Delivering Multicast Content Through Secure D2D Communications in the Internet of Things

Chiara Suraci, Sara Pizzi, Antonio Iera, Antonella Molinaro, Giuseppe Araniti

Delivering Multicast Content through Secure D2D Communications in the Internet of Things

C. Suraci, S. Pizzi, A. Iera, A. Molinaro, G. Araniti

DIIES, University "Mediterranea" of Reggio Calabria
Via Graziella, Loc. Feo di Vito, 89100, Reggio Calabria, Italy
Email: [name.surname]@unirc.it

Abstract. Device-to-device (D2D) communications and cellular solutions represent key technologies for the development of a future infrastructure in which fifth-generation (5G) systems and the Internet of Things (IoT) will converge. A tricky issue to be carefully investigated in D2D communications is the prevention of threats to privacy and security caused by malicious devices. Thus, security mechanisms must be implemented in order to assure a reliable data exchange among involved devices. In this paper, we propose MtMS-sD2D (Machine-type Multicast Service with secure D2D), an architecture for the delivery of multicast traffic in an IoT scenario that takes advantage from D2D communications made reliable by means of a security mechanism. Furthermore, we discuss the procedures that must be followed to efficiently deliver multicast traffic. Finally, we provide some preliminary yet insightful simulation results.

Keywords: 5G · IoT · D2D · MtMS · Security.

1 Introduction

Device-to-device (D2D) communication appears as a promising paradigm to support the interconnection of heterogeneous objects foreseen by the Internet of Things (IoT) [1]. An IoT system can be seen as a collection of smart devices that interact (by means of different processing and communication architectures, technologies, and design methodologies) on a collaborative basis to fulfill a common goal. The amount of interconnected devices is expected to grow heavily in the near future not only due to the increase in possible use cases, but also to the decrease in device costs and to the widespread use of technologies augmenting device capabilities at minimal costs. Thus, IoT is considered one of the key enabling technologies for the next-to-come fifth-generation (5G) cellular networks [2].

As a consequence, a severe traffic overload problem will have to be faced in cellular networks. D2D communications represent a promising method for data offloading, spectrum efficiency enhancement, network resource utilization improvement, user's throughput improvement, and battery lifetime extension, thanks to its capability of leveraging the proximity between the devices involved

in the direct communication. In literature, many works deal with the D2D technology. In [3], a taxonomy for D2D communications, based on the spectrum utilization, is presented.

The wide scale of IoT inherently magnifies the risk of security threats of the current Internet. In addition, despite the obvious advantages of D2D communications, their intrinsic nature causes also additional problems due to the fact that connections happen directly between devices in proximity. Thus, careful investigations are required on solutions finalized to mitigate the risk of threats to privacy and security caused by malicious devices when exploiting D2D communications. Evidently, any malicious behavior by devices may cause serious consequence and lead to deteriorated user experience.

Actually, an always increasing number of autonomously operated, low-cost devices (i.e. sensors, actuators, drones) requires to be connected with each other by exploiting wireless networks no longer exclusively dedicated to human communications. Not by chance, the support of a new type of traffic known as machine-type communications (MTC) has become one of the key objective of 5G networks. This new communication paradigm could benefit from group-oriented (e.g., multicast) services to achieve a reduction in energy consumption and delay. In particular, multicasting can be an effective means for simultaneously sending data to a group of IoT devices through point-to-multipoint communications [4].

Multimedia Broadcast Multicast Services (MBMS) is the architecture that the 3rd Generation Partnership Project (3GPP) has standardized to manage the delivery of multicast and broadcast services over cellular networks. The main nodes of this architecture are: the *broadcast multicast-service center (BM-SC)*, which is responsible for the initialization of the MBMS session and for some security functions, such as the management of the authorizations for the MBMS subscribers; the *MBMS-gateway (MBMS-GW)*, which is in charge of forwarding the MBMS packets to the BSs involved in the delivery service; the *multi-cell/multicast coordination entity (MCE)*, which has to manage the admission control and the radio resource allocation to every BS.

All these nodes need to be enhanced in order to enable multicasting also over future 5G networks. MBMS also specifies the procedures to follow in order to sustain a multicast session. The legacy MBMS is highly suitable to support multimedia applications, but many changes have to be applied to the standardized architecture and procedures in order to manage future MTC multicast traffic more properly. In this regard, in [5], the Machine-type Multicast Service (MtMS) is proposed to define the architecture and the most adequate transmission procedures to the management of the MTC multicast traffic.

