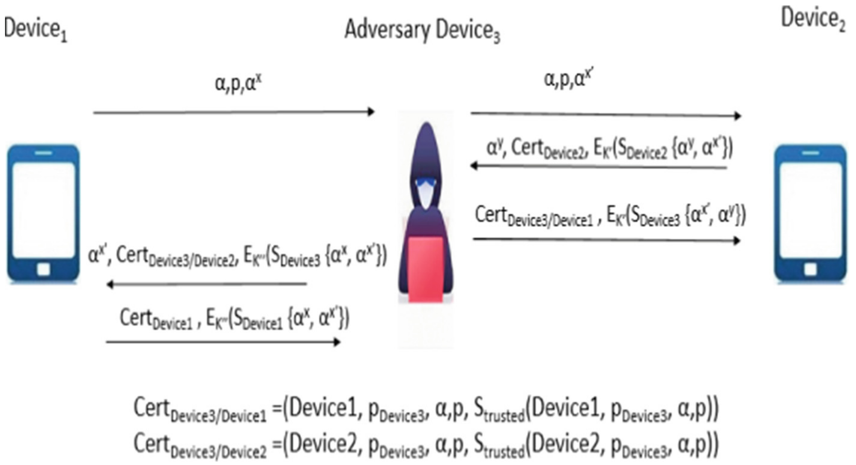
Device1
Adversary Device3
Device2
α,p,α^x
α,p,α^x'
α^y, Cert_Device2, E_K'(S_Device2 {α^y, α^x'})
Cert_Device3/Device1 , E_K'(S_Device3 {α^x', α^y})
α^x', Cert_Device3/Device2, E_K''(S_Device3 {α^x', α^x})
Cert_Device1 , E_K''(S_Device1 {α^x, α^x'})
Cert_Device3/Device1 =(Device1, p_Device3, α,p, S_trusted(Device1, p_Device3, α,p))
Cert_Device3/Device2 =(Device2, p_Device3, α,p, S_trusted(Device2, p_Device3, α,p))

Fig. 2. MITM in STS with certificate forgery by an adversary

secret key. Device$_2$ thinks he has received this message from Device$_1$ and sends the message (2) of the STS Protocol as α^y, Cert$_{\text{Device2}}$, $E_{K'}$ (S$_{\text{Device2}}$ $\{\alpha^y, \alpha^{x'}\}$) where K′ is the shared key between Device$_2$ and Device$_{3.}$

Device$_3$ now sends $E_{K'}$ (S$_{\text{Device3}}$ $\{\alpha^{x'}, \alpha^y\}$) to Device$_2$ along with a forged certificate (Cert$_{\text{Device3/Device1}}$) issued by him carrying Device$_1$'s name. Device$_2$ thinks he is communicating with Device$_1$, however, it is Device$_3$ who has established a connection with him. Now, Device$_3$ sends to Device$_1$ $\alpha^{x'}$, Cert$_{\text{Device3/Device2}}$, $E_{K''}$ (S$_{\text{Device3}}$ $\{\alpha^{x'}, \alpha^x\}$) where Cert$_{\text{Device3/Device2}}$ is the forged certificate carrying Device$_2$'s name and K″ is the shared secret key between Device$_1$ and Device$_3$. Device$_1$ considers this message to be Device$_2$'s and responds with $E_{K''}$ (S$_{\text{Device1}}$ $\{\alpha^x, \alpha^{x'}\}$) along with his original certificate.

Thus Device$_3$ has established independent connections with Device$_1$ and Device$_2$. As Device$_1$ and Device$_2$ are unaware of fraudulent Device$_3$, they continue the communication with Device$_3$.

Proposed Hybrid Protocol to Solve the Issue with STS

The hybrid protocol proposed by the authors of this work assumes a Public Key Infrastructure (PKI) set-up where the sender of a message can easily access the recipient's public key and use it to encrypt messages and send messages.

