A New Certificateless System Construction for Multiple Key Generator Centers to Secure Device-to-Device Communications

Othmane Nait Hamoud[1,2][a], Tayeb Kenaza[2][b] and Yacine Challal[1,3][c]
[1]*Ecole Nationale Supérieure d'Informatique, BP 68M, 16309, Oued-Smar, Alger, Algérie*
[2]*Ecole Militaire Polytechnique, BP 17 Bordj Elbahri, Alger, Algérie*
[3]*Sorbonne Universités, Université de Technologie de Compiègne, Heudiasyc UMR CNRS 7253, Compiègne, France*

Keywords: Security, Device-to-Device Communication, Key Management Scheme, Certificateless Public Key Cryptography, Proximity Services, ProSe.

Abstract: Device-to-Device (D2D) communication technology comes as one brick among many others in the construction of the evolving fifth generation system (5G) architecture. The Third Generation Partnership Project (3GPP) standardized D2D communication technology under the Proximity Services (ProSe) proposal. This technology allows enabling direct communication between proximate devices without passing through an infrastructure network. Security of D2D communications must be assured in all scenarios according to whether communication control is ensured by the Evolved Packet System (EPS) or the devices themselves. Certificateless public key cryptography (CL-PKC) is an interesting solution for securing D2D communications. In this paper, we propose a new CL-PKC construction to overcome security issues in all scenarios related to D2D communications and to deal with inherent conflicting security requirements between privacy, anonymity, and *traceability* by the use of multiple Key Generator Centers (KGCs). This was considered particularly as responsibility decentralization between stakeholders to respond the fully mistrust assumption regarding KGCs. Furthermore, the proposed CL-PKC system can give different networks the opportunity to be compatible and to work cooperatively.

1 INTRODUCTION

Device-to-Device (D2D) communication is expected to be one of the main technology components for the next generation of mobile communication networks (5G). Proximity Services (ProSe) is the standardization of D2D technology. It was introduced for the first time in Release 12 of the 3GPP specifications (3GPP, 2013) to enhance the capacity and performance of traditional cellular networks. ProSe allows LTE-A devices to discover each other and to communicate directly and relies on multiple enhancements to existing LTE-A standards including new air interface and new functional elements (3GPP, 2014). Depending on the degree of implication of a Cellular Network Operator (CNO) in D2D communications, three typical scenarios and use cases were proposed by 3GPP in (3GPP, 2013), which we illustrate in Figure 1.

Security of D2D communications is a major challenge since it concerns the security of both radio access interface, network infrastructure, devices, and applications. In the last decade, many solutions have been proposed in the literature to handle security issues in this new technology (Nait Hamoud et al., 2018a; Haus et al., 2017; Wang and Yan, 2015). Unfortunately, these solutions consider the three typical scenarios separately and depend on a Trust Third Party (TTP) which could raise privacy issues in case of breach of trust. According to our vision, ProSe application server, which provides authentic content to ProSe-enabled User Equipment (UE) such as YouTube's content or a local social network, would be owned by the CNO or another third party.

In this paper, we adopt the case where the ProSe Application server belongs to another operator, called D2D Server Provider (D2D-SP). This has an advantage in terms of security, especially regarding privacy issues which gain more attention in the literature (Haus et al., 2017; Ferrag et al., 2017; Hsu et al., 2018) due to users' awareness of their sensitive in-

[a] https://orcid.org/0000-0001-6078-6939
[b] https://orcid.org/0000-0002-4240-2978
[c] https://orcid.org/0000-0002-9237-6210

Hamoud, O., Kenaza, T. and Challal, Y.
A New Certificateless System Construction for Multiple Key Generator Centers to Secure Device-to-Device Communications.
DOI: 10.5220/0007841500840095
In *Proceedings of the 16th International Joint Conference on e-Business and Telecommunications (ICETE 2019)*, pages 84-95
ISBN: 978-989-758-378-0

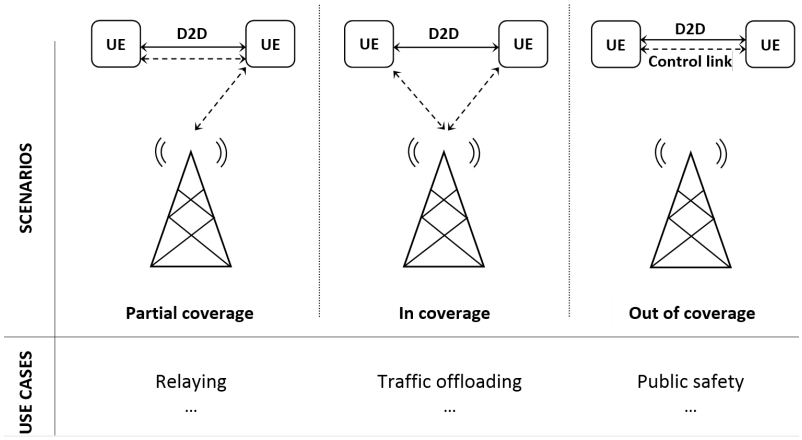


Figure 1: Typical scenarios and use-cases in D2D communications.

formation, especially in the context of globalization. In other words, separating security responsibility between two entities with opposing interests affords us a solution to inherent conflicting security requirements between privacy, anonymity, and *traceability* which are being a novel research area in the last years (Sun et al., 2011; Paja et al., 2013; Alkubaisy, 2017), and an opportunity to find out other business models and pricing issues resolution in D2D communications (Tehrani et al., 2014).

Certificateless public key cryptography (CL-PKC) has been introduced in (Al-Riyami and Paterson, 2003) as an intermediate public key system between Public Key Infrastructure (PKI) and Identity based Public Key Cryptography (ID-PKC) (Shamir, 1984). Indeed, CL-PKC dispenses with the use of certificates and does not suffer from the key escrow problem. In the context of D2D communications, CL-PKC is an interesting solution to face the majority of security issues in the above typical scenarios.

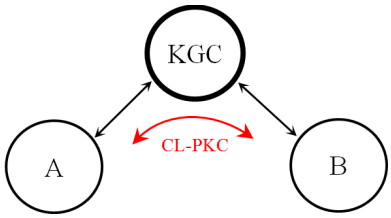
Recently, a new certificateless Generalized Signcryption (CLGSC) scheme was proposed in (Zhang et al., 2017) to secure a multi-hop data transmission protocol in the context of Mobile-Health system. A few months later, Zhou pointed out in (Zhou, 2018) that CLGSC scheme is not secure, particularly in terms of confidentiality. Consequently, he demonstrated that CLGSC authors' Robust Security-Aware D2D-assist data transmission protocol for Mobile-Health systems is also insecure. Zhao *et al.* proposed in (Zhao et al., 2017) a Trustworthy Device Pairing to secure D2D-enabled Mobile Crowdsourcing Systems through D2D communications. The proposed scheme is based on a CL-PKC framework to generate collaboratively by the Backend Server (BS) and registering devices, a pair of a private-public key to each device. However, their TDP does not consider the replacement of devices' public keys by BS especially as it is considered curious. Furthermore, the device's trustworthiness can be adjusted by only the BS after each transaction according to the device behavior. Thus, the proposed solution does not consider the other scenarios. Authors in (Li et al., 2013) proposed a certificateless authentication key agreement (CL-AKA) protocol to secure Session Initiation Protocol with different KGCs. However, there is lack of public key verification mechanism in their solution.

