Distributed Trusted Authority-based Key Management for Beyond 5G Network Coding-enabled Mobile Small Cells

Marcus de Ree
Instituto de Telecomunicações
Aveiro, Portugal
University of South Wales
Pontypridd, UK
mderee@av.it.pt

Georgios Mantas
Instituto de Telecomunicações
Aveiro, Portugal
University of Greenwich
London, UK
gimantas@av.it.pt

Jonathan Rodriguez
Instituto de Telecomunicações
Aveiro, Portugal
University of South Wales
Pontypridd, UK
jonathan@av.it.pt

Ifiok E. Otung
University of South Wales
Pontypridd, UK
ifiok.otung@southwales.ac.uk

***Abstract*—The 5G cellular network is projected to be introduced in 2020 and takes advantage of the small cell technology to deliver ubiquitous 5G services in an energy efficient manner. The next logical step is the introduction of network coding-enabled mobile small cells (NC-MSCs). These are networks of mobile devices which can be set up on-the-fly, based on demand, and cover the urban landscape. Furthermore, they allow network offloading through multi-hop device-to-device (D2D) communication to provide high data rate services. In this paper we introduce DISTANT, a decentralized key management scheme specifically designed to provide security in a network which takes advantage of the benefits of NC-MSCs. In our key management scheme, we distribute the certification authority (CA) functions using threshold secret sharing. Each network node is provided with a share of the master private key such that key management services are available "anywhere, anytime". Finally, our distributed CA takes advantage of the self-generated certificate paradigm. Certificates can therefore be issued and renewed without the interaction of the distributed CA which minimizes the communication overhead.**

***Index Terms*—5G, Beyond 5G, Decentralized Systems, D2D Communications, Key Management, Mobile Small Cells, Network Coding, Security, Small Cells, Wireless Ad Hoc Networks**

I. Introduction

It has been almost a decade since the 4G cellular network was introduced. Since that time, the number of connected wireless devices (e.g., PDA's, smartphones, tablets, machines falling within the Internet-of-Things concept) has seen immense growth. By 2021, the number of connected wireless devices will have grown by a factor between 100 and 10,000 with mobile demands having grown by a factor of 1,000 per device [1]. This surge puts a lot of pressure on the 4G network which has to share its resources among the growing number of devices and its increasing demand in mobile data. This causes a reduction in data rates and an increase in latency.

To address these challenges, new technologies are emerging to create the next generation 5G network. One of these is small cell technology. The small cell technology is the most effective solution to deliver ubiquitous 5G services in an energy efficient manner to its users. The next logical step would be the introduction of network coding-enabled mobile small cells (NC-MSCs). These are networks of mobile devices (i.e. user equipment) which can be set up on-the-fly, based on demand, and cover the urban landscape. The relative close proximity of these mobile devices allow for multi-hop device-to-device (D2D) communications. This removes the necessity for network operators to install and maintain additional network infrastructure, enables network offloading and provides high data rate services such as video sharing, gaming and proximity-aware social networking with improved throughput, energy efficiency and latency [2].

The current network infrastructure secures data transmissions of network subscribers through the offline distribution of cryptographic keys, present in SIM cards. These keys establish a secure channel between the mobile device and the network infrastructures, authenticates network subscribers and provide them access to network resources. The network infrastructure essentially acts as a router to deliver data to communicating network subscribers. To offload the network using NC-MSCs, multi-hop D2D communications has to be secured. This requires cryptographic keys which are shared between any arbitrary set of mobile devices. Furthermore, these keys require updating mechanisms to guarantee security over an extended period of time and revocation mechanisms in the event that a mobile device has been maliciously compromised and no longer correctly authenticates the owner of that mobile device. Therefore, providing secure multi-hop D2D communications in NC-MSCs require its own key management scheme.

Traditionally, a key management scheme relies on an online centralized trusted third party (TTP). This TTP is considered trustworthy and secure by every user inside the network. It can therefore distribute cryptographic keys between any set of network devices which would consequently be used to set up a secure communications channel. However, the authors believe that neither any individual mobile device nor the network

This research work leading to this publication has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement H2020-MSCA-ITN-2016-SECRET-722424.

infrastructure can act as the online centralized TTP to securely distribute cryptographic keying material in an online fashion. Neither of these entities is considered to be secure against physical compromise or denial-of-service (DoS) attacks. These attacks could cause the distribution of falsified cryptographic keys and the unavailability of the key management service respectively.