With reference to this research area, this paper introduces a network architecture, MtMS-sD2D (MtMS with secure D2D), specifically designed for the highly reliable delivery of multicast traffic to a set of machines in IoT scenarios, which leverages D2D communications coupled to a secure mechanism based on the Diffie-Hellman key exchange (DHKE) protocol. Furthermore, it discusses the designed procedures introduced to efficiently transmit multicast data toward a set of MTC devices.

The remainder of the paper is organized as follows. In section 2, the security mechanism implemented in the D2D communication is presented. Section 3 and section 4 describe, respectively, the reference network architecture and the transmission procedures of the proposed MtMS-sD2D protocol. Obtained results are shown in section 4. Conclusive remarks are given in the last section.

2 Securing D2D Communications

In [2], among the listed 5G requirements, improved security mechanisms are recommended that can work effectively in the presence of a likely huge increase in the amount of data transmitted over cellular networks. In the IoT landscape, machines often communicate sensitive data over the unsecure wireless channel; thus security and privacy requirements have to be satisfied to guarantee both data and devices protection. Among security requirements, data confidentiality and integrity, authentication, and authorization are highly relevant to the IoT scenario. As for privacy requirements, it is important to protect personal devices information [6].

This work considers an IoT scenario, wherein the MTC multicast traffic is managed through an enhanced version of the MtMS presented in [5], named *MtMS with secure D2D (MtMS-sD2D)*. D2D communications are established between devices directly served by the BS and those terminals excluded from the multicast transmission, because of their adverse channel conditions. A secure protocol is implemented over D2D communication in order to protect the transmitted data.

A secure data sharing strategy for D2D communication is presented in [7]. This protocol satisfies many security requirements, such as non-repudiation, authentication, authorization, confidentiality, and integrity. It uses some security mechanisms, such as encryption, HMAC, and signature to manage the messages exchanged between the two peers involved in direct communication. To this aim, the encryption of transmitted data is performed by using a symmetric (i.e., private-key) encryption algorithm; this means that the same key is used to encrypt and to decrypt data. The private key is generated through an enhanced version of the *Diffie-Hellman key exchange (DHKE)* protocol, in which the public keys exchange is intermediated by the trusted third party, represented by the BS. The effectiveness of the DHKE algorithm relies on the challenge of computing logarithms over a finite field *GF(q)* with a prime number q of elements.

As in [7], we exploit the DHKE protocol in order to secure data transmission on D2D communications. In the following, the basics of the DHKE protocol will be given.

Let

$$Y = \alpha^X \bmod q, \qquad \text{for } 1 \leq X \leq q-1, \tag{1}$$

where α is a fixed primitive element of $GF(q)$ known to both users involved in the D2D communications, then X is referred to as the logarithm of Y to the base α, mod q:

$$X = \log_{\alpha} Y \bmod q, \qquad \text{for } 1 \leq Y \leq q-1. \tag{2}$$

While computing Y from X is easy, the calculation of X from Y can be much more difficult and it could require a number of operations in the order of $q^{1/2}$, using the best known algorithm. For this reason, it is necessary to choose a number q consisting of at least 300 digits for the system not to be broken. The security of the technique crucially depends on the difficulty of computing logarithms mod q. When users i and j want to communicate privately, first of all they must agree on the values of q and α. Then, each user generates an independent random number X_i chosen uniformly from the set of integers $\{1, 2, 3, ..., q-1\}$ and keeps it secret, but sends to the other user

$$Y_i = \alpha^{X_i} \bmod q. \tag{3}$$

The key used for both enciphering and deciphering by the two users is

$$K_{ij} = \alpha^{X_i X_j} \bmod q. \tag{4}$$

User i obtains K_{ij} by obtaining Y_j from user j and letting

$$K_{ij} = Y_j^{Xi} \bmod q = (\alpha^{X_j})^{Xi} \bmod q = \alpha^{X_i X_j} \bmod q. \tag{5}$$

User j obtains K_{ij} in the similar way

$$K_{ij} = Y_i^{X_j} \bmod q. \tag{6}$$

For an untrusted third party it is impossible to generate the same key K_{ij}, since it can not know X_i and X_j in any way because they are kept secret by users [8]. In addition to the DHKE algorithm, the use of message authentication helps to avoid the man-in-the-middle attack, which represents the main vulnerability of the DHKE algorithm. In this way, many attacks to the D2D communication can be avoided. Among these, eavesdropping, impersonation and masquerading, and the already cited man-in-the-middle.

