Hybrid Cryptography for Secure Data Communication in Wireless Sensor Networks

Shiva Prakash and Ashish Rajput

Abstract Nodes in the wireless sensor networks (WSN) have limited battery power and deployed to run few days. In general, sensor nodes are placed at very complicated locations; therefore, charging or replacement of nodes battery is very difficult. Hence, it is highly unfavourable to use the complex data security method. The main objective of this paper is to develop an enhanced algorithm to provide secure data communication as well as to enhance the lifetime of the WSNs. We have developed a hybrid algorithm for the data encryption and decryption in WSN. As in symmetric approach the key sharing was a major issue, we have performed the key generation using ECC algorithm which is an asymmetric key algorithm, and using this key, we will encrypt and decrypt data using AES which is a symmetric key algorithm. Hence, we have developed an enhanced algorithm which exploits the advantages of the two algorithms. The proposed algorithm, which reduces complexity as compared with ECC, is used only for key generation and not for data encryption/decryption and it is more secure than AES as it has solved the problem of key sharing in AES.

Keywords Key sharing · Secure data communication · Elliptic curve cryptography (ECC) · Advanced encryption standard (AES) Hybrid cryptography

1 Introduction

WSN is a network consisting of a multiple number of autonomous sensor nodes. These nodes are connected to one or more base stations. Due to the continuously growing applicability and scope of WSNs, it has become increasingly susceptible

S. Prakash (✉) · A. Rajput
Department of Computer Science and Engineering,
M.M.M. University of Technology, Gorakhpur, India
e-mail: shiva_pkec@yahoo.com

A. Rajput
e-mail: arashish31@gmail.com

G. M. Perez et al. (eds.), *Ambient Communications and Computer Systems*,
Advances in Intelligent Systems and Computing 696,
https://doi.org/10.1007/978-981-10-7386-1_50

and exposed to different attacks. Hence, there is a great need of providing an efficient security framework to provide secure data communication and aggregation in WSN [1]. Author [2] focus stealthy attacks in sensor network as if aggregation result arrived and user accepts it, then there is a very high chance that the described result is "nearby" to the factual result value. As the computational capability, energy and storage resources of the sensor nodes are very limited in WSN, the identification of a suitable cryptography mechanism in WSN is not an easy process. The process of modifying, transmitting and storing data in such a way such that an illegitimate or an unauthorized attacker cannot gain access to the confidential and personal data stored in the system is termed as cryptography.

The main aim of our paper is to provide an efficient algorithm for Secure Data Communication in WSNs. As the sensor nodes have limited energy, computational and memory resources, our algorithm should be able to fully utilize them and to provide proper security and data confidentiality.

There are basically two types of cryptographic techniques to provide data security and confidentiality [3, 4]. The two cryptographic techniques are symmetric and asymmetric key techniques. Both the techniques have their own pros and cons. The main advantage of symmetric key algorithm is that it is fast and has less complexity. However, it is less secure than asymmetric key techniques because it uses single secure key for data encryption and decryption [5, 6]. On the other hand, asymmetric algorithms are slow and complex, but they provide high level of security.

In this paper, we will be first describing the research challenges in the field of WSN after that we have performed an exhaustive literature review of the topic "Secure Data Communication in WSN" and given proposed algorithm and finally the future scope and conclusion.

1.1 Research Challenges in WSN

The use of WSNs [7, 8] has significantly increased in the areas such as health care, biologic, environment and structure health monitoring.

The key challenges (problems) in WSNs include:

- Data confidentiality and integrity,
- Power management (in large),
- Topology control, localization and dependability of nodes,
- Query processing,
- Storage and routing of data,
- Real-world protocols,
- Programming abstractions,
- Self-calibration and self-healing,
- Maintenance.

In our work, we have focussed on the security and privacy issue. By solving this issue, we would be able to achieve secure data communication of the nodes in WSN.

2 Literature Review

In this section, we have given a brief description of the well-known work done by different authors in secure data communication in WSN. We have focussed ourselves on the two-major data encryption/decryption algorithms: the first one is symmetric key algorithm, advanced encryption standard (AES); the second one is an asymmetric key algorithm elliptic curve cryptography (ECC).