Encrypting the sender's exponential term and Diffie-Hellman parameters with the sender's private key first and the recipient's public key accessed via broadcast or PKI (see Fig. 3 and Table 1) mitigates the attack mentioned in the previous section. Such a modification ensures that the sender's exponential term is viewed by the intended recipient as only the intended recipient can decrypt the message with his own private key first followed by the sender's public key.

Thus any adversary can never find out the sender's exponential as he does not have the recipient's private key which is confidential information. Hence, he cannot establish independent connections with the intended targets, so the MITM attack is prevented.

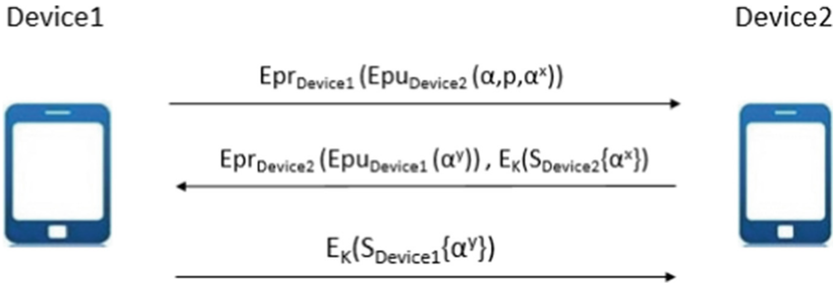
Device1
Device2
$Epr_{Device1}(Epu_{Device2}(\alpha, p, \alpha^{x}))$
$Epr_{Device2}(Epu_{Device1}(\alpha^{y}))$, $E_K(S_{Device2}\{\alpha^{x}\})$
$E_K(S_{Device1}\{\alpha^{y}\})$

Fig. 3. Proposed hybrid scheme with encryption

Table 1. Notations and denotations

Notations	Denotations
$pr_{Device1}$	Public key of device 1 Private key of device 1
$pu_{Device1}$	
$pr_{Device2}$	Private key of device 2
$pu_{Device2}$	Public key of device 2

The proposed protocol is secure from MITM attack but it has certain limitations described as follows:

1. Assumption that the sender initially knows the recipient's public key is not always practical without a large infrastructure set up.
2. Resource-constrained devices like mobiles, sensors and IoT devices have low processing power and battery limitations.
3. Standard Encryption algorithms like Blowfish, and AES have high processing capability requirements and are not suitable for D2D devices that are resource constrained.
4. Cryptographic certificates for authentication require additional computational effort for their authentication purposes.

Due to the resource limitations of typical D2D devices, identity-based schemes that embed the authentication in the key establishment procedure are considered ideal for D2D devices as computational overhead is minimized at the device participating in D2D.

4 Notations and Typical Security Goals

4.1 Notations of Proposed Solution

The notations and denotations of the parameters used in the proposed solution are found in Table 2 below.

Table 2. Notations and denotations for proposed solution

Notations	Denotations
H	Cryptographic hash function
E	Elliptic curve
G	Generator point on curve
q	Prime order selected
KGC	Key generation centre
d	Master private key of KGC
R	Master public key of KGC
H_{device}	Nonce generated by KGC
U_{device}	Public value of device
x_{device}	Secret key of device
KGC_i	KGC selected by device for session communication
p_{device}	Nonce generated by device
E_{device}	Random point on elliptic curve
ID_{device}	Identification parameter of device

4.2 Security Goals

The security goals of a secure authentication protocol include:

1. Secret Key Establishment: D2D devices generate and share a large amount of data that is shared with other devices through untrusted wireless channels. Therefore a secure key via a key agreement and establishment protocol between the two parties is needed for data encryption.
2. Mutual Authentication: Authentication is required to confirm the identity of a device. With the open nature of wireless channels, D2D devices need to be authenticated to ensure that the correct parties are communicating with each other. Authenticated Key exchanges are required that also authenticate the identities of parties involved in the key exchange.

   Diffie Hellman Key Exchange is unauthenticated and hence it needs a separate authentication scheme to authenticate devices.
3. Ephemeral Key Exchange: Static/Fixed keys remain the same over a long period. One key for many instances of a key establishment scheme is not good practice.