3 Reference Network Architecture

The reference scenario for this paper consists of a generic IoT environment where devices receive contents from the network in a multicast delivery modality. Examples of applicative use-cases that could benefit from such a scenario are: (i) software update of a group of machines owned by a customer/tenant, or (ii) delivery of alerting messages.

MtMS has already defined an architecture and procedures to adapt the standardized MBMS to MTC traffic. In order to improve the efficiency of the entire process, devices with the worst channel conditions are excluded from the multicast transmission and are subsequently served via D2D communications by devices which directly received data from the BS. Furthermore, a security protocol is used to protect data transmitted in D2D communications.

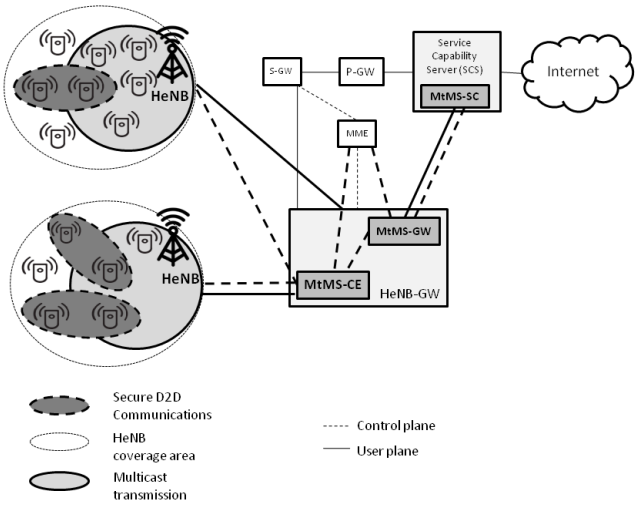

Fig. 1. MtMS-sD2D architecture.

Our MtMS-sD2D architecture, depicted in Fig. 1, derives from the MtMS architecture and properly enhances it to support secure D2D communications. It is composed of the following nodes: *home-evolved NodeB (HeNB)*, also known as femto-cell, which provides connectivity to a small-cell of devices, thus guaranteeing latency and energy consumption reductions and improving coverage and reliability compared to the traditional macro-cell; *HeNB gateway (HeNB-GW)*, which aggregates control and data traffic of various HeNBs; *MtMS serving center (MtMS-SC)*, implemented at the service capability server (SCS), which is responsible for initializing the MtMS session, obtaining the multicast content and the information about the receiving devices; *MtMS coordination entity (MtMS-CE)*, which manages the joining procedure, paging the indicated devices; *MtMS gateway (MtMS-GW)*, which receives data from the MtMS-SC and forwards it to the cells with paged devices. MtMS-GW and MtMS-CE are implemented at the HeNB-GW [5]. In the new MtMS-sD2D protocol, the MtMS procedures (i.e., initialization, joining, and content delivery) are modified to support a more efficient and secure transmission of data.

4 Machine-type multicast service with secure D2D

In the generic IoT environment with MTC data traffic under analysis, three main segments compose the end-to-end path: uplink (UL), core network, and downlink (DL). The *UL* segment includes the random access (RA) procedure and the subsequent transmission of sensed data by the involved devices toward the network. Devices requires the *RA procedure* to retrieve the synchronization

