Security Establishment for IoT Environments in 5G: Direct MTC-UE Communications

Filipe Conceição[1,2], Nouha Oualha[1], and Djamal Zeghlache[2]
[1] CEA, LIST, Communicating Systems Laboratory, 91191 Gif-sur-yvette Cedex, France
[2] Télécom SudParis, Institut Mines-Télécom, 9 rue Charles Fourier, 91011 Evry, France
Email: filipe.conceicao@cea.fr, nouha.oualha@cea.fr, djamal.zeghlache@telecom-sudparis.eu

***Abstract*—In this paper, we propose a new protocol for the key establishment without any prior trust relationship between an UE and a Machine Type Communication (MTC) device, building security from 5G towards the IoT. We use the protocol with ProSe to increase 5G coverage and propose a new secure operating mode that has the potential to save energy and bandwidth by securing direct sessions between UEs and MTC devices.**

I. Introduction

In [1], a projection related to 5G and the Internet of Things (IoT) predicts 29 billion connected devices by 2022. Fom this number, 18 billion will be related to the IoT. Connected devices include cars, machines, meters, sensors, point-of-sales terminals, consumer electronics and wearables. By the same year, a worldwide total of 6.2 billion (all different) mobile subscribers will hold a total of 9 billion subscriptions. With these predictions for IoT devices and subscribers, the connections between devices is expected to increase within IoT networks or in interaction with other types of equipment.

For direct connections between User Equipments (UEs), 3GPP has standardized Device-to-Device (D2D) communications naming them Proximity Services (ProSe) [2]. For Machine Type Communications (MTC), it has released the recommendations for security mechanisms [3]. In the MTC category, the architectural model consists of a client, the MTC device (MTCd), and a MTC Server (MTCs) that is responsible for the security of a group of MTCd. The MTCs can also store particular information sent from each MTCd under its control [3]. This operational mode doesn't account for volatile data or actions that don't need to be recorded and increases, in a general way, the latency. It compels a user who needs information from a group of MTCd or wants to interact with them, to run a security procedure with MTCs to get needed data, through a Base Station (BS). MTCd are expected to authenticate to a MTCs and send their data or receive commands from it. In parallel, a user carrying a UE and authorized to interact with certain MTCd, also needs to authenticate to MTCs. After this procedure is complete, the interaction between MTCd and UEs runs through the MTCs, rather than directly, in a D2D fashion. The MTCs is a participant in the user plane (UP) data flow, which adds energy and bandwidth consumption. As an example, using the simplified path-loss model [4] with a reference distance $d_0 = 10m$, constant $L = 4.38 \times 10^{-8}$ and $\gamma = 3$, and comparing losses for communication distances, e.g., $20m$ for D2D and $300m$ for a cellular link, we see that a D2D link has a loss roughly 3425 times inferior than a cellular link.

978-1-5386-3531-5/17/$31.00 © 2017 IEEE

In the 3GPP architecture the MTC UP data is required to flow through a server beyond the BS to reach back to an UE. There is always a direct connection between the MTCd or a gateway (GW) to a BS, without room for cooperation schemes that could allow for reduced power transmissions and bigger coverage area. If we equate mobility of the MTCd, as in any moving vehicles, devices installed in moving parts or even wearables, we see that MTCd communicating directly with a BS can have a very high cost in terms of power. Some devices will simply suffer from power depletion. However, if they could directly connect to another device for their routine interactions, the power saved could be significant.

Therefore we see an opportunity to shorten communication distance, using the potential of the ProSe functionality. We look at the numbers estimated for the IoT, mobile subscriptions and scenarios in smart cities and Public Protection and Disaster Relief (PPDR) use cases and foresee a bigger number of interactions UE-MTCd, many times higher than the number of deployed devices. These connections need to be secure, even if just to guarantee the integrity of the messages exchanged. Therefore the number of end-to-end pairwise keys is, regardless of the technique used, bigger than the number of users interacting with other devices.