   An Ephemeral key is generated for each execution of a key establishment process. It ensures key freshness and a unique key for each session.
4. Security and Defense from Prominent Attacks:

Security protocols must be resilient against common attacks such as node impersonation, the MITM attack and authentication attacks.

The proposed solution in the subsequent section establishes a unique ephemeral key comprising of a fixed identity-based component and a random component. The Key Establishment phase achieves perfect forward secrecy and is secure and resilient against the MITM attack and other attacks as shown in [18].

5 Proposed Solution

At the end of Sect. 3, identity-based schemes were discussed that embed authentication into the key establishment and saved computational effort. ID-based encryption was first proposed by Adi Shamir in 1984 [14].

The pairing-based Boneh-Franklin scheme solved the IBE problem with pairings [15]. However, it made use of expensive bilinear maps. Two pairing-free identity-based schemes that require minimum computational effort were further introduced. One approach was introduced by Arazi, Qi et al. in 2007 [16]. Another approach was introduced by D Fiore and Rosario Gennaro in 2011 [17].

Based on Arazi, Qi solution; the protocol proposed by [18] has been considered suitable by the authors of this paper for the identity-based Diffie-Hellman Key Agreement. We now introduce Hang et al. solution for wireless sensor networks as a possible key establishment solution for D2D devices. We further propose a change to the Key Generation Centre model proposed by [18] later in this section.

The protocol introduced by [18] based on Arazi, Qi scheme consists of 2 steps: Identity-based Key issuing and Key Establishment.

5.1 Identity-Based Key Issuing

The identity-based key agreement schemes avoid the use of public certificates by making the public key computable easily from some unique identification information of the owner. The identification information can include a unique identification number or device properties. Identity-based cryptography thus avoids cumbersome certificate management infrastructure and saves computational effort.

Key Issuing Model

In this step, each device is presented with an identity, a secret key and a public value. This is an identity-based approach. For Elliptic Curve Key Establishment, a suitable safe elliptic curve E over a finite field along with an initial generator point G of prime order q is chosen by the Key Generation Center.

A cryptographic hash function $H: \{0, 1\}^* \rightarrow \{1, \ldots, q - 1\}$ like HMAC is further needed. The authors of [18] propose a single Key Generation Centre (KGC) for endowing all D2D participants with a secret key and public value. The KGC generates a random number $d \in \{1, \ldots, q - 1\}$ as the master private key of itself and computes its own master public key R as $R = d \times G$. All D2D participants are aware of the elliptic curve, the point G selected, R and the prime order q.

Let us consider two devices A and B want to participate in D2D communication. Before the D2D device is deployed, it is presented with a secret key and public value by the KGC setup. Firstly, the KGC generates a random number $h_A \in \{1, \ldots, q - 1\}$ and calculates $U_A = h_A \times G$. U_A is the public value of device A presented to it by KGC.

Then, the private key $x_A \in \{1, \ldots, q-1\}$ of device A is generated by the KGC as follows $x_A = [H(ID_A, U_A) \cdot h_A + d] \bmod q$.

This public value U_A and secret key x_A are then issued to device A. This key issuing is done for every device interested in D2D communications. On receiving U_A and secret key x_A, a device can verify its issued values by checking whether

$$x_A \times G = H(ID_A, U_A) \times U_A + R \tag{1}$$

The value $x_A \times G$ is the public key of Device A. It is never used explicitly, however, it is computed from the identification parameters of a node and Key Generation Centre public information.

A single KGC was originally proposed. However, a single KGC's failure can disrupt the entire communication setup. The authors of this work propose multiple KGCs, say n KGCs, where each KGC can have an independent curve E, the point G, and master public key R. The prime order q is assumed same for all KGCs. Each D2D participant is aware of the KGC parameters when it receives n distinct (X_A, U_A) pairs and can verify each pair independently.

Such independent KGCs ensure a fault-tolerant key generation system where the compromise of a single KGC facility doesn't lead to a breakdown of the infrastructure.

Figure 4 illustrates the proposed key issuing model with n KGCs.