Through this work, we aim to introduce the CL-PKC in ProSe environment in order to decentralize authentication procedures and to face inherent security requirements' conflicts, in the sense that neither the CNO would be able to profile a ProSe-enabled UE according to its D2D application preferences nor the D2D-SP should know the ProSe-enabled UE real identity. Hence, the proposal of a new CL-AKA protocol that no longer requires CNO's coverage. Thus, our contribution through this paper is summarized as follows. We propose a new construction of a CL-PKC system in the case of multiple KGCs and make concrete this construction by proposing new certifiacteless public key encryption (CL-PKE), signature (CL-PKS), and CL-AKA schemes applied in a ProSe environment. We show that these schemes are secure against a stronger adversary. It should be noted that the main idea of our approach was presented in our previous work (Nait Hamoud et al., 2018b). However, more details and rigorous security analysis are given in this paper.

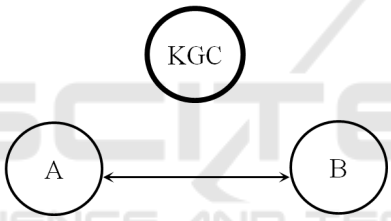
The remainder of this paper is organized as follows. Section 2 summarizes the original model of CL-PKC system (Al-Riyami and Paterson, 2003). In Section 3, we introduce our system and security models. The proposed new CL-PKC system construction and the related schemes are introduced in Section 4. In section 5 we give security analysis of our schemes. Finally, a conclusion is presented in Section 6.

2 CL-PKC: BACKGROUND

Figure 2 shows roughly the main idea of CL-PKC paradigm (Al-Riyami and Paterson, 2003). Initially, a CL-PKC system construction is realized through initialization, registration, and keys setup steps between three participants: a KGC as a TTP and two communicating parties A and B (Figure 2a), and thereafter between only the two communicating parties opportunistically encountered for authenticated key agreement and secure communications, no matter whether the KGC is present or not (Figure 2b).



(a) *Initialization, registration and keys setup*



(b) *CL-AKA and secure communications (encrypt-decrypt and/or sign-verify)*

Figure 2: Main idea of a CL-PKC system.

To make concrete their new paradigm, CL-PKC authors introduced four schemes: CL-PKE, CL-PKS, CL-AKA, and Hierarchical CL-PKE (HCL-PKE) schemes. All these schemes are specified by five common algorithms: *Setup*, *Partial-Private-Key-Extract*, *Set-Secret-Value*, *Set-Private-Key*, *Set-Public-Key*, and additional algorithms: *Encrypt* and *Decrypt* algorithms in CL-PKE scheme, *Sign* and *Verify* algorithms in CL-PKS scheme. Note that in (Al-Riyami and Paterson, 2003), authors focus on CL-PKE showing that a concrete pairing-based CL-PKE scheme is secure provided that an underlying problem closely related to the Bilinear Diffie-Hellman Problem is hard. In the following, we describe briefly the basic scheme of CL-PKE as shown in Figure 3:

1. *Setup*: performed by the KGC. It takes as input a security parameter k and returns the system public parameters *param*, the system's master public key P_0 and the system's master private key s.
2. *Partial private key extract*: for a user A with its identity ID_A, the KGC takes *params*, s and ID_A as inputs and returns to A, over a confidential and authentic channel, a partial private key D_A.
3. *Set secret value*: a user A takes *params*, ID_A and a random x_A and outputs its secret value x_A.
4. *Set private key*: a user A takes as inputs its partial private key D_A, its secret value x_A and *params* and outputs its full private key S_A.
5. *Set public key*: a user A takes as inputs its secret value x_A and *params* and outputs its public key P_A.
6. *Encrypt*: a user B, intending to transmit an encrypted message to a user A, takes as inputs *params*, a message M, A's public key P_A and identity ID_A and outputs a cipher text C.
7. *Decrypt*: a user A, receiving an encrypted message C, takes as inputs *params*, C and its private key S_A and outputs the message M.

3 SYSTEM AND SECURITY MODELS

In this section we describe the system model with its components and the corresponding functions. We describe also the security model based on the attacker model.

3.1 System Model

We consider the 3GPP system model which proposed ProSe as an underlay D2D communications network of existing LTE-A networks (3GPP, 2014). 3GPP integrated three specific entities: ProSe Function, ProSe Application Server and ProSe Application. Figure 4 shows a simplified network architecture for the ProSe system. But in this system model, we consider that ProSe function belongs to the CNO while the ProSe Application Server belongs to a D2D-SP. In the following, we briefly describe these elements.

- ProSe Application is run on ProSe-enabled UE.
- ProSe Function is a logical entity located inside the Evolved Packet Core (EPC) that belongs to the CNO. It may provide connections between ProSe application servers and ProSe-enabled UEs and acts as a KGC in our new CL-PKC system, and supports functionalities related to ProSe Application server and ProSe-enabled UE Identities (storage, verification, charging, etc.).

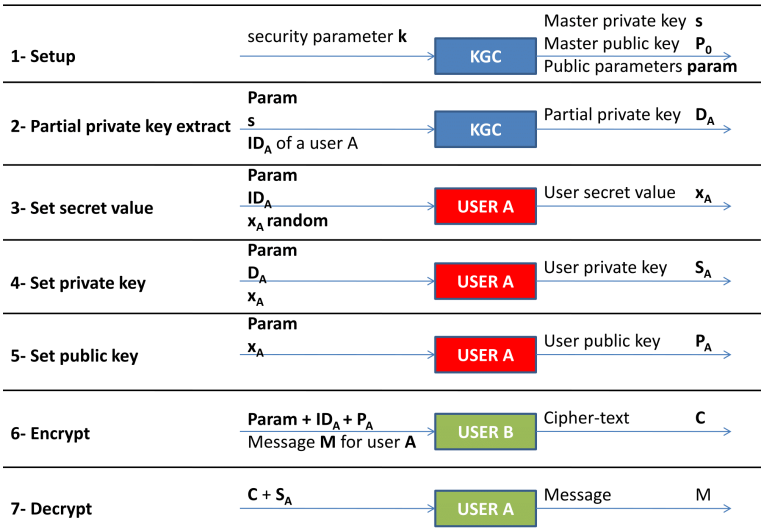


Figure 3: Basic scheme of CL-PKE.