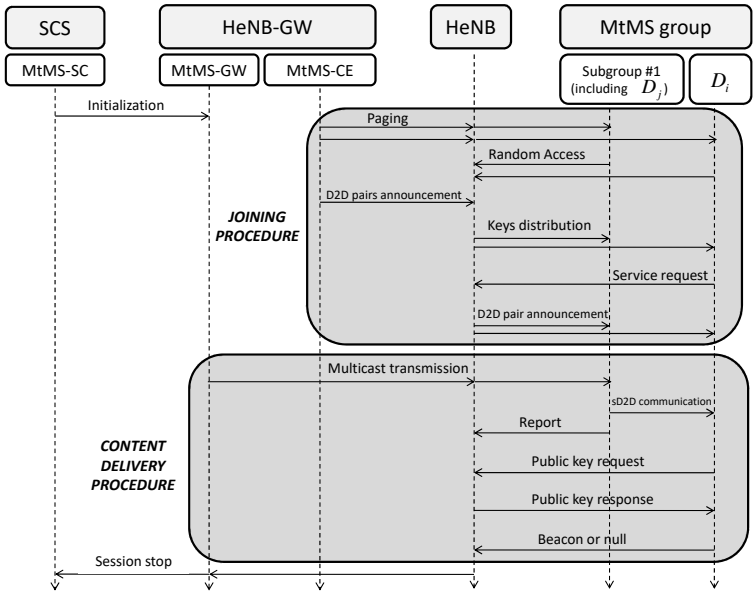

Fig. 2. MtMS-sD2D session.

with the BS. Indeed, in an IoT environment devices can switch to an idle state, during which they turn off the radio interface to save energy. After the RA procedure is accomplished, they can send sensed data. IoT devices can collect data on demand when they are triggered by the network (e.g., in the case of emergency notices) or periodically, as in the case of sensors that monitor environmental pollution. During the RA procedure devices also send to the BS information about their channel conditions (i.e., CQI values). Each device has to send not only information about the condition of the channel that connects it to the BS, but also about the channels (i.e., D2D links) that connect it to its neighbors. Thanks to these information, the BS can evaluate which nodes shall be excluded from the multicast transmission, because of their bad channel conditions, and shall be reached through the establishment of D2D connections. The BS chooses the set of nodes to be served in multicast and those to be served in D2D through an iterative procedure. It analyzes the different possible CQI values in ascending order. For each considered CQI value, the BS determines how many devices, among those in the cell, can receive and decode the data sent through the Conventional Multicast Scheme (CMS) with the Modulation and Coding Scheme (MCS) related to the considered CQI. Devices that fail to receive data, since their CQI level towards the BS is lower than the considered one, must be served in D2D. If the BS can find a transmitter for each D2D receiver, an eligible configuration is created. The iterative procedure ends when

the BS analyzes a CQI value which does not allow the reception of the data sent in CMS to any device in the cell. Among all eligible configurations, the BS selects the one that allows the maximization of transmission performance.

Data sent by devices are processed in the *core network*.

In the *DL* segment, the MtMS-sD2D session is accomplished. Fig. 2 depicts the different phases of the MtMS-sD2D session. The MtMS-SC initializes the *MtMS session* by sending to the MtMS-GW the multicast content, the list of devices to be served, and the D2D pairs formed on the basis of the CQI values, (in case they are sent by devices during the previous UL procedures). After that, the *joining procedure* begins. The MtMS-CE handles the paging of all multicast devices, also those which will be served via D2D communications. This step is necessary to trigger devices which must receive data. Paging is a very delicate procedure, especially because of scalability problems and overhead. A good solution to these issues is the enhanced group-paging procedure, which consists of simultaneously paging subgroups of devices belonging to the same multicast group [5]. In this case, the size of each subgroup depends on the amount of available resources, and also affects the number of created subgroups. Paging is accomplished when devices perform the RA procedure. In the case of absent UL segment (e.g., software update applications), during this RA procedure devices have to communicate their CQI values. The formed D2D pairs are communicated by the MtMS-CE to the HeNB, which will be in charge of their coordination. Before concluding the joining procedure, HeNB selects a public and a private key for each D2D device, randomly choosing $x_i \in Z_q^*$ (i.e., a set of integers with a prime number q of elements) as the private key and computing $X_i = g^{x_i}$ as the public key, where g is the fixed primitive element of Z_q^* used as generator/base. The couple (x_i, X_i) is sent from the HeNB to each D2D device via a secure control channel. After receiving the keys, the D2D receiver, from here indicated as D_i, sends a service request message to the HeNB. It is composed by

$$ID_i||z||h[(x_i^+ \oplus \text{opad})||h[(x_i^+ \oplus \text{ipad})||ID_i||z]] \tag{7}$$

where:

- ID_i uniquely identifies D_i in the network;
- z is the first public key for generating the secret key k_c that will be used for data encryption and decryption. It is computed as $z = g^c$, where $c \in Z_q^*$ is randomly chosen by D_i;
- $h[(x_i^+ \oplus \text{opad})||h[(x_i^+ \oplus \text{ipad})||ID_i||z]]$ is the $HMAC_{x_i}(ID_i||z)$. Generally, the $HMAC_k(m)$ is used to guarantee the integrity and authentication of the message m. It is based on the use of any cryptographic hash function h applied to a combination of the original message m and the secret key k. In (7), x_i^+ is the key padded out to size, *opad* and *ipad* are specified padding constants. For the simplification of expression, in the remainder of the paper $h[(k^+ \oplus \text{opad})||h[(k^+ \oplus \text{ipad})||m]]$ will be expressed as $h(\bullet, k)$, where $\bullet$ denotes the message attached by the HMAC and k is the secret key hashed together with the message. Note that x_i is only known by the sender D_i and the

receiver HeNB. In all future steps, the verification of the HMAC will always be performed by the recipients of the messages to verify message integrity and authentication, hence from here on this procedure will be omitted.

After receiving the service request message, the HeNB authenticates the requesting device in the normal cellular communication mode, checking if its ID is registered. In the positive case, the HeNB has to inform both D2D devices of their imminent communication. So, it randomly selects $a \in Z_q^*$ and computes $u = g^a$ as the first public key for generating the secret key k_s to use in the exchange of private messages with the selected D2D transmitter (i.e., the relay node). To communicate to D_j that has been chosen as relay of the D2D communication, the HeNB sends to it the following message:

$$ID_j||ID_i||z||u||h(\bullet, x_j). \tag{8}$$

Simultaneously, to acknowledge the reception of the service request message, the HeNB sends to D_i a response message with ID and public key of the selected transmitter:

$$ID_i||ID_j||X_j||h(\bullet, x_i). \tag{9}$$

This concludes the joining procedure. The following step is the *content delivery procedure*. First of all, MtMS-GW must sign data to send to devices with σ_1:

$$\sigma_1 = H_1(M)^{x_0} \tag{10}$$

where H_1 is a hash function, x_0 is the private key of the MtMS-GW, M is the data to be transmitted. After that, it sends data to devices belonging to the first paged subgroup, using a CMS. In particular, MtMS-GW first excludes devices with the lowest CQI values from the multicast transmission and, subsequently, chooses the MCS that all the remaining devices can support. When D_j, which belongs to the served subgroup, receives data from the HeNB, it already knows it has to forward them to the previously notified D2D receiver; so it carries out all the operations aimed at a secure D2D data transmission. First of all, to allow the receiver to generate the secret key k_c, it randomly selects $b \in Z_q^*$ and computes $y = g^b$ as the second public key for k_c. It does not send y directly to D_i, but it sends it to the HeNB, randomly choosing $f \in Z_q^*$, generating the secret key $k_s = u^f = g^{af}$ and using it to encrypt the public key y. Then, D_j must encrypt data, so it generates the communication key $k_c = z^b = g^{cb}$ and uses it to encrypt the data M. After computing $M' = Enc_{k_c}(M)$, D_j signs the message calculating:

$$\sigma_2 = H_1(ID_j||M'||T_s||\sigma_1)^{x_j} \tag{11}$$

where T_s is the timestamp used against the replay attack. Thus, the D2D communication takes place when D_j sends the following message to D_i:

$$ID_i||ID_j||M'||T_s||\sigma_1||\sigma_2. \tag{12}$$

In order to allow the HeNB to generate the secret key k_s used to encrypt the public key y, D_j computes $v = g^f$ as the second public key for k_s, using $f \in Z_q^*$ previously chosen. Finally, it sends to the HeNB a report:

$$ID_i||ID_j||Enc_{k_s}(y)||v||T_s||h(\bullet, x_j). \tag{13}$$

After receiving data, D_i first has to verify the identity of the transmitter. To this aim, it compares the ID_j reported on the message received by D_j with that communicated by the HeNB and, if the two do not match, the packet is dropped, otherwise it proceeds with the next steps. So, it checks the signature of the transmitter σ_2 and, if it is valid, data are considered sent by the entity corresponding to ID_j. Once the identity of the sender is verified, D_i needs to generate the decryption key k_c to obtain the plaintext. To do this, it sends a public key request message to the HeNB:

$$ID_i||ID_j||T_s||h(\bullet, x_i). \tag{14}$$

After receiving this message, the HeNB generates the decryption key k_s in order to decrypt $Enc_{k_s}(y)$, thus it sends the response message to D_i:

$$ID_i||ID_j||y||T_s||T_i||h(\bullet, x_i) \tag{15}$$

where T_i is employed to record the feedback time.