In this paper, we introduce our design of the novel key management scheme called DISTANT (DIStributed Trusted Authority-based key managemeNT). This is the first key management scheme that is specifically designed to provide security in a network architecture wishing to take advantage of the benefits of NC-MSCs. The main features are the use of threshold secret sharing [3] and self-generated certificates [4]–[6]. The trust distributing capabilities of threshold secret sharing allows a distributed TTP to provide key management services. Furthermore, verifiable secret sharing and proactive secret sharing provides robustness against malicious users. The paradigm of self-generated certificates provides keying material such that proxy signatures can be created. These are signatures created by network users on behalf of the TTP. This removes the necessity to periodically contact the distributed TTP to renew certificates and thus minimizes the communication overhead. Moreover, the proposed protocols are based on the Discrete Logarithm Problem (DLP) which uses modular arithmetic and are therefore computationally more efficient than schemes using pairing-based operations. Lastly, the survey [7] proposed seven requirements (security, connectivity, overhead, scalability, sustainability, fairness and secure routing independence) that a key management scheme must satisfy in order to efficiently secure beyond 5G mobile small cells. This key management scheme satisfies all of these seven requirements.

The remainder of the paper is organized as follows. Section II reviews the related works. Section III discusses the network model. Section IV describes the proposed key management protocols. Section V discusses our future work and section VI concludes this paper.

II. Related Work

The idea of a distributed TTP was born in 1999 with the emergence of mobile ad hoc networks (MANETs) and the necessity to provide security in these types of networks. MANETs are unable to rely on any network infrastructure such that the mobile devices making up the network are required to self-organize the key management and provide security services. Zhou et al. [8] proposed the first decentralized key management scheme by distributing trust from an online centralized CA to a proper set of network users, called servers. By utilizing threshold secret sharing [3], the master private key (usually in possession by the centralized CA) is divided into n shares. These shares are then distributed to n users, such that a threshold t users can collectively provide certification services. The signed certificates can be distributed throughout the network and their signatures can be verified using the master public key. Furthermore, verifiable [9], [10] and proactive secret sharing [11], [12] were proposed to provide robustness against malicious attackers [13]. However, the asymmetric distribution of trust and the associated workload to provide certification services may lead to an unavailability of the service in large parts of the network, cause selfish behavior and complicates the creation of cooperating mechanisms. Luo et al. [14] solved these issues by distributing the shares of the master private key among all the network users. Even joining network nodes are provided with their own master private key share such that certification services are always locally available. Unfortunately, this decentralized scheme still requires the periodic interaction with a threshold number of network users due to expiring certificates. This translates to a large communication overhead during network deployment.

Advancements in the field of cryptology gave rise to self-generated certificates [4]–[6]. This paradigm combines certificateless public key cryptography [15] with self-certifying keys [16], [17]. Baek et al. [4] and Liu et al. [5] based their scheme on the earlier works of self-certifying keys [16], [17] which are unable to provide explicit authentication. Lai et al. [6] based their scheme on the improved concept of self-certificates [18] which is able to provide explicit authentication and thus reaches an increased level of trust. The construction of self-generated certificates allows users to create proxy signatures. These are signatures created by a network user on behalf of the TTP. This removes the necessity to periodically contact the distributed TTP to renew certificates. The concept of self-generated certificates was eventually proposed for a MANET environment in [19], however this scheme suffers from three major drawbacks. First, verifiable secret sharing is not included which allows malicious users to transmit false keying material without being detected. Second, proactive secret sharing is not included which allows a mobile adversary [13] from collecting a threshold amount of shares and break the security of the entire network. Finally, the master key pair is created in a distributed fashion. This requires the complete trust of the initial users since a participating malicious user could already compromise security during the network initialization.

III. Proposed Model

A. Network Model

The densification of the urban landscape by means of NC-MSCs and network offloading by means of enabling multi-hop D2D communications leads to a network which is capable of increasing data rates and energy efficiency while reducing the latency and interference. This network model is introduced by the EU-funded H2020-MSCA project "SECRET" [20] and provides opportunities for both network operators and network users. The network model is illustrated in Fig. 1.