- ProSe Application Server is located outside the EPC and it serves ProSe-enabled UEs requesting ProSe services. It belongs to the D2D-SP, which provides authentic content to ProSe-enabled UEs such as YouTube's content or a local social network. This content will be shared between other ProSe-enabled UEs in order to offload cellular network. The ProSe Application Server acts also as a KGC in our new CL-PKC system and supports also functionalities related to ProSe Function and ProSe-enabled UE Identities (storage, verification, charging, etc.).

In addition to the existing model, we have considered a new entity, that we called Regulatory Authority (RA) and will detail in Section 4.3. RA supplies the different KGCs with the appropriate CL-PKC system parameters without a *system_wide_master_key*. This is in order to overcome the *malicious_but_passive_KGC* problem which we will detail in Section 3.3.

3.2 Security Requirements

In the following we describe the most important security requirements that a D2D communications system should guarantee while underlining the inherent conflicting nature of these requirements, particularly between anonymity, privacy, and *traceability*:

- *Authenticated key agreement (AKA)*: in order to secure D2D communications in the three typical scenarios, especially in the out-of-coverage scenario, the authentication of any two ProSe-enabled UE opportunistically encountered in CL-PKC environment must be carried out based on public keys presented by each entity.
- *Confidentiality*: after an AKA is performed between any two ProSe-enabled UE opportunistically encountered, their D2D content should be encrypted by use of a new established symmetric key.
- *Anonymity*: refers to the protection of a ProSe-enabled UE identity to not be linked with other D2D communication sessions so that a profile could be defined. It should be noted that exchanged data during these sessions are not necessarily encrypted.
- *Privacy*: through symmetric cryptography, privacy refers to the ability to keep ProSe-enabled UE's sensitive information away from an unauthorized entity. It should be noted that ProSe-enabled UE identity is not necessarily protected.
- *Revocability*: refers to the ability to reprieve ProSe-enabled UE's privilege of a D2D service if it is detected as malicious or when the commercial agreement expires.
- *Traceability*: refers to the ability to keep D2D communications in check, especially to track the source of security violation attempts without compromising either privacy or anonymity.

3.3 Security Model

In (Al-Riyami and Paterson, 2003), authors adopted two adversarial models to proof that their CL-PKE scheme is semantically secure against a fully-adaptive

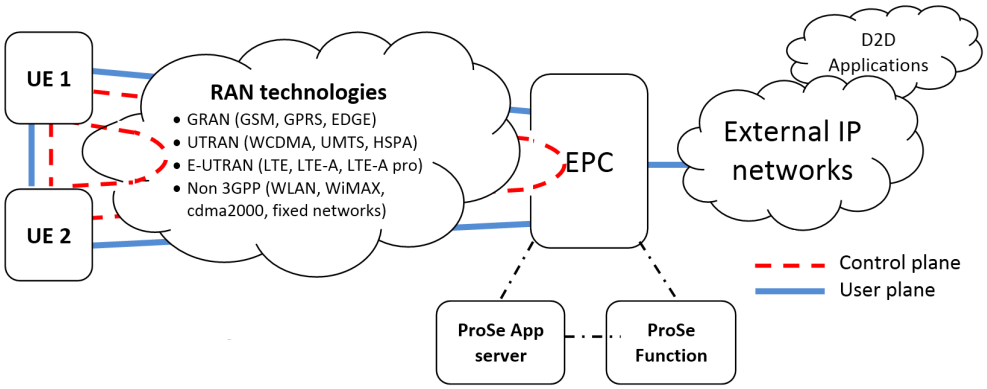


Figure 4: Basic Architecture of the ProSe underlying 3GPP's EPS.

chosen ciphertext attacker (IND-CCA): Type I adversary which cannot access to the KGC's master key but has the ability to replace the public key of all identities with a value of its choice, and Type II adversary which can access to the KGC's master key but may not replace public keys of the legitimate users. Type I adversary models an external attacker while Type II adversary models an internal one. We refer to the later as *curious_but_passive_KGC* which might engage in an adversarial activity such as eavesdropping on ciphertexts and making decryption queries.

In 2007, Au *et al.* proposed for the first time in (Au et al., 2007) a formal model of *malicious_but_passive_KGC*, where the KGC is allowed to generate at the beginning of the system initialization its master public/secret keys pair maliciously so that it can launch the Type II adversary attack more easily in the later stage of the system by eavesdropping passively the ciphertexts sent to a user and trying to decrypt them using its knowledge of the user partial private key. In their model, the assumption that the KGC is trusted at the beginning of the setup stage is removed. The KGC may even have already targeted a particular victim when choosing its master keys pair.

What if we have a nastier KGC? From our point of view, if such malicious and curious KGC exists, nothing can prevent that KGC from being active in the sense that it is able to replace user's public keys, by generating and making available to an illegitimate user a fake public/private keys pair in order to impersonate legitimate users. By eliminating any trust to the KGC, we consider in our security model a *malicious_and_active_KGC* which in addition to having already been malicious at the beginning of the setup stage of the system, impersonates a target user by replacing its public key. That is, the KGC is malicious if it generates its master public/secret keys pair maliciously in order to derive the user secret value, and is active if it is able to replace user public key by generating a fake public/private keys pair for an illegitimate user on behalf of a legitimate user.

Surely, this attack will leave evidence exposing the KGC's actions, since detecting two working public keys for a legitimate user can only result from the existence of two partial private keys binding that legitimate user's identity to these two different public keys. But in practice, and with the possibility of keys' revocation, the detection of the existence of such keys is not obvious especially since the malicious and active KGC can temporarily replace an entity's public key (whose private key is known) in an attempt to obtain sensitive information regarding either the user whose public key has been changed or the user to whom this encrypted information was sent, and then resets the true public key. Furthermore, since the very essence of the introduction of CL-PKC paradigm was authentication procedures decentralization, hence the proposal of CL-AKA protocol that no longer requires infrastructure like that of a PKI, the authentication of two entities opportunistically encountered in CL-PKC environment will be carried out based on public keys presented by each entity.

In order to project the above attacker model to our system model, neither the CNO nor the D2D-SP should be trusted. We should consider them as malicious and active KGCs. However, we suppose that they might not engage in an adversarial activity in a cooperative way. Thus, we design two attacker models: insider and outsider attackers. The first one refers to the CNO or a D2D-SP and the second one refers to any entity which does not belong to the D2D communication system.

