IMPACT: International Journal of Research in Humanities, Arts and Literature (IMPACT: IJRHAL)
ISSN (P): 2347–4564; ISSN (E): 2321–8878
Vol. 7, Issue 1, Jan 2019, 545–556

PERFORMANCE ANALYSIS OF KEY GROUP MANAGEMENT STRUCTURE IN WSN FROM NODE TO NODE

A. Singaravelan

Professor, Department of Computer Science, ESSM College of Arts and Science, Tiruvallur, Tamil Nadu, India

Received: 18 Jan 2019 **Accepted:** 25 Jan 2019 **Published:** 31 Jan 2019

ABSTRACT

Group communication is a special type of communication over wired and wireless networks wherein data are exchanged in the form of broadcast packets. During the last decades, many new technologies and concepts, especially based on the theory of group communication, have been implemented

KEYWORDS: *Scalable Group Key Management; Secure Group Communication; Node-To-Node Communication; Group Message Encryption Algorithm*

INTRODUCTION

The security problem of our concern is the establishment and maintenance of a secure Node to Node(N2N)communication via broadcast messages in a dynamic and distributed key management fashion. In this paper, the N2N broadcast communication is referred to as N2N group communication. Recent studies have proposed several approaches for enhancing the N2N group communications security with improved availability, authentication, integrity, and non-repudiation. To provide a secure group communication in N2N, it is necessary to create, manage, and distribute the group keys securely with a low communication overhead. Group key management algorithms can be categorized into two types: centralized and distributed group key management. While the centralized group key management is suitable for symmetric cryptography algorithms, it suffers from high overhead in computation, communication, and storage. The distributed group key management allows every node in the group to participate in the interactive computation of the group key. Hence, it distributes the key management load to all the group members, thereby providing a higher security level and fault-tolerance in integrity and confidentiality. In view of these benefits, we chose the distributed approach. Distributed key management methods are built without a central entity or authority. In these methods, each member of the group is equally trusted and required to participate in managing the keying material. Distributed key management in group communication includes the following operations: generation of cryptographic keys, exchange of keys, re-keying, and update of the keys. The distributed key management methods are commonly used in ad hoc and dynamic networks.

- The general architecture adopts a *hybrid* design approach consisting of two levels (domain and area) of key managers. Each level is independently governed by a key manager. As the number of multicast groups' increases, as well as the number of group members, additional LKMs can be added to support larger group operations.

- Re-keying due to group membership change is contained. In general, scalability problems are reduced by designing the architecture in such a way that any changes in group membership in a particular area do not go beyond that area, and other areas are not affected by the change.

Security Analysis

In this section we analyze security of the proposed framework.

- Availability of secure encryption algorithms.
- Use of secure key establishment techniques to establish long-term keys.
- Use of secure entity and data origin authentication mechanisms to extend simplified protocols,
- Use of some form of *time variant parameter* such as a time stamp in the *text* field within protocol messages for checking that a message received is not a replay of previous ones,
- The key managers (MKM and LKMs) in a domain are fixed and have been securely established prior to commencement of any multicast group communication, and every LKM has established a long-term *Domain-Area* key and a common domain key D−Key with the MKM
- All keys managers (MKM and LKMs in a domain) are trusted entities which all group members trust.
- Availability of secure storage of cryptographic keys for all group communication entities.

Security Assessment

We analyze the security of the proposed KGMF in general to see that the proposed framework meets the requirements.