In [9], the authors proposed Secure Data Aggregation and Verification (SDAV), a secure aggregation with verification. The authors used the elliptic curve cryptography (ECC) due to its low key and its effectiveness in terms of computation and bandwidth over traditional asymmetric algorithms. Indeed, the ECC enables one sensor to effectively calculate a signature while the verification is devoted to the base station. In SDAV, the aggregator collects the encrypted data of its members, decrypts, averages and returns the result to its members. Each member compares the measurement with the average received; if the difference exceeds a certain threshold, it generates a partial signature using the group key generated in the deployment phase and sends it to the aggregator. The aggregator generates and sends the full signature to the base station. If it is rejected, the integrity of measures is verified using the Merkle hash tree. The solution accepts as the average aggregate function [10]. This solution is robust against sneak attacks. It is robust also against Sybille attacks since each node has its own identifier and shares a secret key with the CH. However, this solution is vulnerable to attack by sending selective packets and to replay attack. Indeed, a compromise aggregator can ignore packets.

In [7], the author proposed an efficient aggregation of encrypted data. The proposed solution preserves the integrity and confidentiality of data. The authors proposed an additive homomorphic encryption operation on the data. The basic idea is to replace the XOR operation (traditionally used in stream cipher) by a simple modular addition. This solution is robust against passive attacks. Security against active attacks has not been considered. The proposed system is probabilistic. This feature makes the complicated cryptanalysis since the sensed measure is covered by the injected random value. However, the source of this value constitutes the greatest threat to the analysis of encrypted data.

In [11], author proposed a safety threshold for aggregation. This protocol allows the base station to accept the aggregate data only if many of nodes send similar data. The protocol consists of three phases: a phase for key management, a second for secure aggregation and one for authentication. In the first phase, each node shares a key with the corresponding aggregator, once the latter has a key with each of the members; it is a broadcast of a temporary group key for all its members. In the second phase, each node captures, encrypts and transmits its data to the

aggregator, it calculates the encrypted aggregate data using the group key, and then it makes a broadcast of encrypted data to all its members. In [12], author proposed a solution to monitor the aggregation nodes. These authors combine statistics with artificial intelligence, a combination in which a node can detect whether the aggregator sends incorrect aggregates. For this, the authors developed a mechanism that allows nodes to record a reputation score for their respective parents. Each node only knows its value and the aggregated data of the aggregator [13], and each node uses statistics information to calculate the probability that the aggregator transmits a correct aggregate data. This probability is used to update the reliability of the parent node using a learning algorithm.

Authors [14] have proposed a secure reliable data aggregation and indicated that the only cryptography cannot guarantee sufficient security for WSNs and propose a trusted network. The idea is that each sensor node observes the behaviour of its neighbours to develop a trust level.

In [3, 8], the authors propose a secure solution that consists of detection intrusion system of aggregated data. In this solution, the authors proposed detection at several levels: the first level consists of a set of IDS agents (Intrusion Detection System) which applies detection policies based on rules for normal behaviour modelling of a node. Authors [15] have presented a robust hybrid security algorithm for wireless sensor networks, and its results show better security for smaller encryption and decryption time.

The main limitations are as follows:

- In the solution proposed in [16], there was extremely limited amount of memory and, if even a single node is compromised, the security of the entire WSN is compromised. Here, they are using symmetric key cryptography. The attacker can compromise many more nodes to compromise communication links.
- The algorithm [9] makes use of the decryption operation. It is not suitable for the encryption of a large-sized data. If it is implemented on large-sized data, then the system will become extremely slow. Even the integrity is compromised.
- The authors [17] have proposed a solution for secure data communication in WSNs in which they have used RSA algorithm for the generation of key and AES for data encryption/decryption, and the major limitation in their work is that the key size in RSA is very large due to which it was high resource consuming.
- The high security is provided in [12] due to ECC. However, integrity is compromised and even the private key is not secure.
- It provides a high-level security [14] thanks to the improvement in the key size. Confidentiality and authentication are checked. However, it is not fit for the low-capacity sensor nodes.

- It ensures data confidentiality and authentication [3] and it provides robust end-to-end communication. However, the attacker can disrupt any event report, and if an attacker determines a node's private key, his or her entire messages can be read. The inefficient operations applied in the encryption phase makes the algorithm relatively vulnerable.