- Insider attacker: we design our security model by challenging both the trust between the CNO and the D2D-SP, and the trust placed by ProSe-enabled UEs in either the CNO or the D2D-SP in the sense that, neither CNO would be

able to profile a ProSe-enabled UE according to its D2D application preferences nor the D2D-SP should know the real identity of a ProSe-enabled UE. Thus, insider attacker embodies the well-known Type II Adversary model but with more powerful abilities to the point that it becomes *malicious_and_active_KGC*.

- Outsider attacker: it embodies the well-known Type I Adversary model, so other attacks can be made by UEs which are not ProSe-enabled UEs.

Attackers may compromise both ProSe-enabled UEs pairing and D2D content. Thus, the potential attacks that could be conducted by either insider or outsider attackers are summarized as follows:

1. Eavesdropping attack: typically, it consists in listening passively the radio channel in order to get sensitive data. During ProSe-enabled UEs pairing, this attack could be disastrous for the key management scheme, since it targets ProSe-enabled UE's credentials. On another hand, the eavesdropper tries to decrypt the encrypted D2D content.
2. *Malicious_but_passive_KGC* Attack: it consists in obtaining the private key of a targeted ProSe-enabled UE by choosing a generator P and setting a trapdoor rather than generating a random one during the establishment of the CL-PKC system parameters. That chosen P depends on the ProSe-enabled UE victim's identity.
3. Signature Forgery Attack: it consists in forging the D2D content whether during ProSe-enabled UEs pairing or D2D transactions.
4. Key Compromise Impersonation Attack: it consists in impersonating a legitimate ProSe-enabled UE B by an attacker to communicate with another legitimate ProSe-enabled UE A whose long term private key was learned by that attacker (Li et al., 2013).
5. Man-In-The-Middle-Attack (MITMA): it consists in stealthily intercepting and replacing two ProSe-enabled legitimate UEs' credentials in order to establish D2D connections in the middle.

In the next section we detail our new CL-PKC system construction for multiple KGCs to secure ProSe. Considering the presence of multiple KGCs, our approach is simply the application of the CL-PKE, CL-PKS, and CL-AKA schemes proposed in (Al-Riyami and Paterson, 2003), (Huang et al., 2005), and (Li et al., 2013), respectively, with the appropriate modifications. Each of the KGCs is assigned by the RA the same CL-PKC system parameters without a wide master public key so that it can generate its own public/private keys pair. Then, we aggregate these parameters so that we define a new logical KGC whose common public and private keys will be calculated from KGCs' public and private keys, respectively. Our proposed CL-PKE and CL-PKS will be detailed in Sections 4.3, 4.4, and 4.5. As for our CL-AKA, it will be detailed in Section 4.6.

4 OUR NEW CL-PKC CONSTRUCTION TO SECURE D2D COMMUNICATIONS

4.1 Overview

Our main idea is to aggregate two CL-PKC systems parameters into one (Figure 5). This allows, in addition to struggling security issues in the three typical scenarios, eliminating the conflicting security requirements as mentioned above, particularly, the privacy against the CNO which should play the role of KGC_1 and anonymity against the D2D-SP which should play the role of KGC_2, and to avoid any possible breach of trust by either KGC_1 or KGC_2. Furthermore, the new aggregated CL-PKC system gives different networks the opportunity to be compatible and to work in a cooperative way. On another hand, we have considered an additional new entity (RA) as a TTP to overcome the *malicious_but_passive_KGC* (see Section 3.3). Here, the notion of trust concerns only the CL-PKC system parameters, which are public in essence, that the RA supplies to the different KGCs without a *system_wide_master_key*.

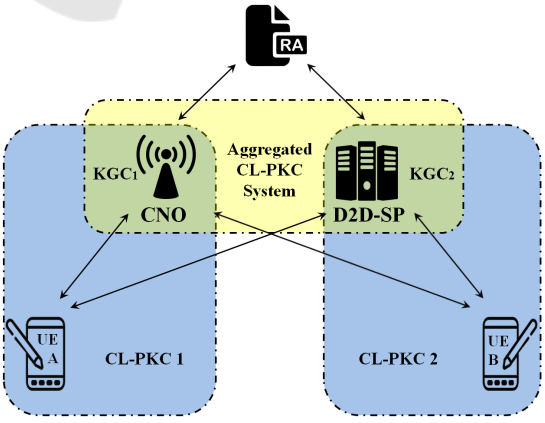


Figure 5: Two CL-PKC system parameters' aggregation.

4.2 Background Definitions

The notations that we use in the rest of the paper are summarized in Table 1. Let e be a bilinear map. As mentioned in (Al-Riyami and Paterson, 2003), e will be derived from either the Weil (Lang, 1987) or Tate (Frey et al., 1999) pairing on an elliptic curve over a finite field. Pairing map e's properties are as follows:

1. **Bilinearity:** the map e is bilinear if given $Q,W,Z \in \mathbb{G}_1$, we have $e(Q,W+Z) = e(Q,W).e(Q,Z)$. As consequence, we have for any $a,b \in \mathbb{Z}_q$: $e(aQ,bW) = e(Q,W)^{ab} = e(abQ,W) = e(Q,abW)$.
2. **Non-degeneracy:** the map e is non-degenerate if $e(P,P) \neq 1_{G2}$.
3. **Computability:** the map e is efficiently computable.

For a more comprehensive description of curves selection with suitable properties and for a practical implementation of pairings, we refer to (Galbraith et al., 2002; Barreto et al., 2002a; Barreto et al., 2002b).

In the following, we introduce the Bilinear Diffie-Hellman Problems (BDHP) on which security of our schemes is based. Given as input $< P,aP,bP,cP > \in \mathbb{G}_1$ with uniformly random choices of $a,b,c \in \mathbb{Z}_q^*$:

- Computational BDHP (CBDHP): output $e(P,P)^{abc} \in \mathbb{G}_2$.
- Generalized BDHP (GBDHP): output a pair $< Q \in \mathbb{G}_1^*, e(P,Q)^{abc} \in \mathbb{G}_2 >$.
- Decisional BDHP (DBDHP): given also $h \in \mathbb{G}_2$, output whether or not $h = e(P,P)^{abc}$.
- Gap BDHP (GapBDHP): given also DBDH oracle that is able to decide whether a tuple $< P,aP,bP,cP,h >$ satisfies $h = e(P,P)^{abc}$ or not, output $e(P,P)^{abc}$.

For further details and a comprehensive description of these mathematical problems, we refer to (Cheon and Lee, 2002).

4.3 System Initialization

As mentioned in Section 3.1, we consider here an additional entity, that we call Regulatory Authority (*RA*), which is responsible only for generating the parameters of our CL-PKC system without a master public key as it was done traditionally, and this to avoid the *malicious_but_passive_KGC*. *RA* will provide the different KGCs with the appropriate parameters of a CL-PKC system so that they can choose separately their sub-master public/secret keys pair as a first step, and then agree on a common public key related to the new aggregated CL-PKC system, whose private key could only be defined if they agree to exchange their secret keys (which we suppose impossible since they have conflict of interest).