- *Provision of entity authentication: B*oth group members and key manager entity(s) to authenticate and verify each other's identities.
- *Provision of backward and/or forward secrecy:* A particular security service that is specific to multicast group communication is the provision of confidentiality with respect to backward and forward secrecy.
- *Data (Message) integrity and authentication:* The use of data origin authentication mechanisms in our proposed protocols, provides a means for both group members and key manager(s) to verify the integrity of data received.
- *Secure data exchange:* It can be achieved through careful application of security techniques and mechanisms, by the availability of secure encryption algorithms and that access is only allowed to authorize group members.
- *Secure key distribution:* The distribution of long-term keys to key managers and group members is done in a secure manner, by means for key managers to protect the distribution of short-term keys to all group members.
- *Secure key updates (re-keying):* Supports re-keying of short-term keys which may need to occur whenever there is a change in group membership.
- *Additional key management during host mobility:* Supports the establishment of short-term *session mobility key* that needs to occur prior to host mobility using *Protocol IV(c): Establishment of session mobility key.Member*

moving with backward secrecy Table-1 summarizes re-keying operations of both a traffic key T−Key and an area key A−Key, which occurred due to group membership change. We indicate re-keying of each key with √ notation, otherwise they are indicated with a *dash.*

Table 1: Re-Keying of Traffic Key and Area Key

Re-keying Opertions	Group Key Management Protocols					
	New Joins		Member Leaves		Member Moves	
	Without BS[a]	with BS	Without FS[b]	with FS	Without BS	with BS
Re-key traffic key T-Key	-	√	-	√	-	-
Re-key traffic key T-Key	-	√	-	√	-	√

[a] Backward Secrecy [b] Forward Secrecay

Re-keying of T−Key does not need to occur during host mobility because moving members are still in a same group session. Protocol I: Creation of New Group and Initial Distribution of Keys In this protocol, a new multicast group is created and initial distribution of a traffic key T−Key and an area key A−Key is conducted for all LKMs in a domain, and to the first member of a multicast group.

We analyze the protocol as follows:

- Any host M who wishes to create a multicast group must first establish a secret *Area-Member* key with an LKM, and we have assumed that this was done securely
- We have implicitly assumed that data origin authentication is provided by using a MAC. Thus, we can conclude that if an adversary wants to masquerade or initiate a bogus multicast group, the adversary will not be able to do so unless he has access to the MAC key. In the event that the MAC value received is not the same as the value a member computes, the message will be discarded.
- The host M uses the *Area-Member* key for protecting the communications between itself and the LKM. If an adversary gets hold of the encrypted messages between M and LKM, the adversary has no way of decrypting the messages because he does not have access to the *Area-Member* key.
- After granting the permission to create a multicast group, MKM generates and distributes a traffic key T−Key to all LKMs. The distribution of this key to all LKMs is protected either by the *Domain-Area* key (if the key is to be sent separately via unicast to every LKM), or a common domain key D−Key (if the key is to be sent one time via multicast). If an adversary wants to get hold of the T−Key, the adversary has no access to either of these keys (*Domain-Area* or D−Key), so he cannot obtain the T−Key. (e) Similarly, an LKM (when the host joins a multicast group) generates and distributes an area key A−Key to the host M along with the T−Key it receives from the MKM. The distribution of these keys is protected under the *Area-Member* key which is shared only between the LKM and host M. The adversary has no access to the *Area-Member* key, so he cannot obtain the T−Key or the A−Key.
- Other information distributed during this protocol is also protected under secret keys known only to key managers (MKM and LKMs) and host M. Thus, a passive observer knows nothing about the properties of the new multicast group.

Protocol II(a): New Member Joining without Backward Secrecy

In this protocol, a new join of a host to become a member of a multicast group is conducted with no provision of backward secrecy. This means that when a new member joins a multicast group, the same keys (T−Key and A−Key) that are currently in use are given to the newly joined member.

We analyze the protocol as follows:

- Any host M who wishes to join a multicast group must first establish a secret *Area-Member* key with an LKM, and we have assumed that this was done securely.
- After receiving the join−request from M, LKM relays the request to the MKM protected under the *Domain-Area* key. If an adversary gets hold of the encrypted messages between LKM and MKM, the adversary has no way of decrypting the messages because he does not have access to the *Domain-Area* key shared only between LKM and MKM.
- After receiving the join−granted message from MKM, the LKM sends the current keys to M in the form of Join−Token. This message is protected under an *Area-Member* key shared only between LKM and M. If an adversary wants to get hold of the token, the adversary has no access to the *Area-Member* key, so he cannot obtain T−Key, A−Key or other group related information. If an adversary intercepts or modifies the message content, this can easily be detected by MKM, LKM or M when the implicit MAC value is checked against the value received.