Thanks to the reception of the public key y, D_i can get the communication key by computing $k_c = y^c = g^{bc}$. So, it can decrypt the message M' to obtain the original data M. To verify the origin of data, it also checks the signature σ_1 and, if it is valid, the data are accepted. Otherwise, it is possible that data may have corrupted. In this case, D_i must send to the HeNB a *beacon* as the evidence of the fake message and to track the malicious attacker:

$$\beta = ID_i||ID_j||M'||T_s||\sigma_1||\sigma_2||h(\bullet, x_i). \tag{16}$$

The beacon must be sent within the timestamp T_i', which satisfies that $T_i' < T_i + \Delta T$.

Finally, HeNB must keep track of any malicious behavior by devices. If any beacon arrives during the time interval ΔT, HeNB first checks the validity of σ_1 and if it is invalid, it is judged that the message did not come from the MtMS-GW and may be fabricated by the transmitter. So, HeNB also verifies the validity of σ_2 to ensure that the fake message comes from the entity corresponding to ID_j. A malicious behavior amount (MBA) counter is stored by HeNB for each device which does not transmit data correctly in the D2D communication. Thus, in case of a malicious behavior by D_j, HeNB increments by one its MBA counter. The MBA counter of a node is incremented if, and only if, the HeNB verifies that the node has not correctly transmitted the data, tampering them before sending to the D2D receiver. Thanks to this, it is not possible to attribute a malicious

behavior to a device if this has not actually occurred. When the MBA related to one device exceeds a given threshold, then the device is punished by the network; for example, it is considered non-eligible as D2D transmitter in future communications. The value of the maliciousness threshold has to be defined on the basis of the number of D2D transmissions performed, then based on the expected MBA values for users. When it is set to zero, only users with a MBA value of zero can be selected as relays, which means that selecting a malicious user is only possible if the user has never had a malicious behavior so far. As the value of the threshold increases, the algorithm is always less selective, this means that more malicious users are chosen as transmitters, even those which in the past have already behaved maliciously.

A summary of the keys used in the system is reported in Table 1.

Table 1. Keys used in the system.

Secret keys	Public keys	Description
k_c	$z = g^c$	First public key for k_c, sent by D_i to HeNB
	$y = g^b$	Second public key for k_c, sent by D_j to HeNB
k_s	$u = g^a$	First public key for k_s, sent by HeNB to D_j
	$v = g^f$	Second public key for k_s, sent by D_j to HeNB

5 Performance Evaluation

We carried out a simulation campaign by using MATLAB tool to analyze the performance of the MtMS-sD2D architecture.

The considered scenario for the results reported in this paper consists of 1000 devices distributed in the edge of a circular LTE-A cell of 1000 m of radius. A bandwidth of 20 MHz, which corresponds to 100 RBs, is available. A TDD LTE frame type 2 configuration 3 is used. Each slot (or Transmission Time Interval, TTI) in the frame lasts 1 ms, so the entire frame has a duration of 10 ms. The Inband D2D mode is chosen, so uplink slots are reserved to D2D communications. In the downlink slots, a multicast transmission allows to send data to in-coverage devices.

In this performance evaluation, the malicious device is the D2D transmitter that does not correctly terminate data transfer to the receiver. In particular, in any case, data sent by a malicious transmitter are lost and cause waste of resources.

The performance of the proposed MtMS-sD2D protocol is evaluated on the basis of the following metrics:

- *Data lost in D2D communications* because of an unreliable transmitter;
- *Percentage of malicious relays* that have been selected by the network.

A comparison between the secure version of the proposed protocol and the non-secure version is shown.

Fig. 3 shows the results in terms of data loss under increasing values of percentage of malicious devices. As can be inferred by the figure, MtMS-sD2D guarantees the lower data loss thanks to the implemented security mechanism. In particular, the achieved improvement is about 30%.