- **Init:** This algorithm is executed by *RA*. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be an additive and a multiplicative groups with a large prime order q, respectively. e is a pairing map. Let H_1,H_2,H_3,H_4 and H_5 be cryptographic hash functions. Let $P \in \mathbb{G}_1$ a random generator and S the signature space. The system parameters are $params =< \mathbb{G}_1,\mathbb{G}_2,e,n,P,H_i,\mathcal{M},\mathcal{C},\mathcal{S} >$, where $i \in \{1,..,5\}$. It is worth stressing that a system's *master_public_key* P_0 is missing at this stage compared to original work (Al-Riyami and Paterson, 2003). It will be calculated later by the different KGCs based on their public keys *sub_public_keys* and their secret keys *sub_master_keys*.
- **Setup:** This algorithm is executed by the different KGCs: the CNO as KGC_1 and the D2D-SP as KGC_2, to agree on the common public key P_0. Each of them chooses randomly its *sub_master_key* : s_1 and s_2, respectively ($s_1,s_2 \in \mathbb{Z}_q^*$). Using *params*, they calculate their *sub_public_keys* : $P_{01} = s_1P$ and $P_{02} = s_2P$, respectively. After exchanging their respective *sub_public_keys*, they calculate the common public key $P_0 = P_{01} + P_{02} = (s_1+s_2)P = sP$ which represents the *public_key* of the new aggregated CL-PKC system parameters. The corresponding logical *master_key*: $s = s_1 + s_2$ remains unknown for both KGCs which publish their CL-PKC system parameters $params_{KGC_i} =< params, P_{0i}, P_0 >$, respectively, where $i = 1,2$.

4.4 Registration

Let ProSe-enabled UE A with identifier ID_A performs two registrations to the different KGCs. As a result, ProSe-enabled UE A sets up two pairs of public and private keys: (P_{A1},S_{A1}) based on KGC_1's CL-PKC system parameters, and (P_{A2},S_{A2}) based on KGC_2's CL-PKC system parameters. Once the public and private keys are established, ProSe-enabled UE A aggregates the two public keys into one public key $P_A = P_{A1} + P_{A2}$ and the two private keys into one private key $S_A = S_{A1} + S_{A2}$. In the following, we give the appropriate algorithms:

- **Set-secret-Value:** This algorithm is executed by ProSe-enabled UE A. It takes as inputs A's identifier $ID_A \in \{0,1\}^*$ and $params_{KGC_i}$ ($i = 1,2$). Then, it outputs $x_A \in \mathbb{Z}_q^*$ as a random secret value.

Table 1: Notations and Definitions.

Notation	Meaning	Definition
k	security parameter	$k \geqslant 1$
q	large prime order	-
n	bit-length of plain-texts	$n \approx log_2 q$
$\mathbb{G}_1$	additive group of order q	-
$\mathbb{G}_2$	multiplicative group of order q	-
e	pairing map	$e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$
P	a random generator	$P \in \mathbb{G}_1$
H_1	cryptographic hash functions	$H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$
H_2		$H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n$
H_3		$H_3 : \{0,1\}^n \times \{0,1\}^n \rightarrow \mathbb{Z}_q^*$
H_4		$H_4 : \{0,1\}^n \rightarrow \{0,1\}^n$
H_5		$H_5 : \{0,1\}^* \times \mathbb{G}_2 \rightarrow \mathbb{Z}_q^*$
$\mathcal{M}$	message space	$\mathcal{M} = \{0,1\}^n$
$\mathcal{C}$	ciphertext space	$\mathcal{C} = \mathbb{G}_1 \times \{0,1\}^n$
$\mathcal{S}$	signature space	$\mathcal{S} = \mathbb{G}_1 \times \mathbb{Z}_q^*$
ID_A	ProSe-enabled UE A's identifier	$ID_A \in \{0,1\}^*$
x_A	ProSe-enabled UE A's secret value	$x_A \in \mathbb{Z}_q^*$
$params$	CL-PKC system parameters	$params =< \mathbb{G}_1, \mathbb{G}_2, e, n, P, H_i, \mathcal{M}, \mathcal{C}, \mathcal{S} >$ $i \in \{1,..,5\}$

- **Set-public-Key:** This algorithm is executed by ProSe-enabled UE A. It takes as inputs $params_{KGC_i}$ $(i = 1,2)$ and A's secret value x_A. Firstly, it checks that the equality $P_0 = P_{01} + P_{02}$ holds. If not, it aborts the algorithm. Otherwise, it calculates A's sub public keys P_{A1} and P_{A2}, and then the ProSe-enabled UE A's aggregated public key $P_A =< X_A, Y_A >$ as follows:
  $P_{A1} =< X_A = x_A P, Y_{A1} = x_A P_{01} = x_A s_1 P >$
  $P_{A2} =< X_A = x_A P, Y_{A2} = x_A P_{02} = x_A s_2 P >$
  $P_A =< X_A, Y_{A1} + Y_{A2} >=< X_A, x_A(P_{01} + P_{02}) >$
  $=< X_A, x_A(s_1 + s_2)P >=< X_A, x_A P_0 >$.
- **Set-partial-Private-key:** This algorithm is executed by each KGC_i $(i = 1,2)$. It takes as input ID_A and outputs A's sub partial private key $D_{Ai} = s_i Q_A$, where $Q_A = H_1(ID_A) \in \mathbb{G}_1^*$, and transmits D_{Ai} to A through an authentic and secure channel.
- **Set-private-Key:** This algorithm is executed by ProSe-enabled UE A. It takes as inputs: $params$, A's sub partial private keys D_{Ai} and A's secret value x_A. Then, it verifies the correctness of these sub partial private keys by checking if the equality $e(D_{Ai}, P) = e(Q_A, P_0)$ holds. If not, it aborts the algorithm. Otherwise, it computes A's sub private keys S_{A1} and $S_{A2} \in \mathbb{G}_1^*$ and then the aggregated private key S_A by following these steps:
  $S_{A1} = x_A D_{A1} = x_A s_1 Q_A$
  $S_{A2} = x_A D_{A2} = x_A s_2 Q_A$
  $S_A = S_{A1} + S_{A2} = x_A(s_1 + s_2)Q_A = x_A s Q_A$.