Protocol II(b): New Member Joining with Backward Secrecy

In this protocol, a new join of a host to become a member of a multicast group is conducted with provision of backward secrecy. When a new member joins a multicast group, re-keying of cryptographic keys occurs. This results in the new member and other members in the area (where the new join occurs) obtaining new keys T−Keynew and A−Keynew. This also results in other group members across the domain obtaining a new T−Keynew. As for *Protocol II(a)*, we have assumed that *Protocol I* was successfully conducted.

We analyze the protocol as follows:

- Any member M wishing to leave a multicast group sends a leave−notify message to LKM protected under an *Area-Member* key, who then passes the message to the MKM protected under a *Domain-Area* key, and we have assumed that these keys were established securely between the member M and LKM, and between LKM and MKM
- After receiving the leave−notify message from LKM, MKM updates its *HisList*, and the reason for leaving is logged. We have assumed that this list is maintained and kept securely by the MKM.

Protocol III(b): Existing Member Leaving with Forward Secrecy

In this protocol, an existing member leaving a multicast group is conducted with provision of forward secrecy. When a member leaves, the remaining members of the multicast group need to be re-keyed. This results in all remaining group members in an area where the leave occurs obtaining a new area key A−Keynew, and all LKMs and group members in the

domain obtaining a new traffic key T−Keynew.. As in *Protocol III(a)* the information about the leaving member is logged in *HisList*.

Due to the similarity with *Protocol III(a)*, I just analyze the differences, as follows:

- After receiving the leave−notify message from M (or an eject−notify message from MKM), LKM initiates the re-keying of its area key A−Key. This results in all remaining group members in the area (excluding the leaving member) obtaining a new area key A−Keynew. This new key is sent via *unicast*, protected under the *Area-Member* keys. If an adversary gets hold of the encrypted message, he will not be able to decrypt it as he has no access to the secret shared only between each member and LKM.
- After receiving the leave−notify message from LKM (or after sending an eject−notify to LKM), MKM initiates the re-keying of the group's traffic key T−Key. This results in all LKMs and group members (via LKM) in the domain obtaining a new traffic key T−Keynew. As in *Protocol II(b)* MKM can send this new key to all LKMs either via *multicast* protected under D−Key, or via *unicast* protected under *Domain- Area* keys.
- We have implicitly assumed the provision of data origin authentication using MACs. We can conclude that if an adversary wants to masquerade as someone else in order to leave a multicast group when the MAC value computed differs from the value obtained from the received message.

Protocol IV(a): Existing Member Moving without Backward Secrecy

In this protocol, the transfer of a group member from one area to another is conducted with no consideration for backward secrecy. This means that when a member moves from one area to another, the member is given the area key A−Keyv of the visited area. All key managers (MKM and all LKMs) in the domain need to update their *MobList* whenever a *move* occurs. We have assumed that these lists are maintained and kept securely by the key managers.

We analyze the protocol as follows:

- A member Mi who wishes to move into another area must first establish a short-term *session mobility* key with the LKM of the visited area, and we have assumed that this was done securely.
- A member Mi uses this short-term key to secure communications with the LKM of the visited area. If an ad versary wants to masquerade as some moving member in order to get hold of the area key A−Keyv of the visited area, he will not be able to do so because he has no access to the *session mobility* key shared only between the moving member and the LKM of the visited area.
- We have implicitly assumed the provision of data origin authentication using MACs. Thus, we can conclude that if an adversary wants to masquerade as some moving member in order to get hold of A−Keyv, the adversary will not able to do so because he has no access to the MAC key. Other entities (MKM, LKM and Mi) can easily check the integrity of messages received via the same process.
- After obtaining the *session mobility* key, Mi initiates the *move* protocol by sending a move−notify message to its local Local key manager LKMi, protected under the *Area-Member* key, and to the visited Local key manager LKMv, protected under the *session mobility* key