3 Problem Statements

After performing an exhaustive literature review of different research papers, we have identified the limitations in the existing work as stated above.

We have observed that if the sensor nodes use asymmetric encryption and decryption of data, then it will result in performance deterioration due to the limited memory, energy and computation resources; if we use symmetric key encryption, then data transfer is less secure as if the key of the algorithm is exposed then the security of the entire system will be compromised.

Thereby, we have designed an algorithm which is secure enough to provide proper data security along with low complexity. Our algorithm should be able to provide data integrity and confidentiality. We are using ECC algorithm for the key generation and AES for data encryption and decryption.

4 Comparative Study

In this section, we have performed comparative study as in Table 1 of well-known work done in the field of Secure Data Communication in WSN by the different authors. We are presenting the solution which they have proposed along with advantages and disadvantage. We will be using both encryption and decryption. But the main serious disadvantage of this is that data confidentiality is compromised. On the other hand, if we use asymmetric key algorithm both data integrity and confidentiality are preserved; however, it is highly complex and inefficient. In the study of symmetric encryption algorithms of various authors in the field of secure data communication in WSN, we found that the encryption and decryption of data are efficient and fast and it also preserves data integrity for small-sized data if we use symmetric key algorithm because it uses single key for both.

Table 1 Secure data communication in WSN

Author (Year)	Proposed solution	Advantages	Disadvantages
B. Przydatek (2007)	Data will be accepted only if the aggregated data is in good approximation of previous data [2]	Prevention of stealthy attacks and forward secure authentication	Extremely inefficient for large volume of data, no confidentiality
H. Hayouni (2009)	Additive homomorphic encryption is applied on the data by the nodes [14]	High efficiency, confidentiality, prevention of replay attack	Complicated cryptanalysis, probabilistic
Vu (2014)	Threshold aggregation and separate phase for key management, aggregation, and authentication	Data confidentiality, integrity	Unfavourable for heterogeneous low-capacity nodes
R. Pahal (2013)	Use of AES symmetric cryptography [4]	Fast, simple, single key for encryption–decryption	Prone to eavesdropping
S. Ozdemir (2015)	Use of elliptic curve cryptography for secure data aggregation [13]	High confidentiality, suitable for large-sized data	Integrity is compromised, slow
Rawya Rizk (2015)	A robust hybrid security algorithm [15]	Low processing overhead and lower energy consumption	High complexity of proposed solution
Z. Dawahdeh (2016)	Use of Menezes-Vanstone elliptic curve cryptography for secure data aggregation [10]	High confidentiality, faster for small-sized data	Integrity is compromised, not suitable for larger data

5 Proposed Approach

In this paper, we are presenting an enhanced algorithm for providing Secure Data Communication in WSNs. We have tried to overcome the limitations discovered in the previous work done by different authors in the field of Secure Data Communication in WSNs. We have proposed an algorithm which makes the optimum use of the limited resources in WSN with the target of providing high security.

AES is one of the most effective symmetric key algorithms that requires single key for encryption and decryption of data. It works with fixed key size that can be of 128/192 or 256 bits.

The complexity of ECC is directly proportional to the size of the data to be transmitted. It is more secure but slower and high resource consuming than AES.

We are using ECC for key generation and sharing and AES for the purpose of data encryption and decryption.

The proposed solution requires three stages for both encryption and decryption of data at the sender and the receiver:

- Initialization of resources, key generation and data encryption/decryption.

Initialization of resources

This will be done in the beginning during the set-up of the nodes before the transmission of any data. Given below are the operations that will be performed in this step:

- Firstly, a common elliptic curve and a general point lying on it will be sent to all the nodes.
- All the nodes must perform this operation over a secure channel.
- This operation will be performed once in the lifetime of a node, and this is the only step that must be performed over a secure channel.

Key Generation

The major problem occurring in the AES algorithm was key sharing as the same key was being used for both the encryption and decryption of data. If the key is released, then the security and privacy of the entire data to be transmitted will be compromised.

To overcome this, we will be using asymmetric key algorithm, i.e. MVECC for effective key sharing. Key generation will take place as follows:

- Both the sender and the receiver will choose a random number n_A € [1, p-1]. This number will be termed as its private key.
- The sender will now compute its public key by multiplying its private key by the common point that was shared in the initialization process $P_A = n_A.G$.
- This public key will be broadcasted to every other node present in the WSN.
- The general point that will be used in the computation of the public key will be the key used in the previous data encryption.