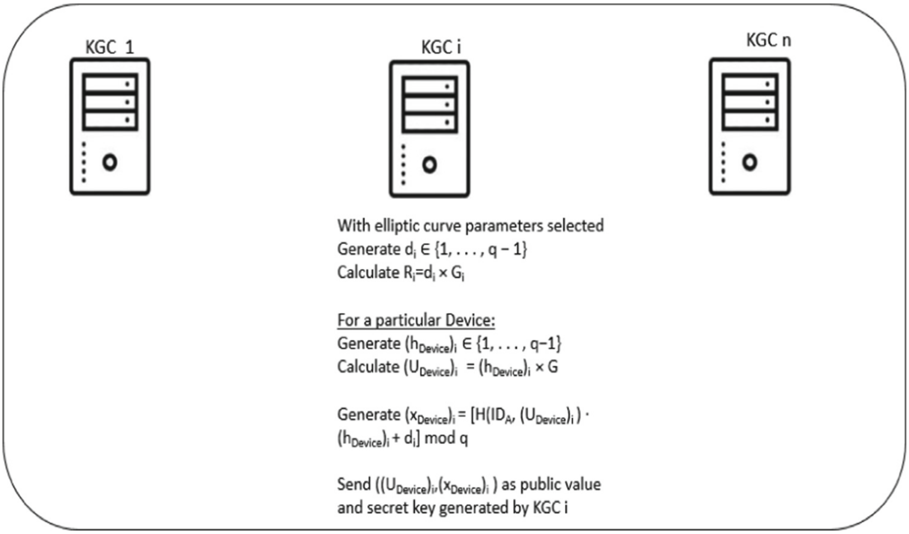
KGC 1
KGC i
KGC n
With elliptic curve parameters selected
Generate $d_i \in \{1, \ldots, q-1\}$
Calculate $R_i = d_i \times G_i$
For a particular Device:
Generate $(h_{Device})_i \in \{1, \ldots, q-1\}$
Calculate $(U_{Device})_i = (h_{Device})_i \times G$
Generate $(x_{Device})_i = [H(ID_A, (U_{Device})_i) \cdot (h_{Device})_i + d_i] \bmod q$
Send $((U_{Device})_i, (x_{Device})_i)$ as public value and secret key generated by KGC i

Fig. 4. Key issuing model with n KGCs

Key Establishment Procedure

For two D2D participants to communicate with each other, they should have their identity-based keys issued at first as described by the key issuing model. For key establishment between devices A and B, the protocol is as follows:

Each node generates independent numbers as follows: $p_A \in \{1, \ldots, q-1\}$ is generated by device A and a nonce $p_B \in \{1, \ldots, q-1\}$ is generated by device B. A indicates to B its desire for D2D communication and shares its (ID_A, U_A, KGC_i, $E_A = p_A \times G$) where KGC_i is the KGC chosen by Device A for this particular D2D communication instance. Sending this parameter is important so that the receiving node can then select the correct secret key, public key-value pair, master public key of KGC, elliptic curve considered and initial point G from all possible pairs available to him.

Now Device B processes the message (1) and responds with (ID_B, U_B, KGC_i, $E_B = p_B \times G$). Now both the Devices have agreed to use the (secret key, public value) pairs issued by KGC_i and public information of KGC_i for key establishment.

Device A checks whether $E_B \neq 0$, $E_B \in E$ and now calculates the current session key K_{AB} as

$$\begin{aligned} K_{AB} &= \left[x_A \cdot H(ID_B, U_B) \bmod q\right] \times U_B + x_A \times R + p_A \times E_B \\ &= x_A x_B \times G + p_A p_B \times G. \end{aligned} \tag{2}$$

Similarly, after checking whether $E_A \neq O$ and $E_A \in E$, the session key K_{BA} is also computed by Device B as

$$\begin{aligned} K_{BA} &= \left[x_B \cdot H(ID_A, U_A) \bmod q\right] \times U_A + x_B \times R + p_B \times E_A \\ &= x_B x_A \times G + p_B p_A \times G \end{aligned} \tag{3}$$

The two parties have now generated a common secure session key and can initiate their communication.