- After receiving the move−notify message from MKM, LKMv acknowledges the move by Mi and sends its area key A−Keyv (A−Keyvnew is sent if there has been re-keying of its area key) to Mi protected under the *session mobility* key.
- All affected key managers (MKM, LKMi and LKMv) update their *MobList*, and area(s) visited are logged. We assume that these lists are maintained and kept securely by the key managers.

Protocol IV(b): Member Moving with Backward Secrecy

In this protocol, the transfer of a group member from one area to another is conducted with provision for backward secrecy. When a member moves from one area to another, the area where the member is moving to needs to be re-keyed with a new area key. This results in all group members in the visited area, including the moving member, obtaining a new area key A−Keyvnew. As in *Protocol IV(a),* the information on the member moved is logged in each affected key manager's *MobList.*

Protocol IV(c): Establishment of Session Mobility Key

In this protocol, a *session mobility* key for *host mobility* is established between the moving member Mi and the LKM of the visited area LKMv. This results in Mi and LKMv obtaining the session mobility key Sm−Keyiv.

As part of *Protocol IV(a)* and *Protocol IV(b),* we have assumed that there is an established multicast group.

We analyze the protocol as follows:

- A member Mi who wishes to establish a session mobility key with LKMv, must first send a move−wish message to its local Local key manager LKMi. This message is protected under the Area-Member key shared only between Mi and LKMi, and we have assumed that this was done securely.
- After receiving the move−wish message from LKMi, MKM generates a session mobility key, and we have assumed that this was done securely

Performance Analysis

As mentioned earlier, the analysis on performance and scalability of the proposed framework is presented in terms of *operational complexity*, *re-keying complexity*, *storage complexity*, and *communication complexity*. We use the following notation to analyze the performance of the protocols:

Generic notation such as MKM, LKM andM (orMi) to denote Master key managers, Local key managers and group members of a multicast group as in earlier protocol designs are also used here In addition:

- Let |Ax| be the number of group members in an area x.
- Let |AD| be the number of areas in a domain D (and hence LKMs).
- Let |TKD| be the number of traffic keys in a domain.
- Let |TKA| be the number of traffic keys in an area.

- Let |hMob| be the number of hMob in an area, where an hMob is a set of security parameters, consisting a *session mobility key* and an *area key* of a visited area, needed by a group member for *host mobility*.

The performance assessment of our basic protocol designs is categorized based on the costs incurred, as follows:

Operational complexity This assessment demonstrates the framework performance with respect to the number of encryptions (or decryptions) that need to be performed during secure group operations. We note that *one encryption (or decryption) is equivalent to one cost*, and denote this by E.

Table 2: Cost for Each Group Operation Reasonably Spread amongst the Key Managers

Group Operations		Operational Complexity						
		DKM	AKMs			Group Members		
			AKM_i	AKM_x	$\|A_D\|$-1	M (or M_i)	M_{Ax}	$\|M_{Ai}\|$-1
Creation & Initial Keys Distribution		2E	4E		$(\|A_D\|-1)E$	2E		
New joins	Without BS[a]	2E	4E			2E		
	With BS	2E	5E		$(\|A_D\|-1)E$	2E		$(\|M_{Ai}\|-1)E$
Member leaves	Without FS^b_v	E	2E			E		
	Without FS_{inv}	E	2E			E		
	With FS_v	2E	4E		$(\|A_D\|-1)E$	E		$(\|M_{Ai}\|-1)E$
	With FS_{inv}	E	3E		$(\|A_D\|-1)E$	E		$(\|M_{Ai}\|-1)E$
Member moves	Without BS	$4E + T_{Mob\_agree}$	$3E + T_{Mob\_agree}$	$4E + T_{Mob\_agree}$		$3E + T_{Mob\_agree}$		
	With BS	$4E + T_{Mob\_agree}$	$3E + T_{Mob\_agree}$	$5E + T_{Mob\_agree}$		$3E + T_{Mob\_agree}$	E	