Many of the prescribed advantages can be credited to the introduction of ordinary small cells. Since the strength of a radio signal diminishes with the square of the distance, replacing large transmissions to and from the base station (BS) by multiple shorter transmissions provides significant energy savings. Similarly, the shorter and less powerful signals can reduce interference which allows for a higher throughput and

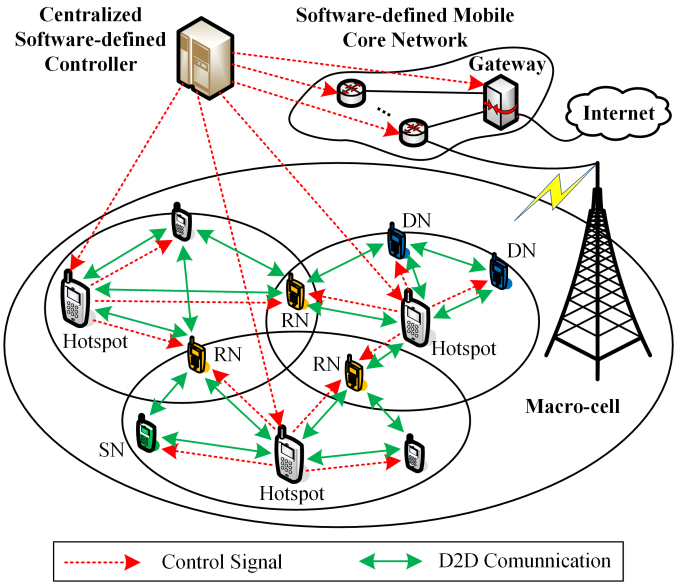

Fig. 1. The network model as introduced by project "SECRET" [21].

increased data rates. Lower latency is realized by providing a more direct route between a source node (SN) and a destination node (DN). Nevertheless, mobile small cells provide additional advantages. They can be set up on-the-fly, based on demand, at any place, at any time, using existing mobile devices. This wireless ad hoc network can therefore function at a low cost since network operators are not required to install and maintain additional network infrastructure. Furthermore, mobile small cells support time and space varying traffic [22]. Finally, network coding can be utilized which optimizes the throughput of data.

In the network model, the cellular network is partitioned into a network (or cloud) of NC-MSCs. Each of these is controlled and maintained by a hotspot (i.e. cluster-head). This is a mobile node within the cluster that is selected to become the local radio manager. In addition, each hotspot is controlled by a centralized software-defined controller. Through cooperation these hotspots form a wireless network that has several gateways/entry points to the mobile network using intelligent high-speed connections. Data traffic between mobile nodes is established through multi-hop D2D communications. Suppose that a mobile node wishes to share data, such as a multimedia file or data related to an online multiplayer game, with two other mobile nodes. The mobile node in possession of this data (i.e. SN) transmits the data to the mobile nodes requesting the data (i.e. DNs). Notice that these mobile nodes are not required to be in the same mobile small cell. Using multi-hop D2D communications and the assistance of relay nodes (RNs), data can be transmitted from the SN to the DNs.

B. Discrete Logarithm Problem

The security of this key management scheme is based on the difficulty of solving the Discrete Logarithm Problem (DLP). No polynomial time algorithm is believed to exist which can solve this problem, provided that $p-1$ is not a product of small primes and p is sufficiently large.

Let $\mathbb{Z}_p^*$ be the cyclic multiplicative group of non-zero integers modulo prime p and let g be a generator. Given an element $a \in \mathbb{Z}_p^*$, find the element b such that $a \equiv g^b \pmod p$.

TABLE I
NOTATION TABLE.