Therefore, in this work, we focus on the key establishment for connections between UEs and MTCd and aim at reducing the communication costs of these connections by taking advantage of proximity and the ProSe standard (that cannot be used with resource constrained devices), and present a lightweight key distribution scheme. We account for service authorization and authentication of all participating devices, MTCd, UEs or GWs. Specifically, we propose a protocol for mutual authentication of a MTCd and a MTCs using an UE as a relay. At the same time, the MTCd and the UE establish a symmetric master key. This allows them to communicate directly, making it possible for the MTCs not to participate in the UP data flow. We present also a cooperation scheme based on the proposed protocol that extends 5G coverage towards MTCd. The main contributions of this paper are: 1) we provide a method to distribute a shared secret between each MTCd and an UE that wishes to communicate with them. The key pair is symmetric, respecting the resource constrained nature of MTCd. 2) Our method provides authentication and authorization services of all participating devices. 3) The proposed solution is able to resist to known attacks. We used the automatic protocol formal verification tool ProVerif to prove our protocol's security. 4) Our solution limits the communication range of MTCd or a GW to an UE in proximity, rather than a BS. It also removes the MTCs from the UP data flow to save energy in the MTCd, the UEs and in the overall 5G network.

Section II presents related work. Section III describes our system model. Our solution an its extension are reported in section IV. Security and performance evaluation are discussed in section VI.

II. Related work

The coverage extension possibilities opened by the ProSe standard are not well explored. ProSe offers a way to offload some communications away from the BSs. As for UEs, the standard is clear now but MTC communications are still not well defined and have room for improvements. Although there are many solutions for resource constrained devices for key establishment in IoT environments [5], to the best of our knowledge, they do not involve the ProSe standard in 5G. The cooperation schemes for coverage extension in 5G are not abundant either. The works mentioned in this section all relate to 5G, establishing D2D security or authenticating MTCd to the Core Network. However, they all differ from our solution, the only one providing security establishment for UE and MTCd direct communications. Authors of [6] propose a protocol for coverage extension where UEs in coverage of a BS serve as anchors to MTCd to send their data. A set of key indexes is advertised by UEs to devices outside coverage expecting they share at least one key with the sender. If there is no shared key, connection is not established. For this, we define the scheme as a probabilistic key establishment scheme. Reference [7] proposes a cooperation scheme that allows for coverage extension based on a coalition of UEs that cooperate and decide whether to accept or not to start a direct link connection with another device. This proposal relies on certificates and an asymmetric cryptosystem, generally considered computationally heavy for resource constrained MTCd. Work reported in [8] addresses authenticating MTCd towards MTCs and provides mutual authentication between MTCd and MTCs. However, this solution does not consider the possibility of expanding coverage or direct communication with the MTCd. It needs 6 messages to authenticate devices with LTE radio capabilities, requires grouping the MTCd together, sharing a group key.

III. System model

We now describe the considered system model. The participants involved are UEs, MTCd, GWs and MTCs, whether it is part of the 3GPP network or not [3]. Fig. 1 shows the considered network architecture.

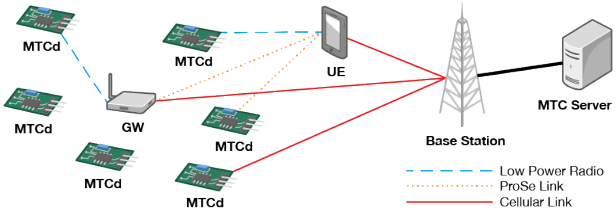

Fig. 1: System Model

We consider as an UE any device that has, amongst others, 5G and LTE radio interfaces and is usually held and controlled by a person. MTCd are generally devices with low power radio technologies like the ones based on IEEE 802.15.4, Wi-Fi, Bluetooth, etc., and may have 5G and LTE radio interfaces. Regardless of their embedded radio interfaces, we consider them to be resource constrained in terms of computational capabilities, memory space and battery capacity and to be equipped with Machine Type Communication capabilities [3]. GWs are devices that can serve as a radio GW for a cluster of MTCd in their vicinity. The GW role is to receive information from MTCd, and send the information through a more powerful, LTE or 5G radio channel, to a BS. The MTCs is the element that is assigned a number of MTCd, and that is responsible for the security of these devices and/or storing their data.

IV. Our proposed protocol

We describe our protocol, the messages and their content, as well as our solution security features and use the notation shown in table I. We propose to use our protocol, that has initialization and key exchange phases, for an IoT coverage extension, using the ProSe standard.