[a] Backward Secrecy
[b] Forward Secrecy
$_v$ Voluntary leave
$_{inv}$ Involuntary leave

The use of symmetric encryption is also an advantage as it is computationally faster, hence saves battery power. *Table-3* illustrates an example of performance overhead that may incur in a larger scale of network size. (BBC, 2007) and (TechNews, 2007) report that at least 7 millions (MIL) people are anticipated to be using iPhone (the latest smart phone technology) (Apple, 2007) in UK by the end of 2008. Based on this information, we illustrate the example by using the same network size. With some degree of host mobility that may occur during the participation in a multicast group across multiple areas, *Table-3* An example of cost estimation on performance overhead.

The estimation cost provided is based on the E cost obtained from *Table-2*.

- The first column represents the number of user participation(%) in multicast group.
- The other columns represent cost estimation in terms of performance overhead due to provision of security and host mobility. Note that the average cost listed in each performance overhead is obtained from *Table-2*.

Table 3: An Example of Cost Estimation on Performance Overhead

Network size: 7 millions (MIL) users, potentially with 50% host mobility.

User participation in group communication (%)	Estimation of Total Operational Overhead at NSP		
	No provision for security* (with average cost of 6E per user)	With provision for security+ (with average cost of 12E per join/leave operation)	Due to host mobility (with average cost of 15E per user)
10 (700,000 users)	≈ 4.2 MIL	≈ 8.4 MIL	≈ 5.25 MIL
30 (2.1 MIL users)	≈ 12.6 MIL	≈ 25.2 MIL	≈ 15.75 MIL
50 (3.5 MIL users)	≈ 21 MIL	≈ 42 MIL	≈ 26.25 MIL
70 (4.9 MIL users)	≈ 29.4 MIL	≈ 58.8 MIL	≈ 36.75 MIL
100 (7 MIL users)	≈ 42 MIL	≈ 84 MIL	≈ 52.5 MIL

*Except for initial set up of a multicast group

+for provision of backward and/or forward secrecy, which requires re-keying to occur whenever new member joins and existing member leaves.

NSP – Network Service Provider such Vodafone, Orange, T-Mobile or O2.

The average cost of 6E per user (in 2nd column) is the average of operational cost incurred by MKM and LKMi (see MKM and LKMi columns in the 1st row *Table-2*). Similarly, the average cost of 15E for host mobility (last column) is the average of operational cost incurred during member moves protocol in *Table-2* (see last row in MKM, LKMi and LKMv columns). It shows that as the network size increases along with host mobility, as well as group membership, the amount of performance overhead (that the network has to manage) also increases.

Re-keying Complexity

This assessment demonstrates the cost in terms of the number of key updates (or re-keying) that has to occur. Assessment is based on *one re-keying is equivalent to one cost*, and we summarize this in *Table-4*. Cost assessment based on re-keying complexity.*Table-4* shows that while there is no re-keying cost at creation of a multicast group, two key updates (re-keying of a traffic key T−Key and re-keying of an area key A−Key) are required every time a new join, or a leave occurs. There is no need for key update if provision of backward and forward secrecy is not required, thus no cost in terms of re-keying incurs. On the other hand, a *move* requires only one re-keying cost, and that is for re-keying the area key of the visited area.