Since $x_A \times R$ is fixed for each session, it can be pre-stored in the device multiplication. Also, E_A can be precomputed before key establishment to save a scalar multiplication. Therefore at the key establishment, 2 scalar multiplications are required at each device.

Figure 5 illustrates the key establishment procedure between two devices.

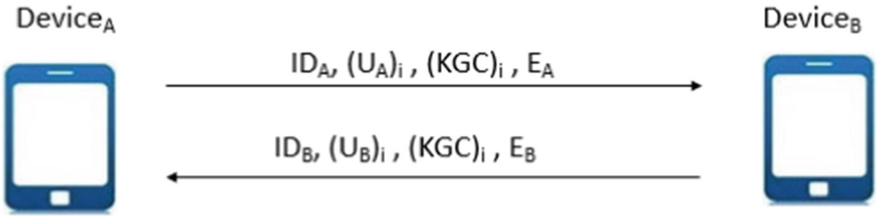
Device$_A$
Device$_B$
ID_A, $(U_A)_i$, $(KGC)_i$, E_A
ID_B, $(U_B)_i$, $(KGC)_i$, E_B

Fig. 5. Key establishment between two devices

6 Results

The authors of this work intend to implement the proposed key generation model and key agreement protocol for D2D devices shortly. For now, the quantitative and qualitative analysis of the proposed scheme is performed in terms of its communication and computation overheads. We further compare our solution with some existing solutions in the literature and prove why our solution is ideal for D2D communications with Wi-Fi Direct.

The key issuing procedure is performed at the Key Generation Centres (KGCs). It is a reasonable assumption that Key Generation Centres have enough processing capabilities and resources to generate and issue keys that embed user identity.

Key Establishment between two devices involves only two message exchanges that have minimum communication overhead. ECC Scalar multiplications are typically the most costly operations in ECC and our proposed scheme employs 2 Scalar multiplications at each stage. This is a reasonable amount of computation overhead at each device as storage overhead, verification, communication and computation overhead of certificates are eliminated with such a protocol.

Table 3 illustrates the communication overhead of some common cryptography primitives.

Table 3. Communication overhead specifications of some cryptography primitives

Parameters	No of bits required
$\mathrm{ID}_{\mathrm{device}}$	128 bits
$\mathrm{U}_{\mathrm{device}}$	256 bits
$\mathrm{E}_{\mathrm{device}}$	256 bits
$\mathrm{N}_{\mathrm{device}}$	128 bits
g^{a}	1024 bits
Ts	64 bits
L_{T}	64 bits
HMAC	160 bits

The communication overhead for key establishment at each device in our protocol amounts to overheads of $\mathrm{ID}_{\mathrm{device}} + \mathrm{U}_{\mathrm{device}} + \mathrm{E}_{\mathrm{device}} + \mathrm{KGC}_{\mathrm{i}}$. This amounts to a net overhead of (128 + 256 + 256 + log(n) bits) = 640 + log(n) bits at each device where n is the no of KGCs in the KGC and is implementation-dependent (depends on number of KGCs selected in the KGC model).

Assuming elapsed time for scalar multiplications to be T_{SM}, elapsed time for modular exponentiation to be T_{ME} and elapsed time for hash message function to be $\mathrm{T}_{\mathrm{Hash}}$. It is also considered that elapsed time for XOR, $\mathrm{T}_{\mathrm{XOR}}$ is negligible. From the research work in [19] where the elapsed time of certain cryptographic primitives is verified by the OpenSSL [20] library written in C++, it is evident that $\mathrm{T}_{\mathrm{ME}} > \mathrm{T}_{\mathrm{SM}} > \mathrm{T}_{\mathrm{Hash}}$. The computation overhead for key establishment at each device in our proposed scheme is thus $2 * \mathrm{T}_{\mathrm{SM}} + \mathrm{T}_{\mathrm{Hash}}$.