TABLE I: NOTATION

Abbreviation	Definition
PSKey	Pre shared key
DMKey	Derived Master key
MIC	Message Integrity Code
MTCdMIC	MIC calculated by MTC device
MTCsMIC	MIC calculated by MTC Server
MTCdID	MTC device's ID
MTCsID	MTC server's ID
UEID	UE ID
GWID	GW ID
MTCdNonce	Nonce generated by MTC device
MTCsNonce	Nonce generated by MTC server
KDF	Key Derivation Function
MTCdInfo	Information used by the device
MTCsInfo	Information used by the server
$\|$	Concatenation

A. Initialization phase

The initialization phase consists of pre-determined conditions representing our assumptions for the protocol to run properly. Namely, we consider the following:

- Each MTCd has at least one MTCs responsible for security material distribution, authentication and authorization of the MTCd assigned to it;
- Each MTCd has a pre shared secret key that is only known to itself and to the MTCs it is assigned to;
- Each MTCd is assigned with an unique ID inside its own MTCs cluster and knows its MTCs IDs;
- The UEs have access through a secure channel to the MTCs under the 3GPP system responsibility;
- UEs communicating directly or with a GW, communicate using a private channel established according to ProSe [9].

We also assume that an MTCd can start communication with another device and indicate that its radio channel evaluation is out of the scope of this work.

B. Key exchange phase

In the key exchange phase, the protocol is simply composed of 4 messages and is represented in Fig. 2.

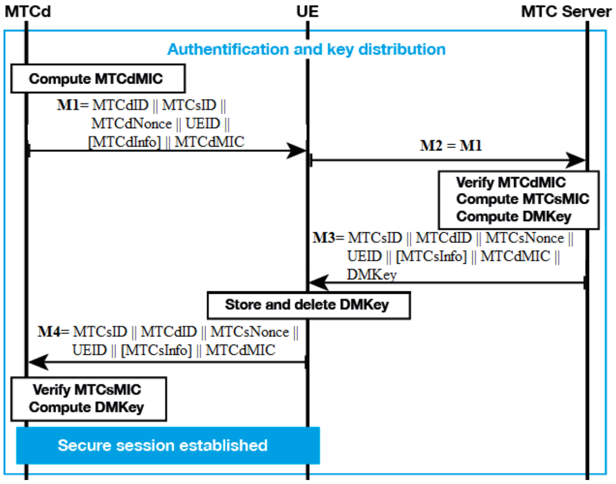
MTCd
UE
MTC Server
Authentification and key distribution
Compute MTCdMIC
M1= MTCdID || MTCsID || MTCdNonce || UEID || [MTCdInfo] || MTCdMIC
M2 = M1
Verify MTCdMIC
Compute MTCsMIC
Compute DMKey
M3= MTCsID || MTCdID || MTCsNonce || UEID || [MTCsInfo] || MTCdMIC || DMKey
Store and delete DMKey
M4= MTCsID || MTCdID || MTCsNonce || UEID || [MTCsInfo] || MTCdMIC
Verify MTCsMIC
Compute DMKey
Secure session established

Fig. 2: Authentication protocol

We describe the four messages in more detail:
Message 1: MTCd generates MTCdNonce (for freshness of the message) and uses it with PSKey, MTCdInfo and the IDs of the participants in the protocol to calculate MTCdMIC keyed with PSKey. In this way, the IDs of all participants in the routing path are binded to mitigate the risk of spoofing attacks.
$MTCdMIC = MIC(PSKey, MTCdID, MTCsID, UEID, MTCdNonce, MTCdInfo)$
MTCdInfo may contain information about the UE/GW ID connecting to MTCd, contextual information (e.g., location) or any other that the MTCd needs to send to MTCs, as for example related to the cryptographic algorithms supported (e.g., MIC algorithm, KDF function) or its current status (e.g., battery level). We do not advocate any particular algorithm for MTCdMIC calculation although we would follow the recommended in [10], [11]. MTCd then computes and sends to the UE:
$M1 = MTCdID||MTCsID||MTCdNonce||UEID|| [MTCdInfo]||MTCdMIC$
Message 2: When the UE receives **M1**, it simply forwards it to the MTCs with identity MTCsID.
Message 3: Upon reception of **M2**, MTCs can check the UE's service authorization with the Core Network. Therefore, mutual authentication between MTCs and UE can take place (e.g., using TLS/SSL). Then, it verifies MTCdMIC using the PSKey and the elements in **M2**. If the challenge was correctly answered by the MTCd, its authentication in MTCs is completed. It then generates MTCsNonce and calculates its own MIC:
$MTCsMIC = MIC(MTCdID, MTCsID, MTCdNonce, MTCsNonce, UEID, MTCdInfo, MTCsInfo, PSKey)$
MTCsNonce and IDs are used again for the novelty of M3 and to bind IDs. MTCs then generates DMKey:
$DMKey = KDF(MTCdID, MTCsID, MTCdNonce, MTCsNonce, UEID, PSKey)$
MTCs then computes and sends to the UE:
$M3 = MTCsID||MTCdID||MTCsNonce|| UEID||[MTCsInfo]||MTCsMIC||DMKey$
MTCsInfo can be useful if the MTCs wants to select a particular KDF, or to limit the actions of a user towards the specific MTCd in terms of usage type, duration and DMKey expiration/revocation. We leave it as an open topic for MTCs policy. We do not advocate the use of any specific KDF. The MTCs should be able to select the best option for the related MTCd. However, we advocate the KDF recommendations in [12].
Message 4: When the UE receives **M3** from MTCs it extracts, stores and deletes DMKey from the message. By receiving DMKey, the UE has the implicit indication that the MTCd has been successfully authenticated. It then sends to the MTCd:
$M4 = MTCsID||MTCdID||MTCsNonce||UEID ||[MTCsInfo]||MTCsMIC$
Upon reception, the MTCd uses these elements to verify MTCsMIC. If the verification is successful, the MTCd authenticates the MTCs and the mutual authentication process is complete. It then computes DMkey:
$DMKey = KDF(MTCdID, MTCsID, MTCdNonce, MTCsNonce, UEID, PSKey)$
After the protocol is executed, the MTCd also implicitly authenticates UEs as they now both share a DMKey, a shared secret key that can be used to derive further confidentiality or integrity protection keys. This solution can act as the underlying mechanism for ProSe, using the methods described in [9] to derive further keys.