Symbol	Description
p, q	Two large prime numbers.
$\mathbb{Z}_p^*, \mathbb{Z}_q^*$	The multiplicative group of integers modulo p and q respectively.
g	A generator of $\mathbb{Z}_p^*$ with order q.
$h(\cdot)$	A collision-free hash function.
$f_M(x)$	The master polynomial of degree $t-1$.
$f_u(x)$	The update polynomial of degree $t-1$.
t	The threshold value of a secret sharing scheme.
a_i	A random coefficient of the master polynomial $f_M(x)$.
(SK_M, PK_M)	The master private key and the master public key.
s_i	The secret share of node i.
ID_i	The network identifier of node i.
w_i	The witness value i of t.
sv_i	The secret value randomly picked by entity i.
$c_{i\to j}$	The partial commitment value created by entity i for node j.
$c_{i,k}$	The commitment value of node i's k^{th} certificate.
$SK_{i\to j}$	The partial private key created by entity i for node j.
$(SK_{i,k}, PK_{i,k})$	The private and public key of node i's k^{th} certificate.
$CERT_{i,k}$	The k^{th} self-generated certificate of node i.
(r, R)	The random signature creation values.
$\sigma = (\alpha, \beta)$	The signature for a self-generated certificate.
$(\bar{R}, \bar{\alpha})$	The signature verification values.
$K_{i,j}$	The pairwise key between node i and j.
Φ	The set of t neighbors of a node.
$\lambda_i^{\Phi}(x)$	The Lagrange coefficient for node i in the set Φ.
$\delta_{i,j}$	The random shuffling value agreed by node i and j.
δ_i	The sum of random shuffling values of node i.

IV. Proposed DISTANT Protocols

This section describes the protocols of the proposed key management scheme. This scheme is divided up into six main protocols: network initialization, creation and update of self-generated certificates, verification of a self-generated certificate, establishment of secure communication, node joining and share updating. Table I specifies the notation used throughout the proposed protocols.

A. Network Initialization

It is assumed that a TTP, such as a network operator or a collaborative effort from multiple network operators initialize

the network. During network initialization, an initial set of t nodes (i.e. mobile devices) are selected and provided with keying material. This keying material allows t nodes to collectively initialize joining nodes during network deployment. Therefore, this scheme does not rely on an online centralized TTP for key management services. The master key pair and its shares are created as follows:

1) The TTP generates two large prime numbers p and q such that $q|p-1$, selects a generator g of the cyclic multiplicative group $\mathbb{Z}_p^*$ with order q and a collision-free hash function $h(\cdot)$.
2) The TTP generates a random master polynomial $f_M(x)$ of degree $t-1$ in which each $a_i \in \mathbb{Z}_q^*$:
$$f_M(x) = a_0 + a_1 x + \cdots + a_{t-1}x^{t-1}. \quad (1)$$
3) The master key pair (SK_M, PK_M) is defined as:
$$SK_M = f_M(0), \quad (2)$$
$$PK_M \equiv g^{SK_M} \pmod{p}. \quad (3)$$
4) The TTP provides each node i with their identifier ID_i and their share s_i of the master private key:
$$s_i \equiv f_M(ID_i) \pmod{q}. \quad (4)$$
5) To allow the verifiability of shares, the TTP computes witness values $w_i \equiv g^{a_i} \pmod{p}$ for $i = \{0, \ldots, t-1\}$ and provides each node with public values (p, q, g, $h(\cdot)$, PK_M, $\{w_i\}$). Each node verifies its share s_i as follows:
$$g^{s_i} \equiv \prod_{j=0}^{t-1} (w_j)^{(ID_i)^j} \pmod{p}. \quad (5)$$

The TTP initializes an interactive protocol with each node i to establish its keying material. This protocol works as follows:

1) The TTP picks a random secret value $sv_{TTP} \in \mathbb{Z}_q^*$ and computes the partial commitment $c_{TTP\to i}$:
$$c_{TTP\to i} \equiv g^{sv_{TTP}} \pmod{p}. \quad (6)$$
2) The TTP transmits $(c_{TTP\to i})$ to node i.
3) The node i picks a random secret value $sv_i \in \mathbb{Z}_q^*$ and computes its commitment c_i:
$$c_i \equiv g^{sv_i} \cdot c_{TTP\to i} \pmod{p}. \quad (7)$$
4) The node i transmits (ID_i, c_i) to the TTP.
5) The TTP computes the partial private key $SK_{TTP\to i}$:
$$SK_{TTP\to i} \equiv sv_{TTP} + SK_M \cdot h(ID_i, c_i) \pmod{q}. \quad (8)$$
6) The TTP transmits $(SK_{TTP\to i})$ to node i.
7) The node i computes its key pair (SK_i, PK_i):
$$SK_i \equiv sv_i + SK_{TTP\to i} \pmod{q}, \quad (9)$$
$$PK_i \equiv g^{SK_i} \pmod{p}. \quad (10)$$
8) The 2-tuple (c_i, SK_i) is the TTP's signature on certification information (ID_i, c_i) and can be verified as follows:
$$PK_i \equiv g^{SK_i} \pmod{p} \equiv c_i \cdot PK_M^{h(ID_i, c_i)} \pmod{p}. \quad (11)$$

Once the t initial nodes have obtained their keying material, the TTP destroys the master polynomial $f_M(x)$ and the associated master private key SK_M. The initialized nodes can create and update their self-generated certificate and collectively provide key management services to joining nodes.