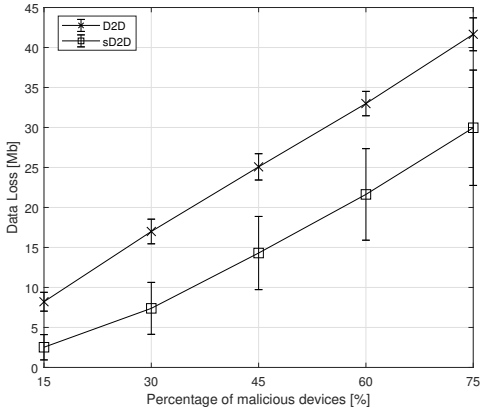

Fig. 3. Data loss under increasing percentage of malicious devices.

Fig. 4 shows the percentage of malicious relays when the percentage of malicious nodes grows. The malicious relays are the D2D transmitters that do not correctly transmit data. As previously discussed, the fact that MtMS-sD2D performs a better selection of D2D nodes is also demonstrated by the fact that a smaller number of nodes that effectively perform a malicious action are chosen to be relay of a D2D communication.

6 Conclusions

In this paper, we have proposed MtMS-sD2D (Machine-type Multicast Service with secure D2D), an architecture for the delivery of multicast traffic in an IoT scenario that takes advantage from D2D communications made reliable by means of a security mechanism based on the Diffie-Hellman key exchange (DHKE) protocol. Furthermore, we have discussed the designed procedures to be introduced in order to efficiently transmit multicast data toward a set of MTC devices.

Preliminary results showed that the proposed architecture is able to improve system performance by avoiding useless allocation of network resources to malicious nodes. Future work will focus on a further analysis of the performance of the algorithm, also evaluating its impact on the energy consumption required for its implementation on resource-constrained devices.

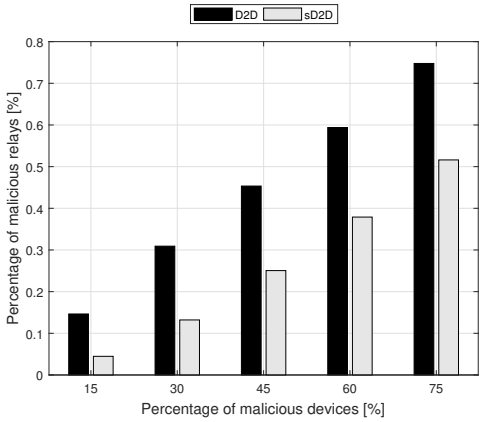

Fig. 4. Percentage of selected malicious relays under increasing percentage of malicious devices.

References

1. F. Boccardi, R.W. Heath, A. Lozano, T.L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," IEEE Communication Magazine, vol. 52, pp. 7480, 2014.
2. Ian F.Akyildiz, Shuai Nie, Shih-Chun Lin, and Manoj Chandrasekaran, "5G roadmap: 10 key enabling technologies," Elsevier Computer Networks, vol. 106, pp. 17 - 48, September 2016.
3. A. Asadi, Q. Wang, and V. Mancuso, "A Survey on Device-to-Device Communications in Cellular Networks," IEEE Communications Surveys & Tutorials, vol. 16, pp. 1801-1819, April 2014.
4. G. Araniti, M. Condoluci, P. Scopelliti, A. Molinaro, and A. Iera, Multicasting over Emerging 5G Networks: Challenges and Perspectives," IEEE Network, vol. 31, pp. 80 - 89, March/April 2017.
5. M. Condoluci, G. Araniti, T. Mahmoodi, and M. Dohler, "Enabling the IoT Machine Age With 5G: Machine-Type Multicast Services for Innovative Real-Time Applications," IEEE Access, vol. 4, pp. 5555 - 5569, May 2016.
6. S. Sicari, A. Rizzardi, L. A. Grieco, A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," Elsevier Computer Networks, vol. 76, pp. 146-164, January 2015.
7. A. Zhang, J. Chen, R. Qingyang Hu, and Yi Qian, "SeDS: Secure Data Sharing Strategy for D2D Communication in LTE-Advanced Networks," IEEE Transactions on Vehicular Technology, vol. 65, pp. 2659 - 2672, April 2016.
8. W. Diffie, and M. E. Hellman, "New Directions in Cryptography," IEEE Transactions on Information Theory, vol. 22, pp. 644-654, November 1976.