C. Protocol extension

We now consider in our scenario the addition of another UE in proximity of a group of MTCd or a GW. A user needing to connect to one or more MTCd can request the connection establishment to their ProSe links, UEs or GWs. If it doesn't have any ProSe pair, ProSe discovery request can start as defined in [2]. We advocate that ProSe Direct Discovery and Direct Communication concepts [2] can be used between an UE and a GW for cooperation and coverage extension. After this connection is established, the UE/GW try to access the MTCd the user requested. If so, the protocol proposed in the previous section is executed, with one more actor, the second UE or a GW. Fig. 3 ilustrates the extended version of the protocol. In addition to the messages defined and explained in section IV, we now add two more messages, **M2** and **M6**. They are however the same as described before, only the new UE or GW is forwarding them to the correct destination. This is a way to use the ProSe standard to leverage the coverage of the 5G network.

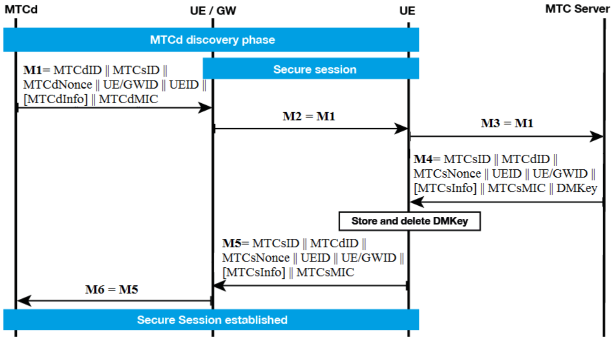
MTCd
UE / GW
UE
MTC Server
MTCd discovery phase
Secure session
M1= MTCdID || MTCsID || MTCdNonce || UE/GWID || UEID || [MTCdInfo] || MTCdMIC
M2 = M1
M3 = M1
M4= MTCsID || MTCdID || MTCsNonce || UEID || UE/GWID || [MTCsInfo] || MTCsMIC || DMKey
Store and delete DMKey
M5= MTCsID || MTCdID || MTCsNonce || UEID || UE/GWID || [MTCsInfo] || MTCsMIC
M6 = M5
Secure Session established

Fig. 3: Extended authentication protocol

V. Security evaluation and analysis

In this section we evaluate the security properties of our solution. Our security analysis is built based on Dolev-Yao threat model. Our protocol is fully compliant with all the security requirements of 3GPP [9]. A trust relation is built between all elements as the protocol runs.