B. Creation and Update of Self-Generated Certificates

In our key management scheme, nodes can independently create and update their self-generated certificate. If these certificates are renewed frequently (e.g. on a daily basis) then key revocation becomes redundant [19]. Let the initial keying material of a node i be $(c_{i,0}, SK_{i,0}, PK_{i,0})$ and kept secret during network deployment. The initial keying material is used to derive keying material $(c_{i,k}, SK_{i,k}, PK_{i,k})$ for the k^{th} self-generated certificate as follows:

1) The node i picks a random secret value $sv_i \in \mathbb{Z}_q^*$ and computes its new commitment $c_{i,k}$:
$$c_{i,k} \equiv g^{sv_i} \pmod{p}. \quad (12)$$
2) The node i computes its new key pair $(SK_{i,k}, PK_{i,k})$:
$$SK_{i,k} \equiv sv_i + SK_{i,0} \cdot h(ID_i, c_{i,k}) \pmod{q}, \quad (13)$$
$$PK_{i,k} \equiv g^{SK_{i,k}} \pmod{p}. \quad (14)$$

With this keying material, node i creates its k^{th} self-generated certificate $CERT_{i,k}$. This certificate contains the node's identity ID_i, the initial commitment $c_{i,0}$, the new commitment $c_{i,k}$, the new public key $PK_{i,k}$, a timestamp[1] TS (e.g. the day or period that the certificate is valid) and a signature σ. The signature is created from the new private key $SK_{i,k}$ using Schnorr's signature scheme [23]:

1) The node i picks a random secret value $r \in \mathbb{Z}_q^*$ and computes $R \equiv g^r \pmod{p}$.
2) The node i computes the signature $\sigma = (\alpha, \beta)$ in which:
$$\alpha = h(ID_i, c_{i,0}, c_{i,k}, PK_{i,k}, TS, R), \quad (15)$$
$$\beta \equiv r + SK_{i,k} \cdot \alpha \pmod{q}. \quad (16)$$

C. Verification of a Self-Generated Certificate

The k^{th} self-generated certificate of node i is denoted as $CERT_{i,k} = (ID_i, c_{i,0}, c_{i,k}, PK_{i,k}, TS, \sigma)$. This certificate can be distributed and its authenticity verified as follows:

1) The verifier checks the validity of the certificate based on the timestamp TS.
2) The verifier computes the values of $\bar{R}$ and $\bar{\alpha}$:
$$\bar{R} \equiv g^{\beta} \cdot (PK_{i,k})^{-\alpha} \pmod{p}, \quad (17)$$
$$\bar{\alpha} = h(ID_i, c_{i,0}, c_{i,k}, PK_{i,k}, TS, \bar{R}). \quad (18)$$
3) The verifier accepts the signature on the self-generated certificate if $\alpha = \bar{\alpha}$ is correct.
4) The verifier accepts the content of the self-generated certificate if the following equivalancy is correct:
$$PK_{i,k} \equiv c_{i,k} \cdot (c_{i,0} \cdot PK_M^{h(ID_i, c_{i,0})})^{h(ID_i, c_{i,k})} \pmod{p}. \quad (19)$$

[1]The timestamp prevents a malicious node from disrupting communication by replacing a node's current certificate for a stored certificate which the node no longer uses.