4.5 Main Cryptographic Operations

After the system initialization and the registration steps, any ProSe-enabled UEs A and B can perform the following cryptographic operations:

- **Encrypt:** This algorithm is executed by any ProSe-enabled UE aiming to send an encrypted message M for ProSe-enabled UE A with identifier $ID_A \in \{0,1\}^*$ and public key $P_A =< X_A, Y_A >$. First, the sender checks the validity of A's public key by verifying that $X_A, Y_A \in \mathbb{G}_1^*$ and that the equality $e(X_A, P_0) = e(Y_A, P)$ holds. If not, it cancels encryption. Second, the sender computes $Q_A = H_1(ID_A) \in \mathbb{G}_1^*$, and chooses a random $\sigma \in \{0,1\}^n$ to calculate $r = H_3(\sigma, M)$. Finally, it computes the ciphertext:
  $C =< U, V, W >$
  $=< rP, \sigma \oplus H_2(e(Q_A, Y_A)^r), M \oplus H_4(\sigma) >$.
- **Decrypt:** This algorithm is executed by ProSe-enabled UE A. To decrypt an encrypted message M using its private key S_A, ProSe-enabled UE A computes: $\sigma' = V \oplus H_2(e(S_A, U))$, then computes the decryption of the ciphertext C as: $M' = W \oplus H_4(\sigma')$.
  To verify the correctness of M, it sets $r' = H_3(\sigma', M')$ and test if equation $U = r'P$ holds.
- **Sign:** This algorithm is executed by ProSe-enabled UE A aiming to send a signed message M using its private key S_A. First, ProSe-

enabled UE A chooses random $a \in \mathbb{Z}_q^*$, and computes $r = e(aP,P) \in \mathbb{G}_2$. Then it sets $v = H_5(M,r,e(S_A,P)) \in \mathbb{Z}_q^*$, and computes $U = vS_A + aP \in \mathbb{G}_1$. Finally, the algorithm outputs the signature $< U,v >$.

- **Verify:** To verify a signature $< U,v >$ on a message M from ProSe-enabled UE A with identity ID_A and public key $P_A =< X_A,Y_A >$, ProSe-enabled UE B checks the validity of P_A, and computes $r^{'} = e(U,P).e(Q_A,-Y_A)^v$. If the equation $v = H(M,r^{'},e(Q_A,Y_A))$ holds, the signature is valid.

Notice that *Encrypt* and *Decrypt* algorithms are identical to the provably secure encryption and decryption algorithms in the $FullCL-PKE$ in (Al-Riyami and Paterson, 2003), and that *Sign* and *Verify* algorithms are also identical to those specified in (Huang et al., 2005). We did not place any reliance on the CL-PKS scheme introduced in (Al-Riyami and Paterson, 2003) because of its insecurity as it was pointed out and proofed in (Huang et al., 2005).

Formally, our CL-PKE and CL-PKS schemes are specified by the above five common algorithms: *Setup*, *Partial-Private-Key-Extract*, *Set-Secret-Value*, *Set-Private-Key*, *Set-Public-Key*, and the above four additional algorithms: *Encrypt* and *Decrypt* for the CL-PKE scheme, and *Sign* and *Verify* algorithms for the CL-PKS scheme.

4.6 User Equipment Pairing

After performing the registration step (Section 4.4), any two ProSe-enabled UEs opportunistically encountered execute our CL-AKA scheme as follows. Firstly, they exchange their public keys for verification and authentication. And after, they establish common credentials for their secure D2D communications.

Let ProSe-enabled UEs A and B with their respective pairs of public/private keys $< P_A,S_A >$ and $< P_B,S_B >$, be the intended participants in the pairing step. First, each of them chooses random values $a,b \in \mathbb{Z}_q^*$ and calculates $T_A = aP$ and $T_B = bP$, respectively. After exchanging their triplets $< ID_A,P_A,T_A >$ and $< ID_B,P_B,T_B >$, both of them verify that the equality $e(X_{UE},P_0) = e(Y_{UE},P)$ where $UE \in \{A,B\}$ holds in order to check the validity of each others' public key. Finally, they may calculate the following possible symmetric keys:

- $K_{AB}^1 = e(D_A,X_B+T_B) = K_{BA}^1 = e(Q_A,P_0)^{(x_B+b)}$
- $K_{AB}^2 = e(Q_B,P_0)^{(x_A+a)} = K_{BA}^2 = e(D_B,X_A+T_A)$
- $K_{AB}^3 = x_AX_B = K_{BA}^3 = x_BX_A$
- $K_{AB}^4 = x_AT_B = K_{BA}^4 = x_BT_A$
- $K_{AB}^5 = aT_B = K_{BA}^5 = bT_A$
- $K_{AB}^6 = aX_B = K_{BA}^6 = bX_A$
- $K_{AB}^7 = e(Q_B,Y_B)^a.e(S_A,T_B) =$ $K_{BA}^7 = e(Q_A,Y_A)^b.e(S_B,T_A)$.

Let k be the session key as in (Li et al., 2013): $k = H_1(ID_A|ID_B|P_A|P_B|T_A|T_B|K_{AB}^1|K_{AB}^2|K_{AB}^4|K_{AB}^5)$.

5 SECURITY ANALYSIS

In this section, we analyze the different security requirements and prove the resistance of our system against well known attacks. First, we briefly discuss the security of $FullCL-PKE$ (Al-Riyami and Paterson, 2003), $CL-PKS$ (Huang et al., 2005) and $CL-AKA$ (Li et al., 2013) which are utilized in the proposed aggregating CL-PKC system. Later, we discuss how the proposed aggregating CL-PKC system achieves our security goals. The $FullCL-PKE$ (Al-Riyami and Paterson, 2003) and the $CL-PKS$ (Huang et al., 2005) provide confidentiality (as *indistinguishability* against adaptive chosen ciphertext attack (IND-CCA)) and *unforgeability* for encrypted and signed messages, respectively. These security requirements are based on the intractability of the GBDH and ECDH problems, respectively. That is, it is impossible to expose or forge the full private key of a ProSe-enabled UE based on the difficulty of GBDH and ECDH problems, without the knowledge of both KGC's sub-master private keys and ProSe-enabled UE's secret value. As for $CL-AKA$ (Li et al., 2013), it is provably secure against a fully adaptive adversary in the random oracle model, provided that the underlying problem of Gap BDHP is hard. Further details on security proofs are provided in (Al-Riyami and Paterson, 2003; Huang et al., 2005; Li et al., 2013).

5.1 Illustrative Scenario of a D2D Application

Although D2D communications can take advantage from key management scheme already available in LTE-A, our solution is proposed independently from the existing one in order to achieve more and higher security levels, particularly to avoid any possible breach of trust by a TTP and to face the inherent conflicting security requirements.

Below we give a practical example of a D2D scenario to effectively analyze our new CL-PKC system construction and the related schemes. Let RA be a

regulatory authority in a country, CNO and D2D-SP foreign investors in this country such as the CNO supports ProSe environment and the D2D-SP provides authentic videos to ProSe-enabled UE such as YouTube's content through a set of D2D domain applications.