Data Encryption and Decryption

AES is one of the most effective and fast algorithm for data encryption and decryption. It is a symmetric key algorithm that uses single key for encryption and decryption which makes it vulnerable and less secure. However, if the problem of key sharing is solved, then the speed and efficiency of the system can be drastically incremented. We will be performing the data encryption and decryption using 256-bit keyed AES. Given below are the various steps involved:

- The data to be transmitted will be first divided into blocks of 128 bits.
- Now, the 10 rounds of AES will be applied on this data using the key that is generated in the previous step.
- The key will be of fixed length, i.e. 128 bits, and it will be added in each round with the data.
- The encrypted data will be sent across the unsecure channel to the receiver node.
- The receiver will use AES decryption to get back the original data using the key using ECC.
- Note that the key that will be used for the data encryption and decryption will change in each transmission.

- A node will generate its private key randomly, and it will get its public key by using ECC arithmetic by multiplying its private key by the general point in the graph that is the key used in the previous transmission (Fig. 1).

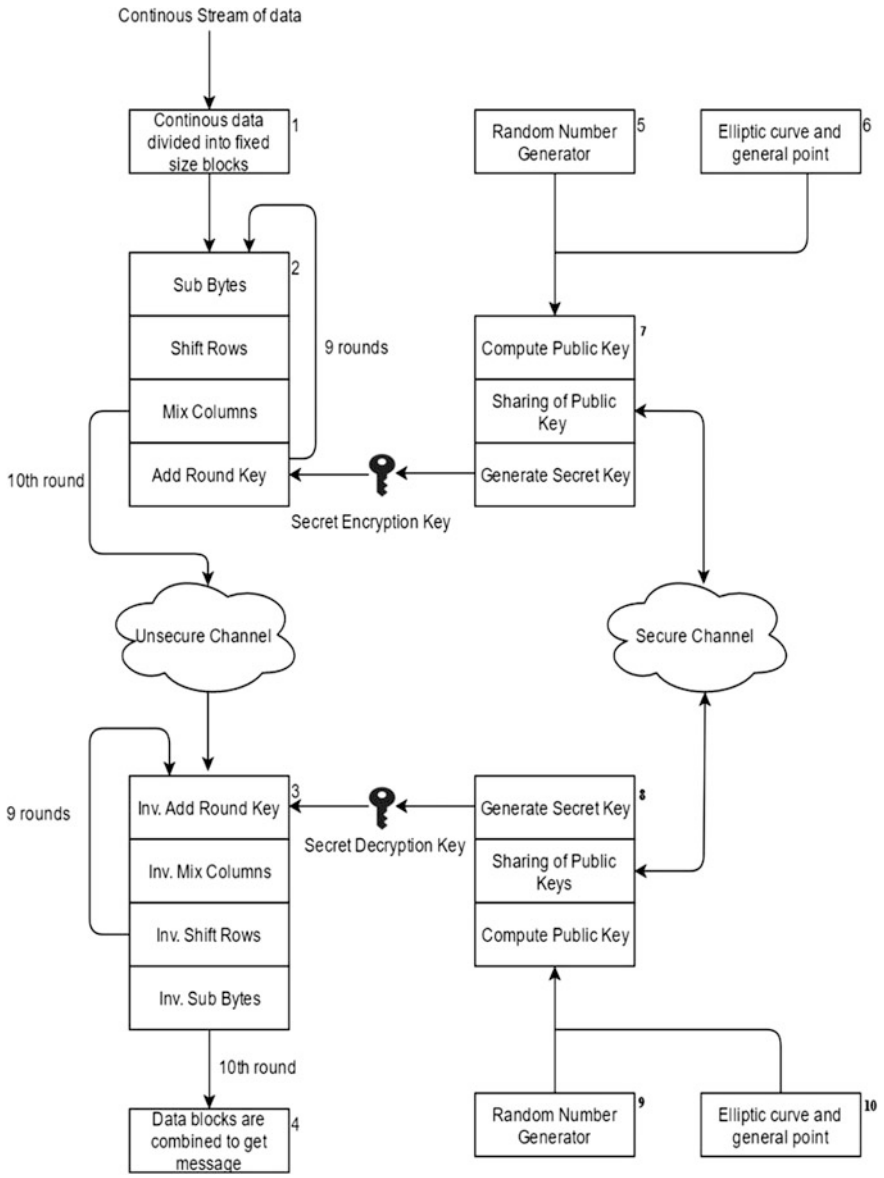

Fig. 1 Block diagram of proposed framework

5.1 Proposed Algorithm

// Initialization of resources N: set of nodes, N_i: i^{th} node, a, b, p are the parameters of the
// elliptic curve : $y^2 = x^3 + ax + b$ (mod p), G is general point on elliptic curve
1. nodes N_i €N
2. Share the elliptic curve parameters and the general point over a secure channel
// Key Generation and sharing using ECC
3. All nodes N_i €N
4. Randomly generate the private key (n_i) such that n_i € [1,p-1]
5. Compute the public key $P_{i =} n_A.G$ // using ECC multiplication
6. Broadcast the public key P_i to other nodes N_j €N
// Computation of Secret key for data encryption/ decryption between two nodes
7. if (Node U wants to send some data to node V)
8. $k = n_u.P_v = n_v.P_u = n_u.n_v.G$ // k is secret key for encryption/ decryption
9. Node U takes the first 8 and last 8 bytes of the 48 bytes key k to get 16 bytes key
10. Input the data to be send of size at max 16 bytes as string s
11. Perform Add-Round Key operation on string s. In this step bitwise XOR operation is performed between the key and the string.
12. For (i=0 to 9) do{
13. Perform Sub-Byte operation on the string s. In this the 16 byte data is converted into a matrix M of size 4 x 4.
14. Perform Shift-Rows operation on the matrix M. In this the i^{th} row is circular right shift by i columns.