D. Establishment of Secure Communication

This key management scheme allows for the establishment of a shared pairwise key $K_{i,j}$ of arbitrary bit-length between node i and j from the exchange of self-generated certificates. This symmetric key can be used to encrypt and authenticate data and is computed as follows:

$$\begin{aligned} K_{i,j} &\equiv h(PK_j^{SK_i} \pmod p)) \equiv h(g^{SK_j \cdot SK_i} \pmod p)) \\ &\equiv h(g^{SK_i \cdot SK_j} \pmod p)) \equiv h(PK_i^{SK_j} \pmod p)) \equiv K_{j,i}. \end{aligned} \tag{20}$$

E. Node Joining

Admission of a new node happens in a distributed fashion. Each network node in possession of its keying material and a share of the master private key is capable of providing key management services. The protocol underneath provides a joining node A with its initial keying material:

1) The node A broadcasts its identifier ID_A, provided by the network operator, to t neighboring nodes. Let Φ denote the set of t neighboring nodes.
2) Each node $i \in \Phi$ interacts with the cellular network to verify the authenticity of node A and its identifier ID_A. Furthermore, the cellular network informs whether A can be provided with a share of the master private key[2]. If the binding between node A and its identifier ID_A is correct, the protocol continues.
3) Each node $i \in \Phi$ picks a random secret value $sv_i \in \mathbb{Z}_q^*$ and computes the partial commitment $c_{i \to A}$:

$$c_{i \to A} \equiv g^{sv_i} \pmod p). \tag{21}$$

4) Each node $i \in \Phi$ transmits $(c_{i \to A}, CERT_i)$ and public network parameters (p, q, g, $h(\cdot)$, PK_M) to node A.
5) The node A verifies the authenticity of each self-generated certificate $CERT_{i \in \Phi}$ and obtains the identifiers $ID_{i \in \Phi}$ of its neighbors.
6) The node A picks a random secret value $sv_A \in \mathbb{Z}_q^*$ and computes its commitment c_A:

$$c_A \equiv g^{sv_A} \cdot \prod_{i \in \Phi} (c_{i \to A})^{\lambda_i^\Phi(0)} \pmod p). \tag{22}$$

The value of $\lambda_i^\Phi(x)$ is the Lagrange coefficient:

$$\lambda_i^\Phi(x) \equiv \prod_{j \in \Phi, j \neq i} \frac{x - ID_j}{ID_i - ID_j} \pmod q). \tag{23}$$

7) The node A chooses a temporary key $K_{A,i \in \Phi}$ and uses the public keys on the received certificates to securely transmit $(c_A, K_{A,i \in \Phi})$ to its neighbors Φ.
8) Each node $i \in \Phi$ computes the partial private key $SK_{i \to A}$:

$$SK_{i \to A} \equiv sv_i + s_i \cdot h(ID_A, c_A) \pmod q). \tag{24}$$

[2]A share of the master private key is provided if the cellular network is convinced that node A will be unable to perform a Sybil attack [24] (e.g. providing node A with a share keeps the total number of shares within A's household below t).

9) Each node $i \in \Phi$ securely transmits $(SK_{i \to A})$ to node A.
10) The node A computes its key pair (SK_A, PK_A):

$$SK_A \equiv sv_A + \sum_{i \in \Phi} (SK_{i \to A} \cdot \lambda_i^\Phi(0)) \pmod q), \tag{25}$$

$$PK_A \equiv g^{SK_A} \pmod p). \tag{26}$$

11) The node A verifies its keying material as follows:

$$PK_A \equiv g^{SK_A} \pmod p) \equiv c_A \cdot PK_M^{h(ID_A, c_A)} \pmod p). \tag{27}$$

If the cellular network allowed node A to obtain a share of the master private key, then the protocol continues:

1) The node A creates its first self-generated certificate $CERT_A$ and transmits $(CERT_A, CERT_{i \in \Phi})$ to its neighbors Φ.
2) Each node $i \in \Phi$ is hereby informed of the neighbors which are providing the key management service for node A. Then, each node $i \in \Phi$ randomly selects the certificate $CERT_j$ of a node $j \in \Phi$.
3) Each node $i \in \Phi$ verifies the authenticity of $CERT_j$ and computes the shared pairwise key $K_{i,j}$.
4) Each node $i \in \Phi$ contacts neighbor j and agrees upon a random shuffling value[3] $\delta_{i,j} \in \mathbb{Z}_q^*$. To cancel out these shuffling values when node A computes its share, and depending on the node identifiers, either node i or j will use the negative value of $\delta_{i,j}$:

$$\delta_{i,j} = \begin{cases} -\delta_{i,j} & ID_i < ID_j, \\ \delta_{i,j} & ID_i > ID_j. \end{cases} \tag{28}$$