The initialization and setup of our CL-PKC system is performed by the RA through the execution of *Init* algorithm and both the CNO and the D2D-SP through the execution of $Setup$ algorithm, respectively (Section 4.3). Let A be the real identity of a UE which aims to benefit from the ProSe and D2D domain applications. First, it registers to RA in order to get a pseudo-identity ID_A. Thereafter, it registers to both CNO and D2D-SP with that pseudo-identity through the execution of registration algorithms (Section 4.4): $Set-secret-value$, $Set-public-key$, $Set-partial-private-key$ and $Set-private-key$. At this point to the registration process, A becomes a ProSe-enabled UE and can benefit from either the D2D domain applications and ProSe.

Once the CL-PKC system is established, the D2D-SP send new authentic videos to ProSe-enabled UEs so that they can share them with other ProSe-enabled UEs through D2D communications in order to offload the cellular network. A's ProSe-enabled UE download a new video through the appropriate D2D application and share it to any other ProSe-enabled UE B opportunistically encountered. But before that, our CL-AKA scheme must be performed by the concerned ProSe-enabled UEs in the pairing step (Section 4.6).

5.2 Resistance against Malicious but Passive KGC Attack

In our system model (Section 3.1), RA does not provide the different KGCs a common public key P_0 through its CL-PKC system parameters *params* as it was done traditionally. This was considered particularly as responsibility decentralization in terms of security between RA and the two KGCs so that the *malicious_but_passive_KGC* could not exist. By eliminating any trust to the different KGCs, we have avoided this attack by entrusting the generation of CL-PKC parameters, especially the random generator P to the RA rather than entrusting it to the KGCs. Of course, nothing can prevent RA from mounting such an attack, but we have considered that RA is trustful and is external from the system (Figure 5).

5.3 Resistance against Malicious and Active KGC Attack

Firstable, and according to Theorem 1 in (Al-Riyami and Paterson, 2003), it is trivial to show that our CL-PKE is IND-CCA secure against standard adversarial type I and type II models, which follows directly from the employment of the $FullCL-PKE$ (Al-Riyami and Paterson, 2003). We are only left with the proof of our CL-PKE against a malicious-and-active KGC (CNO or D2D server provider) which, in addition to replacing the secret values of legitimate ProSe-enabled UEs with values of its choice, replaces also the long-term legitimate ProSe-enabled UEs' public keys.

Thanks to the aggregation technique of two CL-PKC system parameters, we have been able to define another set of aggregated parameters, notably the master public key of the system whose corresponding master private key is not explicitly defined, hence not known by both KGCs. This means that even if both KGCs are malicious-and-active, and may therefore engage in a *malicious_and_active_KGC* attack, they can not impersonate a legitimate ProSe-enabled UE. This is mainly because ProSe-enabled UE's partial private key is dependent on the two KGC master secrets: $D_A = (s_1 + s_2)Q_A$. In other words, it is impossible to expose or forge the aggregated master key s of our CL-PKC system based on the difficulty of CBDHP, without the knowledge of both KGC's sub-master private keys s_1 and s_2. Of course, nothing can prevent the two KGCs from exchanging their sub-master private keys, but this is not obvious especially in the presence of a conflict of interest between them.

5.4 Resistance against Signature Forgery Attack

According to Theorem 3 in (Huang et al., 2005), it is also trivial to show that our CL-PKS is unforgeable against standard type I adversary model which follows directly from the employment of $CL-PKS$ (Huang et al., 2005). We are only left with the proof of our CL-PKS against a malicious-and-active KGCs. Here, also thanks to the aggregation technique of two CL-PKC system parameters, no one of both KGCs could forge the signature of any ProSe-enabled UE as it was defined in (Huang et al., 2005).

5.5 Resistance against MITMA

Generally, key agreement protocols suffer from MITMA which consists in replacing the secret values of legitimate participants with values of the at-

tacker's choice. As a consequence, an attacker can impersonate both legitimate participants, and the secret key that was to be secret does not become so. Under a *malicious_and_active_KGC*, CL-AKA in (Al-Riyami and Paterson, 2003) and any scheme sharing the same key structure and generation procedures as that of (Al-Riyami and Paterson, 2003) are vulnerable to such an attack even if the pairs of public and private keys are generated by binding a ProSe-enabled UE's public key to its identity. This is possible if a KGC, in addition to replacing the secret values of legitimate participants with values of its choice, replaces also the long-term legitimate participants' public keys. In our CL-AKA, a *malicious_and_active_KGC* cannot mount such an attack since the aggregated master key $s = s_1 + s_2$ is implicitly defined based on the two KGC master secrets. That is, it is impossible to expose or forge the aggregated master key s based on the difficulty of CBDHP, without the knowledge of both KGC's sub-master private keys s_1 and s_2.

Furthermore, according to Theorem 1, 2, 3, and 4 in (Li et al., 2013), our proposed CL-AKA is secure against all known attacks to an authenticated key agreement protocol providing that the underlying GapBDHP is hard.

5.6 Key Compromise Impersonation Attack

Unlike the work of (Li et al., 2013), our CL-AKA is KCI resistant not only because the attacker cannot compute $K_{AB}^2 = e(Q_B, P_0)^{(x_A+a)} = K_{BA}^2 = e(D_B, X_A + T_A)$ since he/she does not have neither the value a chosen from A nor the value D_B owned by B, but our CL-AKA immediately resists KCI attack when checking the attacker's public key validity. That is, in (Li et al., 2013) the public key of an entity A is $P_A = x_A P$, so there is no way to check and authenticate it. However in ours the public key $P_A = < X_A, X_A >$, as in the original scheme (Al-Riyami and Paterson, 2003), has two parts $X_A = x_A P$ and $Y_A = x_A s P$. The second part is considered as the KGC's signature which permits to any entity to authenticate A by verifying that the equation $e(X_A, P_0) = e(Y_A, P)$ holds.

5.7 Eliminating Conflicting Security Requirements

As mentioned in Section 1, a security requirement conflict can usually arise when anonymity and *traceability* are simultaneously required of a central authority. In our context, this was solved by decentralizing responsibility between multiple entities to respond the fully mistrust assumption regarding a TTP. Through our system model, RA can play another role as a registration entity which provides ProSe-enabled UE A the pseudo-identity ID_A and keeping its real identity secret. Thus, in the proposed schemes both the CNO and the D2D-SP have not access to the A's real identity, hence anonymity is guaranteed. However, this anonymity remains conditional such that misbehaving entities in the network remain traceable. This *traceability* is guaranteed cooperatively by both the RA on one hand and the CNO and the D2D-SP on the other hand, this is in order to detect misbehaving entities through their pseudo-identities. Further security mechanisms can be incorporated to the proposed CL-PKC construction so that the CNO or the D2D-SP could report security violation to the RA if such a violation necessitates to reveal the real identity of a user.