We now compare our proposed scheme with the Key Establishment phase in SeKeQ [11] and MAKE [12] for communication and computation overheads in Table 4.

Table 4. Comparative analysis with MAKE at each D2D device

Protocol	Communication overhead	Computation overhead
MAKE	1696 bits	$2 * T_{ME} + 2 * T_{Hash}$
SeKeQ	1184 bits	$2 * T_{ME} + T_{Hash}$
SAFE	640 + n bits	$2 * T_{SM} + T_{Hash}$

It is clear that the protocol SAFE proposed by the authors of this work is efficient in terms of computational and communication overheads and is thus an ideal choice for D2D communication technology with Wi-Fi Direct in resource-constrained devices.

7 Latest Elliptic Curve Cryptography Primitives

Elliptic Curve Cryptography (ECC) is the modern encryption technology as it is smaller, faster and lightweight than traditional cryptography primitives. As the name suggests, ECC uses the properties of algebraic elliptic curves over finite fields and the computational hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP) [21].

Elliptic Curve Diffie Hellman (ECDH), uses ECC point multiplications instead of modular exponentiations. ECDH promises very fast key generation and key agreement with smaller keys. The performance analysis by [22] reveals that to achieve a 128-bit symmetric-equivalent security level, Diffie Hellman needs a key length of 3072 bits whereas ECDH requires just 256 bits. This smaller key length improves overall performance and is thus suitable for our resource-constrained D2D devices.

Now we turn our focus to the latest underlying Elliptic curves used by ECC Algorithms. Different Elliptic Curves provide different results in terms of performance, security and resource consumption. Choosing safe elliptic curves is an important criterion to provide ECC security and avoid side-channel attacks. Notable cryptographers like Bernstein believe that most of the curves described in the NIST crypto standards are unsafe and have suspicious origins and hence have defined their own ECC security standards. Safe curves for use in ECC as studied by Bernstein and Lange are documented in [23]. Bernstein in 2005 released an ECDH key agreement protocol, X25519 with Curve25519 as the underlying elliptic curve [24]. It offers significant performance improvements compared to the NIST elliptic curves and its reference implementation is available publicly, thereby it has gained tremendous popularity recently with messaging application giants like WhatsApp.

The authors of this paper recommend Curve25519 as the underlying elliptic curve used by the KGCs for key issuing procedures.

8 Conclusions

D2D communications with Wi-Fi Direct are fast emerging as a popular mode of communication for the exchange of information between devices. Due to unsecure wireless channels and developing infrastructure for D2D communication, it is typically subject to attacks such as MITM with the traditional protocols that are in use. With the above risks kept in mind, the authors analyzed the famous STS solution to prevent the MITM attack and proposed a new protocol. As devices are resource constrained and have low processing capabilities and the above-mentioned solutions are costly in terms of communication and computation overheads, we proposed a new scheme of key establishment using the concepts of identity-based key issuing and key establishment. The proposed identity-based key issuing scheme eliminates the need for certificates for the authentication of devices with (secret key, public value) pairs now being issued by KGCs that embed device identity. With the above key issuing model and key establishment scheme, we propose using certain latest elliptic curve cryptography primitives for scalar multiplications that make the solution safe and suitable for use in D2D devices with resource limitations. The limitation of this proposed scheme possibly is the one-time setup of the key issuing model and the required infrastructure. The future extension of this work might include detailed performance analysis and implementation of the key issuing and key establishment of the proposed model with the cryptographic primitives above discussed. The authors also plan to explore D2D communication in terms of many-to-one, one-to-many and many-to-many scenarios.