5) Each node $i \in \Phi$ obtains at least one shuffling value. We define δ_i to be the sum of the agreed shuffling values obtained by node $i \in \Phi$:

$$\delta_i = \sum \delta_{i,j}. \tag{29}$$

6) Each node $i \in \Phi$ computes the shuffled partial share $s_{i \to A}$ in which $\lambda_i^\Phi(ID_A)$ is the Lagrange coefficient as defined in equation (23):

$$s_{i \to A} \equiv s_i \cdot \lambda_i^\Phi(ID_A) + \delta_i \pmod q). \tag{30}$$

7) Each node $i \in \Phi$ verifies the authenticity of $CERT_A$, computes the shared pairwise key $K_{i,A}$ and securely transmits $(s_{i \to A})$ to node A.
8) The node A computes its share s_A:

$$s_A \equiv \sum_{i \in \Phi} s_{i \to A} \pmod q). \tag{31}$$

9) The node A verifies the correctness of its share s_A using equation (5).

[3]The shuffling value protects the secrecy of every node's share.

F. Share Updating

To prevent a mobile adversary [13] from reconstructing the master private key, we periodically update each node's share to invalidate the shares which have been compromised. As proposed by proactive secret sharing algorithms [11], [12], update polynomials $f_u(x)$ with $f_u(0) = 0$ are periodically created. In the first update period, each node i can compute its new share as follows:

$$s_{i,new} \equiv s_i + f_u(ID_i) \pmod{q} \tag{32}$$

The new share is effectively obtained from the new polynomial $f_{new}(x) = f_M(x) + f_u(x)$ while the master private key is unchanged, since $f_{new}(0) = f_M(0) + f_u(0) = SK_M$. Due to the page limitations, details of this protocol will be provided in a future work.

V. Future work

Future developments of this work include three major sections. First, we plan to expand upon the share updating protocol. Second, the security of the key management scheme will be evaluated based on an adversarial model which is realistic and suitable for NC-MSCs and security proofs will be provided. Finally, the effiency of the key management scheme will be evaluated both analytically and through simulations.

VI. Conclusion

In this paper, we have presented the initial design of the first key management scheme which is specifically designed to establish secure multi-hop D2D communication between users in NC-MSCs. Each of the seven proposed requirements in [7] are achieved: (1) A *high level of security* is achieved through the bootstrapping of a TTP in the network initialization phase and robustness against active adversaries is achieved using verifiable and proactive secret sharing. (2) A *high level of connectivity* is achieved with the exchange and easy verification of self-generated certificates, allowing any arbitrary set of nodes to establish a secure communications channel. (3) The *overhead is low* from both a computational and a communication perspective. The construction of self-generated certificates rely on modular arithmetic while certificates can be renewed independently. (4) The local availability of key management services and the construction of protocols allow for *a scalable network*. (5) *Sustainability is achieved* from a security perspective through share updating, while overhead and connectivity are unaffected by an extended network lifetime. (6) The fully distributed *key management service is fairly distributed* and will minimize nodes from acting selfishly. Finally, (7) this key management scheme *does not rely on secure routing*. Therefore, we can conclude that this key management scheme is suitable and capable of establishing secure communications in NC-MSCs.

References

[1] E. Hossain and M. Hasan, "5G cellular: Key enabling technologies and research challenges," *IEEE Instrum. Meas. Mag.*, vol. 18, no. 3, pp. 11-21, Jun. 2015.

[2] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Commum. Surveys Tuts.*, vol. 16, no. 4, pp. 1801-1819, Nov. 2014.

[3] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612-613, Nov. 1979.

[4] J. Baek, R. Safavi-Naini, and W. Susilo, "Certificateless public key encryption without pairing," in *Proc. 8th Int. Conf. Inf. Secur. (ISC)*, J. Zhou, J. Lopez, R. H. Deng, and F. Bao, Eds. Singapore: Springer, vol. 3650, 2005, pp. 134-148.

[5] J. K. Liu, M. H. Au, and W. Susilo, "Self-generated-certificate public key cryptography and certificateless signature / encryption scheme in the standard model", in *Proc. 2nd ACM Symp. Inf. Comput. Commun. Secur. (ASIACCS)*, Singapore, Mar. 2007, pp. 273-283.