ProVerif [13] was used to test the authentication and secrecy properties of our protocol. After representing it with this tool, we queried the secrecy of DMkey, the authentication of the MTCd by the MTCs and vice versa. We obtained positive results for all 3 queries. We proved therefore, the secrecy and mutual authentication properties of our solution. Our metrics for security evaluation are symmetry of the keys, if the scheme is probabilistic or deterministic and if it is server assisted. Confidentiality and integrity of the messages, authentication, authorization, freshness of messages and resilience to attacks are also evaluated. Our solution uses a symmetric key scheme to account for limitations of MTCd. It is deterministic by design and relies on the MTCs to assist the key establishment. Messages exchanged can have confidentiality and integrity by means of the shared DMKey. The authentication is mutual and explicit for the MTCd-MTCs pair by verification of the MICs. It is mutual for the pairs UE-MTCs. They trust each other as their secure channel was previously established by the 3GPP system. It is mutual and implicit for MTCd-UE pair, the moment their symmetric keys are established, because when MTCd verifies MTCsMIC, it means the MTCs trusted the UE. Authorization can be checked for all participants. MTCs is responsible for authorizing MTCd as per its own policy and to check with the Core Network if the UEs and GWs are authorized to establish the connections. MTCdNonce and MTCsNonce mitigate replay attacks. Finally, our solution is resilient to attacks due to the use of pairwise keys so compromising one node does not compromise the whole network.

VI. Performance

We now evaluate security metrics and performance of our solution and compare them with works [6], [7], [8]. We evaluate works as deterministic or probabilistic for key establishment procedures. In [6], a probabilistic scheme is proposed and probabilities for key establishment are presented. The authentication is via a coalition of UEs in [7] and in both [7] and [6] servers are not involved. In [6], the server is also not required for communication establishment. In this sense, we describe these proposals as providing implicit authentication. In both our solution and in [8], there is support for authorization and explicit mutual authentication between MTCd and MTCs. In our proposal, MTCdInfo and MTCsInfo are reserved to use to limit the access for certain data or application type, access time, or any other information that suits their needs or policies. Finally, we show the Server is a necessary actor in our proposal as in [8] so that the MTCd are authenticated and the protocols executed. Table II summarizes the qualitative comparison assessment of the security features.

TABLE II: SECURITY METRICS COMPARISON

	Key establishment	Authentication	Authorisation	Server assistance
[7]	Deterministic	Coalition of devices implicitly autheticates the new member.	Not specified	Not needed
[6]	Probabilistic	Implicit authentication by having a common key	Not specified	Not needed
[8]	Deterministic	Explicit and mutual for MTCd-GW	Supported	Needed
Our	Deterministic	Explicit and mutual for MTCd-MTCs Implicit and mutual for MTCd-GW	Supported	Needed

We evaluate performance in terms of number of messages necessary for key distribution, computational effort required to run it and memory requirements. Our proposal needs four messages to be executed. The cooperation scheme adds two more messages, but they are simply forwarded from one UE to another UE/GW, without extra computations. The MICs and nonces provide explicit mutual authentication and mitigate replay attacks. MTCsInfo allows MTCs to be able to choose a suitable KDF and to restrict the usage of the MTCd, as per its policy. Therefore we conclude that the elements in the messages are the minimum possible to guarantee these security properties. Symmetric key cryptosystems are well suited for resource constrained MTCd. The KDF executed in the server side and the key delivered to the UE, eliminates the need for the latter to make computations. The MICs are calculated in a standard, recommended way [10], [11] and therefore, the computational cost is normal. The MTCs can choose the KDF from the recommended ones in MTCsInfo. This can be very useful for the MTCd as the MTCs can, for example, select a suitable KDF to generate DMKey. Therefore, we evaluate the computational costs and computing power as minimum to garantee robust security features.

As for memory, some bytes are needed to store keys. The nonces require some more bytes to store previously used values but in very constrained MTCd, they can be replaced by counters. The MTCs needs to maintain a database linking MTCd IDs with the PSKey, DMKey and nonces but memory shouldn't be a problem at a server level. The power consumed in communications can be reduced after the D2D connection is established. It can reduce congestion risk if the interaction is with several MTCd at the same time. The scalability may be affected for a big number of MTCd but as the protocol relies on proximity, this is not foreseen as a problem.

We compare our proposal with the works mentioned in II. We evaluate the performance in terms of the main energy spending contributors: CPU usage, memory usage and numbers of messages Tx/Rx. Our proposal requires less messages exchanged, provides all the modern security features and complies with 3GPP standards [9]. To better demonstrate the benefits for MTCd in power savings, both after the D2D connection is built and during its establishment, we look at these metrics from the point of view of the MTCd. It is worth to mention that, to the best of our knowledge, there is no other proposal for direct MTCd-UE communications. In [8], a protocol to authenticate a group of MTCd is proposed. To make a proper comparison, we assume their protocol authenticating one MTCd. Table III summarizes the comparison of the four proposals in terms of the mentioned performance.