15. if (i=9) then skip 17
16. Perform Mix-Columns operation on the columns of matrix M. In this the values of i^{th} column is added by i.
17. Perform Add-Round Key operation on the matrix M. In this step the matrix M is converted into a string by concatenating row after row and then bitwise XOR operation is performed. }
18. Node V receives the encrypted data from node U
19. Node V takes the first 8 and last 8 bytes of the 48 bytes key k to get 16 bytes key
20. Perform Add-Round Key operation on the matrix M. In this step the matrix M is converted into a string by concatenating row after row and then bitwise XOR operation is performed between the key and the string.
21. For (i =0 to 9) do{
22. Perform Inv-Shift-Rows operation on the matrix M. In this the i^{th} row is circular left shift by i columns.
23. Perform Sub-Byte operation on the string s. In this the 16 byte data is converted into a matrix M of size 4 x 4.
24. if (i=9) skip 26
25. Perform Mix-Columns operation on the columns of matrix M. In this the values of i^{th} column is added by i.
26. Perform Add-Round Key operation on the matrix M. In this step the matrix M is converted into a string by concatenating row after row and then bitwise XOR operation is performed between the key and the string. }
27. Output the final decrypted message

Table 2 Contrast of key size of RSA and ECC

RSA key size (bits)	ECC key size (bits)
1024	160
2048	224
3072	256
7680	384
15360	521

6 Result and Analysis

To encrypt/decrypt data, we have used AES algorithm. It is one of the most powerful symmetric key algorithms. The time and space complexity of AES is O (1). The complexity of AES is constant because AES is a block cipher and it works for a specific number of steps on the fixed sized data, i.e. 128 bits.

We have used ECC (elliptic curve cryptography) for key generation and sharing. The time complexity of ECC key generation is also constant as it will take constant amount of time O (1) to generate the public and the private key of sender and receiver. The space complexity of ECC is considerably less as compared to the other most popular asymmetric key algorithm RSA. The key length is an important security parameter as NSIT provided different algorithms and their key sizes in which we have considered the comparison as shown in Table 2 of the key size of two algorithms RSA and ECC providing same level of security.

Thereby, we have made a hybrid encryption algorithm for Secure Data Communication in WSN. Our algorithm is exploiting the advantages of the two algorithms while avoiding their disadvantages.