[6] J. Lai and W. Kou, "Self-generated-certificate public key encryption without pairing," in *Proc. Int. Workshop Public Key Cryptogr. (PKC)*, T. Okamoto and X. Wang, Eds. Beijing, China: Springer, vol. 4450, Apr. 2007, pp. 476-489.

[7] M. de Ree *et al.*, "Key management for beyond 5G mobile small cells: A survey," *IEEE Access*, vol. 7, no. 1, pp. 59200-59236, May 2019.

[8] L. Zhou and Z. J. Haas, "Securing ad hoc networks," *IEEE Netw.*, vol. 13, no. 6, pp. 24-30, Nov. 1999.

[9] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," in *Proc. 28th Annu. Symp. Found. Comput. Sci. (SFCS)*, Los Angeles, CA, USA, Oct. 1987, pp. 427-437.

[10] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Proc. CRYPTO*, J. Feigenbaum, Ed. Santa Barbara, CA, USA: Springer, vol. 576, Aug. 1991, pp. 129-140.

[11] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive secret sharing OR: How to cope with perpetual leakage," in *Proc. CRYPTO*, D. Coppersmith, Ed. Santa Barbara, CA, USA: Springer, Aug. 1995, pp. 339-352.

[12] S. Jarecki, "Proactive secret sharing public key cryptosystems," M.S. thesis, Dept. Elec. Eng. Comp. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, Sep. 1995.

[13] R. Ostrovsky and M. Yung, "How to withstand mobile virus attacks," in *Proc. 10th ACM Symp. Princ. Distrib. Comput. (PODC)*, Montreal, QB, Canada, Aug. 1991, pp. 51-59.

[14] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang, "URSA: Ubiquitous and robust access control for mobile ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 6, pp. 1049-1063, Dec. 2004.

[15] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography", in *Proc. ASIACRYPT*, C.-S. Laih, Ed. Taipei, Taiwan: Springer, vol. 2894, Nov. 2003, pp. 452-473.

[16] M. Girault, "Self-certified public keys," in *Proc. EUROCRYPT*, D. W. Davies, Ed. vol. 547, Brighton, U.K.: Springer, pp. 490-497, Apr. 1991.

[17] H. Petersen and P. Horster, "Self-certified keys-Concepts and applications," in *Proc. 3rd IFIP TC6/TC11 Int. Conf. Commun. Multimedia Secur. (CMS)*, S. Katsikas, Ed. Athens, Greece: Springer, vol. 3, Sep. 1997, pp. 102-116.

[18] B. Lee and K. Kim, "Self-certificate: PKI using self-certified key," in *Proc. 3rd Int. Conf. Inf. Secur. Cryptol. (ICISC)*, D. Won, Ed. Seoul, South-Korea: Springer, vol. 10, Dec. 2000, pp. 65-73.

[19] J. Lai, W. Kou, and K. Chen, "Self-generated-certificate public key encryption without pairing and its application," *Information Sciences*, vol. 181, no. 11, pp. 2422-2435, Jun. 2011.

[20] J. Rodriguez *et al.*, "SECRET-Secure network coding for reduced energy next generation mobile small cells: A european training network in wireless communications and networking for 5G," in *Proc. Internet Technol. Appl. (ITA)*, Wrexham, U.K., Sep. 2017, pp. 329-333.

[21] M. de Ree, G. Mantas, A. Radwan, J. Rodriguez, and I. E. Otung, "Key management for secure network coding-enabled mobile small cells," in *Proc. Int. Conf. Broadband Commum., Netw. Syst.*, V. Sucasas, G. Mantas, and S. Althunibat, Eds. Faro, Portugal: Springer, vol. 263, Sep. 2018, pp. 327-336.

[22] S.-F. Chou, T.-C. Chiu, Y.-J. Yu, and A.-C. Pang, "Mobile small cell deployment for next generation cellular networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Austin, TX, USA, Dec. 2014, pp. 4852-4857.

[23] C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, no. 3, pp. 161-174. Jan. 1991.

[24] J. R. Douceur, "The sybil attack," in *Proc. Int. Workshop Peer-to-Peer Syst. (IPTPS)*, P. R. U. Druschel, F. Kaashoek, and A. Rowstron, Eds. Cambridge, MA, USA: Springer, vol. 2429, Oct. 2002, pp. 251-260.