P-ISSN 1859-3585 | E-ISSN 2615-9619

A NOVEL SECURITY ROUTING USING IDENTITY-BASED LIGHTWEIGHT DIGITAL SIGNATURE IN WMSNs

ĐỊNH TUYẾN BẢO MẬT MỚI SỬ DỤNG CHỮ KÝ SỐ NHẸ DỰA TRÊN DANH TÍNH TRONG WMSN

Tran Huy Long[1,*], Tran Thien Chinh[1], Tran Hoai Trung[2], Pham Van Vinh[3]

DOI: https://doi.org/10.57001/huih5804.2023.217

ABSTRACT

This paper presents a lightweight digital signature-based routing message and node authentication solution for wireless multimedia sensor networks (WMSNs). The lightweight geographical security routing protocol (ECDSA-TPGF) is proposed on the basis of improving the original two-phase greedy geographical forwarding protocol (TPGF), in which we have added the solution Elliptic curve cryptography (ECC) and cyclic redundancy check (CRC) to create a digital signature for a sensor node attached to a routing message. This ensures that only sensor nodes and messages from trusted sources are accepted and participate in the routing process. By using lightweight authentication methods and efficient encryption algorithms, the ECDSA-TPGF routing protocol helps prevent node spoofing attacks and ensures the security of information transmitted over the network. The effectiveness of the algorithm was confirmed through security analysis and evaluated by simulation.

Keywords: *CRC, ECC, ECDSA, Secu-TPGF, GSTP, MD5, SHA-3, MAC, Routing, WMSN.*

TÓM TẮT

Bài báo trình bày một giải pháp xác thực nút và bản tin định tuyến dựa trên ký số nhẹ cho mạng cảm biến không dây đa phương tiện (WMSNs). Giao thức định tuyến bảo mật nhẹ theo địa lý (ECDSA-TPGF) được đề xuất trên cơ sở cải tiến giao thức định tuyến chuyển tiếp địa lý tham lam hai giai đoạn (TPGF) ban đầu, trong đó chúng tôi đã bổ sung giải pháp mật mã hóa đường cong Elliptic (ECC) + kiểm tra dư thừa theo chu kỳ (CRC) để tạo nên chữ ký số cho một nút cảm biến được đính kèm bản tin định tuyến. Điều này đảm bảo rằng chỉ các nút cảm biến và bản tin từ các nguồn tin cậy mới được chấp nhận và tham gia vào quá trình định tuyến. Bằng cách sử dụng các phương pháp xác thực nhẹ và thuật toán mã hóa hiệu quả, giao thức định tuyến ECDSA-TPGF giúp ngăn chặn các cuộc tấn công giả mạo nút và đảm bảo tính bảo mật của thông tin truyền qua mạng. Hiệu quả của thuật toán đã được xác nhận thông qua phân tích bảo mật và đánh giá bằng mô phỏng.

Từ khóa: *CRC, ECC, ECDSA, Secu-TPGF, GSTP, MD5, SHA-3, MAC, Routing, WMSN.*

[1]Posts and Telecommunications Institute of Technology, Vietnam
[2]University of Transport and Communications, Vietnam
[3]Authority of Information Security, Vietnam
[*]Email: longth@ptit.edu.vn
Received: 15/9/2023
Revised: 20/10/2023
Accepted: 25/11/2023

1. INTRODUCTION

As the use of multimedia sensor nodes can enhance the capabilities of wireless sensor networks (WSNs) in event description, WMSNs have been attracting the attention of many researchers. Among them, the TPGF routing protocol [2] is one of the effective communication protocols for multimedia streams that meet three requirements [1]: i) Multipath transmission; ii) Transmission through holes, and iii) Shortest transmission path. However, the TPGF protocol is not designed to withstand various attacks at the network layer (in particular, attacks against routing protocols) that can disrupt the entire network operation.

Recent studies aimed at authentication nodes and routing messages have been proposed based on the routing mechanism of the original TPGF protocol, such as the SecuTPGF protocol proposed in [3] which used authentication tokens message (MAC) to authenticate the origin and protect mutable information in the routing message, however this will incur a high computational cost. Or by using the MD5 hash function in the geo-secured two-phase routing protocol (GSTP) [4] and the SHA-3 hash function in the secure routing protocol (GSR) [5] to provide authentication of both the node and the message, allowing confidentiality of the identity of the 1-hop node and routing through that 1-hop node. Comparing MD5 and SHA-3 [6] with different parameters such as cost, message length, speed, and attacks has shown that SHA-3 is more secure than MD5. MD5 is faster than SHA-3 thanks to reduced circuit, but with low computational power SHA-3 can perform better on small devices like sensor nodes.

It can be said that MAC is considered more secure than MD5 because MD5 has discovered security vulnerabilities [7]. Attacks using collision attacks on the MD5 hash value have allowed hackers to create two messages with the same hash value, which can lead to deception in the process of checking the authenticity of data. Meanwhile, MAC does not have the same security vulnerabilities and is considered more secure. However, MAC can be slower than MD5 because it uses a secret key to generate the authentication code, while MD5 only uses a data hash algorithm. As for MAC and SHA-3, both are encryption tools used to protect data integrity. However,

they have different purposes and applications. MAC is used to confirm the authenticity of data and ensure that it has not been modified during transmission. MACs are typically generated using a hash algorithm such as SHA-3 along with a secret key. For requiring authentication of data and ensuring that it has not been modified, MAC is a good choice. However, if you just need to check the integrity of the data and ensure that it has not been altered, then SHA-3 may be a better choice because it provides a unique hash value for each set of data inputs.