7 Conclusion and Future Work

As the scope of wireless sensor networks is growing so, it is the need of an efficient algorithm which can provide Secure Data Communication in WSNs with the use of limited resources. In this paper, we have designed an efficient algorithm that can preserve the data integrity and confidentiality with minimum use of system resources. Our algorithm is exploiting the advantages of the two algorithms AES (advanced encryption standard) which is a symmetric key algorithm and ECC (elliptic curve cryptography) which is an asymmetric key algorithm. We are using ECC for key generation and sharing and AES for data encryption and decryption. The proposed hybrid algorithm is much secured than AES and less resource and time-consuming than ECC.

The future scope of work includes the following: we can verify this algorithm by comparing other algorithms based on overhead and will be designing an algorithm which is even more efficient and can be tried to enhance this algorithm for key generation and sharing.

References

1. P. Ganesan, R. Venugopalan and P. Peddabachagari, An Analysing and modelling encryption overhead for sensor network nodes, In Proceedings of the WSNA, pp. 151–159 (2003).
2. B. Przydatek, D. X. Song, and A. Perrig, SIA: secure information aggregation in sensor networks, Conference on Embedded Networked Sensor Systems (SenSys), pp. 255–265 (2003).
3. M. Panda, Security in Wireless Sensor Network using Cryptography Techniques, American Journal of Engineering Research (AJER), Vol. 3, pp. 50–56 (2014).
4. R. Pahal, V. Kumar, An Effective Implementation of AES, International Journal of Advanced Research in Computer Science and Software Engineering, Vol 3, pp. 290–295 (2013).
5. Mahimkar, S. Rappaport, Secure DAV: A secure data aggregation and verification protocol for sensor networks, The IEEE Global Telecommunications Conference, pp. 2175–2179 (2004).
6. S. Nath, P. B. Gibbons, S. Seshan, and Z. Anderson, Synopsis diffusion for robust aggregation in sensor networks. In Proc. of the 2nd international conference on Embedded networked sensor systems (2004).
7. Jelena Misic, Vojislav Misic, Wireless personal area networks performance interconnections and security with IEEE 802.15.4., John Wiley & Sons Ltd, (2008).
8. Adnan Nadeem and Michael P. Howarth, A Survey of MANET Intrusion Detection and Prevention Approaches for Network Layer Attacks, IEEE Communications Surveys and Tutorials, vol. 15, no. 4, (2013).
9. S. Ozdemir and H. Ichakawa et al. (Eds.), Secure and reliable data aggregation for wireless sensor networks, LNCS 4836, pp. 102–109 (2009).
10. Z. Dawahdeh, N.S. Yaakob, A New Modification for Menezes-Vanstone Elliptic Curve Cryptosystem. Journal of Theoretical and Applied Information Technology, Vol. 85, pp. 290–297 (2016).
11. M. Chu, H. Haussecker, and Zhao, Scalable information-driven sensor querying and routing farad ad hoc heterogeneous sensor networks, The International Journal of High Performance Computing Applications, Vol. 16(3), pp. 293–313 (2009).
12. A. Ganesh, An Improved AES-ECC Hybrid Encryption Scheme for Secure Communication in Cooperative Diversity Based Wireless Sensor Network, IEEE- International Conference on Recent Trends in Information Technology, Vol 11 (2011).
13. Z. Dawadeh, N.S. Yakob, Modified Elgamal Elliptic Curve Cryptosystem using Hexadecimal Representation, Indian Journal of Science and Technology, Vol 8(15), 2015.
14. H. Hayouni and M. Hamdi, Secure data aggregation in WSNs, Proc. of the 2nd World Congress on Computer Applications and Information Systems, Vol 5, pp. 58–65 (2012).
15. Rawya Rizk and Yasmin Alkady, Two-phase hybrid cryptography algorithm for wireless sensor networks, Journal of Electrical Systems and Information Technology, pp. 296–313, (2015), Available online at www.sciencedirect.com.
16. Y. Yang, X. Wang, S. Zhu, and G. Cao, SDAPA secure hop-by-hop data aggregation protocol for sensor networks, In Proc. of Acm Mobihoc, Vol. 5, pp. 55–60 (2006).
17. Tadiwa Elisha Nyamasvisva, Halabi Hasbullah, Multi-level security algorithm for random ZigBee wireless sensor networks, Information Technology (ITSim), International Symposium in Kuala Lumpur, IEEE, vol 2, 2010, pp. 612–617, (2010).