Table 4: Shows that while there is No Re-Keying Cost at Creation of a Multicast Group

Key Update		
Group Processes	**Keys**	**Total**
Re-keying at *creation*	-	-
Re-keying at *join*	T_Key, A_Key	2
Re-keying at *leave*	T_Key, A_Key	2
Re-keying at *move*	A_Key	1

Depending on requirements of multicast applications, the cost of re-keying can be reduced to half of the key updates normally required, if only one provision for either backward or forward secrecy is required. For example, if provision of security is not required during host mobility (in other words, no backward secrecy) and group members are free to move between areas, no re-keying needs to occur, thus no cost incurs.

Storage Complexity

This assessment in *Table-5* demonstrates the cost in terms of the amount of key storage required by communicating entities. Assessment is based on *one key stored is equivalent to one cost*. We conclude that the main cost of key storage is

reasonably distributed amongst key managers (MKM and LKMs), while keeping the cost of key storage at group members M minimal.

Table 5: Cost Assessment Based on Storage Complexity

Key Storage		
Entities	**Keys**	**Total**
No. of Keys at Master key manager *MKM*	D_Key + $\|TK_D\|$ + $\|A_D\|$	1 + $\|TK_D\|$ + $\|A_D\|$
No. of Keys at Local key manager LKM_i	D_Key + A_Key + DAi_Key + $\|TK_A\|$ + $\|A_X\|$	3 + $\|TK_A\|$ + $\|A_X\|$
No. of Keys at group member M_i	AiMi_D_key + A_Key + T_Key+$\|h_{Mob}\|$	3+$\|h_{Mob}\|$

A group member with a typical mobile device 206MHz processor with 64MB of RAM (as mentioned earlier) can comfortably cope with the total cost of 3+|hMob| keys storage (see *Table-5*), with each key length of 128 bits (as discussed in. Also, as the main key manager in a domain, MKM usually carries a lot of weight as the primary entity for managing group operations.

The load for storing keys is shared with other key managers (LKMs) in a domain, hence the operational load is reasonably balanced amongst MKM and LKMs. For example, MKM does not need to keep *Area-Member* key pairs shared between an LKM and a group member, which are managed at the area level by the LKM. We observe that the number of keys kept by MKM increases as the number of multicast groups increases.

Communication Complexity

This assessment demonstrates the framework performance with respect Table-6: Cost assessment based on communication complexity. to the number of messages sent by each communicating entity (key managers and group member) involved in group operations.

For every group operation, one cost is incurred when:

- A *unicast* message is sent, and we denote this by a u.
- A *multicast* message is sent, and we denote this by m. This is summarized in *Table-6.*

Note that we do not specify where the messages were being sent to, but rather analyze the number of messages originating from a particular entity.

From *Table-6*, we observe that the number of messages sent by a group member M throughout group operations is reasonably low, at most 2u. The cost incurred during new join and leave operations (with provision of backward and forward secrecy) varies depending on whether a *unicast* or *multicast* message is sent.

Table 6: Cost Assessment based on Communication Complexity

Group Operations		No. of Message (originates from)			
		M(or Mi)	LKMi	MKM	LKMv
Creation & Initial Keys Distribution		U	2U	$\frac{\|A_D\|U}{m}$	----
New joins	Without BS[a]	U	2U	U	----
	With BS	U	$\frac{(2+(\|A_X\|-1))U}{2U+m}$	$\frac{\|A_D\|U}{m}$	----
Member leaves	Without FS[b]	U	U	-	----
	Without FS_{inv}	-	U	U	
	With FSv	U	$(1+(\|A_X\|-1))U$	$\frac{\|A_D\|U}{m}$	----
	With FS_{inv}	-	$(1+(\|A_X\|-1))U$	$\frac{\|A_D\|U}{m}$	----
Member moves	Without BS	2U	U	2U	2U
	With BS	2U	U	2U	$\frac{(2+\|A_X\|)U}{2U+m}$

[a] Backward secrecy $_v$ Voluntary leave u:unicast
[a] Forward secrecy $_{inv}$ In-Voluntary leave m:Multicast

This cost from LKMi can be reduced significantly if *multicast* is used. In the same example, it is reduced to 2u + m, which is a total cost of just three messages. Similarly, MKM can reduce the cost of messages sent to all LKMs in the domain (see *3rd column: MKM*) by using *multicast*, which costs only *one* message, instead of |AD| messages if *unicast* is used. By using the multicast functionality, a message intended to a group of recipients, such as all group members in an area, can be sent once by the LKM of that area. This is important in Wireless Networks where only limited bandwidth is available.