TABLE III: PERFORMANCE COMPARISON

	CPU usage	Memory usage	Messages Tx/Rx
[7]	**High** Assymetric cryptosystem	**High** Use of certificates	Messages exchanged on demand: minimum 3
[6]	**Low** Symmetric cryptosystem (Computes key verification)	**High** use of key rings (higher the number of keys, higher the probability)	Messages sent periodically: every 10ms
[8]	**Low** Symmetric cryptosystem (Computes MICs and KDFs)	**Low**: 1 PSKey, 1 Group Key, 1 Derived Key (per pair MTCd-server)	Messages exchanged on demand: 4
Our	**Low** Symmetric cryptosystem (Computes MICs and KDFs)	**Low**: 1 PSKey, 1 Derived Key (per pair MTCd-UE)	Messages exchanged on demand: 2

VII. Conclusions

In this paper we present a protocol for authentication and establishment of secure sessions between UEs and MTC devices without any prior trust. We rely on cryptographic systems that respect the nature of resource constrained MTCd and guarantee important security features. Our proposal eliminates the need of MTCd or GWs sending data to a server, saving significant amounts of energy, bandwidth and reducing latency. To the best of our knowledge, this is the first solution for this direct interaction in 5G and IoT. We introduce the ProSe standard to enhance the coverage of 5G by means of interaction between two UEs, or one UE and a GW and prove the security of our solution. We advocate this operating mode since it is extremely efficient for coverage extension, saves energy and bandwidth. Our solution is relevant and useful for PPDR scenarios, where UEs may play a key role in maintaining communications, and smart city scenarios where volatile type interactions often take place.

Acknowledgements

This work received funding from the European Unions Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 675891 (SCAVENGE).

References

[1] Ericsson. (2017, June). *Ericsson Mobility Report* [Online]. Available: https://www.ericsson.com/en/mobility-report

[2] *3rd Generation Partnership Project, Proximity-based services (ProSe), Stage 2 (Release 13)*, 3GPP TS 23.303, 2017

[3] *3rd Generation Partnership Project, Service requirements for Machine-Type Communications (MTC), Stage 1 (Release 14)*, 3GPP TS 22.368, 2017

[4] A. Goldsmith, "Cellular systems and infrastructure-based wireless networks", in *Wireless Communications*. New York: Cambridge University Press, 2005, ch 15.3, pp. 514

[5] K. T. Nguyen, N. Oualha, M. Laurent, "Survey on Secure Communication Protocols for the Internet of Things." *Ad Hoc Networks* 32:1731, 2015

[6] G. Steri, G. Baldini, I. N. Fovino, R. Neisse and L. Goratti, "A novel multi-hop secure LTE-D2D communication protocol for IoT scenarios," *2016 23rd International Conference on Telecommunications (ICT)*, Thessaloniki, 2016, pp. 1-6.

[7] A. Ometov, K. Zhidanov, S. Bezzateev, R. Florea, S. Andreev and Y. Koucheryavy, "Securing Network-Assisted Direct Communication: The Case of Unreliable Cellular Connectivity," *2015 IEEE Trustcom/BigDataSE/ISPA*, Helsinki, 2015, pp. 826-833.

[8] C. Lai, H. Li, R. Lu, Rong Jiang and X. Shen, "LGTH: A lightweight group authentication protocol for machine-type communication in LTE networks," *2013 IEEE Global Communications Conference (GLOBECOM)*, Atlanta, GA, 2013, pp. 832-837.

[9] *3rd Generation Partnership Project, Proximity-based Services (ProSe), Security aspects (Release 14), Stage 2 (Release 13)*, 3GPP TS 33.303, 2017

[10] *FIPS 198-1, The Keyed-Hash Message Authentication Code (HMAC)*, 2008.

[11] *NIST SP 800-38B, Recommendation for Block Cipher Modes of Operation The CMAC Mode for Authentication*, 2005.

[12] *NIST SP 800-108, Recommendation for Key Derivation Using Pseudorandom Functions*, 2009

[13] B. Blanchet, "Automatic verification of correspondences for security protocols," *Journal of Computer Security*, 2009, 17(4):363434.