6 CONCLUSION

We proposed in this paper a new CL-PKC system construction for multiple KGCs to secure D2D communications based on aggregating two KGCs' system parameters into one. This has multiple advantages since the common public key can be calculated and published by both KGCs while the common private key remains unknown and implicitly defined. That is, the common private key could be revealed only if the concerned KGCs exchange their sub-private keys assuming the intractability of CBDHP. The main advantage consists in preventing the CL-PKC system from a stronger adversary which is a *malicious_and_active_KGC*. Another advantage consists in giving different networks the opportunity to be compatible and to work cooperatively. To make concrete the proposed construction, we proposed also a new CL-PKE, CL-PKS, and CL-AKA schemes applied in ProSe environment. The proposed schemes overcome security issues in all scenarios, particularly in *out_of_coverage* scenario where any two ProSe-enabled UEs opportunistically encountered can authenticate each other based on their respective public keys, Thereafter, they establish a common session key to encrypt their D2D traffic.

REFERENCES

3GPP (February 2014). Study on architecture enhancements to support proximity services (ProSe) (Rel. 12). Technical Report (TR) 23.703, 3rd Generation Partnership Project (3GPP). V12.0.0.

3GPP (June 2013). Feasibility study for proximity services

(ProSe) (Rel. 12). Technical Report (TR) 22.803, 3rd Generation Partnership Project (3GPP). V1 2.2.0.

Al-Riyami, S. S. and Paterson, K. G. (2003). Certificateless public key cryptography. In *Advances in Cryptology - ASIACRYPT 2003, 9th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, November 30 - December 4, 2003, Proceedings*, pages 452–473.

Alkubaisy, D. (2017). A framework managing conflicts between security and privacy requirements. In *11th International Conference on Research Challenges in Information Science, RCIS 2017, Brighton, United Kingdom, May 10-12, 2017*, pages 427–432.

Au, M. H., Mu, Y., Chen, J., Wong, D. S., Liu, J. K., and Yang, G. (2007). Malicious kgc attacks in certificateless cryptography. In *Proceedings of the 2Nd ACM Symposium on Information, Computer and Communications Security*, ASIACCS '07, pages 302–311, New York, NY, USA. ACM.

Barreto, P. S. L. M., Kim, H. Y., Lynn, B., and Scott, M. (2002a). Efficient algorithms for pairing-based cryptosystems. *IACR Cryptology ePrint Archive*, 2002:8.

Barreto, P. S. L. M., Lynn, B., and Scott, M. (2002b). Constructing elliptic curves with prescribed embedding degrees. In *Security in Communication Networks, Third International Conference, SCN 2002, Amalfi, Italy, September 11-13, 2002. Revised Papers*, pages 257–267.

Cheon, J. H. and Lee, D. H. (2002). Diffie-hellman problems and bilinear maps. *IACR Cryptology ePrint Archive*, 2002:117.

Ferrag, M. A., Maglaras, L., and Ahmim, A. (2017). Privacy-preserving schemes for ad hoc social networks: A survey. *IEEE Communications Surveys Tutorials*, 19(4):3015–3045.

Frey, G., Müller, M., and Rück, H. (1999). The tate pairing and the discrete logarithm applied to elliptic curve cryptosystems. *IEEE Trans. Information Theory*, 45(5):1717–1719.

Galbraith, S. D., Harrison, K., and Soldera, D. (2002). Implementing the tate pairing. In *Algorithmic Number Theory, 5th International Symposium, ANTS-V, Sydney, Australia, July 7-12, 2002, Proceedings*, pages 324–337.

Haus, M., Waqas, M., Ding, A. Y., Li, Y., Tarkoma, S., and Ott, J. (2017). Security and privacy in device-to-device (d2d) communication: A review. *IEEE Communications Surveys Tutorials*, 19(2):1054–1079.

Hsu, R., Lee, J., Quek, T. Q. S., and Chen, J. (2018). Graad: Group anonymous and accountable d2d communication in mobile networks. *IEEE Transactions on Information Forensics and Security*, 13(2):449–464.

Huang, X., Susilo, W., Mu, Y., and Zhang, F. (2005). On the security of certificateless signature schemes from asiacrypt 2003. In *Cryptology and Network Security, 4th International Conference, CANS 2005, Xiamen, China, December 14-16, 2005, Proceedings*, pages 13–25.

Lang, S. (1987). *Elliptic Functions*, volume 12. Springer, New York, NY.

Li, X., Zhang, Y., and Zhang, G. (2013). A new certificateless authenticated key agreement protocol for SIP with different kgcs. *Security and Communication Networks*, 6(5):631–643.

Nait Hamoud, O., Kenaza, T., and Challal, Y. (2018a). Security in device-to-device communications: a survey. *IET Networks*, 7(1):14–22.

Nait Hamoud, O., Kenaza, T., and Challal, Y. (2018b). Towards using multiple KGC for CL-PKC to secure D2D communications. In *2018 International Conference on Smart Communications in Network Technologies (SaCoNeT)*, pages 283–287.

Paja, E., Dalpiaz, F., and Giorgini, P. (2013). Managing security requirements conflicts in socio-technical systems. In *Conceptual Modeling - 32th International Conference, ER 2013, Hong-Kong, China, November 11-13, 2013. Proceedings*, pages 270–283.

Shamir, A. (1984). Identity-based cryptosystems and signature schemes. In *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*, pages 47–53.

Sun, J., Zhang, C., Zhang, Y., and Fang, Y. (2011). Sat: A security architecture achieving anonymity and traceability in wireless mesh networks. *IEEE Transactions on Dependable and Secure Computing*, 8(2):295–307.

Tehrani, M. N., Uysal, M., and Yanikomeroglu, H. (2014). Device-to-device communication in 5g cellular networks: challenges, solutions, and future directions. *IEEE Communications Magazine*, 52(5):86–92.

Wang, M. and Yan, Z. (2015). Security in d2d communications: A review. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 1199–1204.

Zhang, A., Wang, L., Ye, X., and Lin, X. (2017). Light-weight and robust security-aware d2d-assist data transmission protocol for mobile-health systems. *IEEE Trans. Information Forensics and Security*, 12(3):662–675.

Zhao, C., Yang, S., Yang, X., and McCann, J. A. (2017). Rapid, user-transparent, and trustworthy device pairing for d2d-enabled mobile crowdsourcing. *IEEE Trans. Mob. Comput.*, 16(7):2008–2022.

Zhou, C. (2018). Comments on "light-weight and robust security-aware d2d-assist data transmission protocol for mobile-health systems". *IEEE Transactions on Information Forensics and Security*, 13(7):1869–1870.