SUMMARY

In this Paper, I have assessed the proposed framework. I have shown the extent to which the framework meets its specified requirements and design objectives. These, my belief has been addressed and achieved reasonably well, although the actual feasibility of the framework can probably only be verified through practical implementation.

The proposed method satisfies all security requirements for key management and message confidentiality. Formal security algorithm analysis and extensive simulation results show the enhanced performance and effectiveness of the proposed method for N2N group communication. Therefore, the proposed method is applicable for both small-scale and large-scale N2N group communications. In a future work, I will investigate the proposed method under different application scenarios of N2N communications. In this regard, I will further evaluate group key management issues with overlapping clusters 5G technologies.

REFERENCES

1. *Lu, Z.; Qu, G.; Liu, Z. A Survey on Recent Advances in Vehicular Network Security, Trust, and Privacy. IEEE Trans. Intell. Transp. Syst. 2018, 20, 760–776. [CrossRef]*
2. *Li, C.; Ji, S.; Zhang, X.; Wang, H.; Li, D.; Liu, H. An Effective and Secure Key Management Protocol for Message Delivery in Autonomous Vehicular Clouds. Sensors 2018, 18, 2896. [CrossRef]*
3. *Liu, Y.;Wang, Y.; Chang, G. Efficient Privacy-Preserving Dual Authentication and Key Agreement Scheme for Secure V2V Communications in an IoV Paradigm. IEEE Trans. Intell. Transp. Syst. 2017, 18, 2740–2749.*
4. *Niu, Q. ECDH-based Scalable Distributed Key Management Scheme for Secure Group Communication. J. Comput. 2014, 9, 153–160. [CrossRef]*
5. *Yadav, M.; Singh, K.; Pandey, A.S. Key management in efficient and secure group communication. In Proceedings of the 2016 International Conference on Emerging Trends in Electrical Electronics&Sustainable Energy Systems (ICETEESES), Sultanpur, India, 11–13 March 2016; pp. 196–203.*
6. *Mejri, M.N.; Ben-Othman, J.; Hamdi, M. Survey on VANET security challenges and possible cryptographic solutions. Veh. Commun. 2014, 1, 53–66. [CrossRef]*
7. *Engoulou, R.G.; Bellaïche, M.; Pierre, S.; Quintero, A. VANET security surveys. Comput. Commun. 2014, 44, 1–13. [CrossRef]*
8. *G. Sharmaa, S. Balaa, A.K.Vermaa, "Security Frameworks for Wireless Sensor Networks-Review", 2nd International Conference on Communication, Computing & Security, SciVerse Science Direct, 2012.*
9. *C. Cheikhrouhou, A. Koubâa, G. Dini, and M. Abid, "RiSeG: a ring based secure group communication protocol for resource-constrained wireless sensor networks", Personal and Ubiquitous Computing, Vol. 15, No. 8, pp. 783-797, 2011.*
10. *X. Wanga, P. Lia, Y. Suia, and H. Yanga, "A Hexagon-based Key Pre-distribution Scheme for Wireless Sensor Networks", Journal of Information & Computational Science, Vol. 11 (8), pp. 2479-2491, 2014.*
11. *M. Miettinen, N. Asokan, T.D.Nguyen, A-R.Sadeghi, and M. Sobhani, "Context-Based Zero-Interaction Pairing and Key Evolution for Advanced Personal Devices", In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, pp. 880-891, 2014.*