References

1. Haus, M., Waqas, M., Ding, A.Y., Li, Y., Tarkoma, S., Ott, J.: Security and privacy in device-to-device (D2D) communication: a review. IEEE Commun. Surv. Tutor. **19**(2), 1054–1079 (2017). https://doi.org/10.1109/COMST.2017.2649687
2. Wang, M., Yan, Z.: Security in D2D communications: a review. In: Proceedings of the 2015 IEEE Trustcom/BigDataSE/ISPA - Volume 01 (TRUSTCOM 2015), pp. 1199–1204. IEEE Computer Society, New York (2015). https://doi.org/10.1109/Trustcom.2015.505
3. Viehbock, S.: Brute forcing Wi-Fi protected setup, Wi-Fi Prot. Setup 9 (2011)
4. Pasini, S., Vaudenay, S.: SAS-based authenticated key agreement. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 395–409. Springer, Heidelberg (2006). https://doi.org/10.1007/11745853_26
5. Baker, K.A.: Diffie-Hellman Key Exchange. https://www.math.ucla.edu/~baker/40/handouts/rev_DH/node1.html
6. Gretes, M.: MITM Attack. https://open.oregonstate.education/defenddissent/chapter/the-man-in-the-middle/
7. Diffie, W., Van Oorschot, P.C., Wiener, M.J.: Authentication and authenticated key exchanges. Des Codes Crypt **2**, 107–125 (1992)
8. Seok, B., Sicato, J.C.S., Erzhena, T., Xuan, C., Pan, Y., Park, J.H.: Secure D2D communication for 5G IoT network based on lightweight cryptography. Appl. Sci. **10**, 217 (2020)
9. Chow, M.C., Ma, M.: A lightweight traceable D2D authentication and key agreement scheme in 5G cellular networks. Comput. Electr. Eng. **95**, 107375 (2021)
10. Li, S., Li, M., Bao, B., Yu, B., Tang, J.: An efficient authenticated key agreement protocol for D2D communication. In: 2021 7th International Conference on Computer and Communications (ICCC), pp. 199–203 (2021)
11. Belghazi, Z., Benamar, N., Addaim, A., Kerrache, C.A.: Secure WiFi-direct using key exchange for IoT device-to-device communications in a smart environment. Future Internet **11**, 251 (2019)

12. Gaba, G.S., Kumar, G., Kim, T.-H., Monga, H., Kumar, P.: Secure Device-to-Device communications for 5G enabled Internet of Things applications. Comput. Commun. **169**, 114–128 (2021)
13. Munoz, D., Bouchereau, F., Enriquez, R.: Position, Location Techniques and Applications. Elsevier, Amsterdam (2009)
14. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39568-7_5
15. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_13
16. Arazi, B.: Certification of DL/EC keys. In: Proceedings of the IEEE P1363 Study Group for Future Public-Key Cryptography Standards (1999)
17. Fiore, D., Gennaro, R.: Identity-based key exchange protocols without pairings. In: Gavrilova, M.L., Tan, C.J.K., Moreno, E.D. (eds.) Transactions on Computational Science X. LNCS, vol. 6340, pp. 42–77. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17499-5_3
18. Hang, I., Ullmann, M., Wieschebrink, C.: Short paper: a new identity-based DH key-agreement protocol for wireless sensor networks based on the Arazi-Qi scheme. In: Proceedings of the Fourth ACM Conference on Wireless Network security (WiSec 2011), pp. 139–144. Association for Computing Machinery, New York (2011)
19. Gupta, S., Parne, B.L., Chaudhari, N.S.: ISAG: IoT-enabled and Secrecy Aware Group-based handover scheme for e-health services in M2M communication network. Future Gener. Comput. Syst. **125**, 168–187 (2021)
20. OPENSSL-Cryptography and SSl/TLS Toolkit. Technical report. https://www.openssl.org/
21. Rabah, K.: Theory and implementation of elliptic curve cryptography. J. Appl. Sci. **5**(4), 604–633 (2005)
22. Alvarez, R., Caballero-Gil, C., Santonja, J., Zamora, A.: Algorithms for lightweight key exchange. Sensors **17**, 1517 (2017)
23. Bernstein, D.J., Lange, T.: SafeCurves: choosing safe curves for elliptic-curve cryptography. https://safecurves.cr.yp.to. Accessed 5 Aug 2022
24. Bernstein, D.J.: Curve25519: new Diffie-Hellman speed records. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 207–228. Springer, Heidelberg (2006). https://doi.org/10.1007/11745853_14