Clearly, node authentication and proper use of routing messages play an important role in ensuring the integrity and security of the network. They help prevent various threats, such as Sybil attacks, sinkhole attacks, routing information changes, etc., and ensure that information is forwarded to the correct destination and on the optimal path. To ensure information integrity and anti-repudiation of transactions in WSN, the authenticity of the sending node needs to be verified. In this case, a digital signature is added to the routing messages sent by the source node for authentication. However, choosing the appropriate digital signing algorithm is a challenging task for the designer because of the inherent characteristics of WSN, such as limited resources, low computing power, and small storage capacity. Therefore, a secure and efficient routing protocol needs to be designed to prolong the lifetime of the network while preventing as many attacks as possible. Accordingly, research on lightweight encryption aims to create installation solutions that are very compact but do not reduce safety and security too much. It is a solution offered to compromise between security and efficiency in the implementation of encryption algorithms.

There are many types of digital signature algorithms that can be used in WSNs, such as the elliptic curve digital signature algorithm (ECDSA), RSA, OTS, lightweight digital signatures, and identity digital signatures. Each algorithm has its own advantages and disadvantages when used for WSN [8]. The author also experimentally showed that using ECDSA can save energy compared to other public key algorithms. Furthermore, ECDSA offers significant communication benefits thanks to smaller keys. With a given amount of power, ECDSA-160 can perform 4.2 times more key exchange operations than RSA-1024. The average energy cost of signature generation in ECDSA-112 is exactly the same as RSA-1024; however, RSA-512, RSA-1024, and RSA-2048 have better signature verification results. ECDSA-224 is the best algorithm to achieve the highest level of security. The running digital signature algorithm consumes more power when the battery of the sensor nodes is low. Using digital signatures in multi-hop WSNs introduces additional overhead due to network protocol waiting time. Therefore, the security protocol must be designed in a way that reduces latency to the lowest possible level.

A lightweight digital signature algorithm (LWDSA) was proposed [9] with the aim of developing a lightweight authentication protocol using the MBLAKE2b hash function combined with the ECDSA algorithm. The proposed algorithm has proven to be secure in all authentications. The authentication framework contains both one-way authentication and mutual authentication built for WSNs using digital signatures without using certificates. This increases network lifetime and reduces computation time.

Another lightweight digital signature algorithm was first introduced by Shamir [10] (identity-based cryptography, ID), which eliminates the need to check the validity of the certificate. In ID-based cryptography, each user's public key can be easily calculated from a string corresponding to this user's identity (e.g., email address, phone number, etc.). The private key generator (PKG) then calculates the private keys from a master secret key for the user. This attribute avoids the requirement to use a certificate and associate an implicit public key (user identity) for each user in the system. In the case of ID-based signatures (IBS), verification takes only the identity along with the message and signature pair as input and executes the algorithm directly. This differs from traditional public key cryptography, while an additional certificate verification algorithm is required that is equivalent to the two-signature verification process.

Clearly, research into lightweight cryptography aims to create compact implementation solutions that do not compromise security too much. It is a solution that offers a compromise between security and efficiency in the setting of cryptographic algorithms. Therefore, more in-depth research is needed to keep up with and match the rapidly developing needs of WMSN applications.

For a long time, the CRC hash function has been widely used in communication protocols such as Ethernet, Wi-Fi, Bluetooth and many others. It was created in 1961 and initially used to identify unintentional changes to information sent through the communication medium with the purpose of saving energy [11]. Today, there are many variations to generate a more secure and efficient CRC, such as using the product of smaller irreducible polynomials that are easier to calculate instead of a single polynomial to generate the CRC. The author [12] has proven that the scheme using reduced polynomials in the hash function construction process is very suitable for short messages. Also, according to research by author Elena Dubrova [13], most link layers use CRC only to protect against accidental modifications during transmission. Data integrity protection can be achieved using some n-bit message authentication code, e.g., HMAC keyed hash message authentication code, KECCAK KMAC message authentication code, or KMAC message authentication code. authenticate the CBC-MAC cryptographic blockchain message. However, such an approach extends the message by n bits and requires a separate encryption/decryption engine that is more complex than the CRC encryption/decryption engine. For example, it was shown in [14] that KMAC128 occupies 45 times more storage space and consumes 28 times more energy than the 128-bit CRC-based MAC algorithm [15].

Therefore, the paper proposes a modified version of the TPGF protocol called ECDSA-TPGF that uses a combination of both CRC and ECC methods to generate lightweight ID-based digital signatures.

The rest of the paper is organized as follows: Part 2 presents a lightweight ID-based digital signature scheme. Part 3 presents the network model, routing attacks, and network performance measurements. Part 4 presents the ID-based lightweight secure routing protocol used for WMSN. Part 5 presents simulation and evaluation. Part 6 concludes, and future development directions.

2. LIGHTWEIGHT ID-BASED DIGITAL SIGNATURE SCHEME

A group of sensor nodes will be managed and transmit the collected data to the sink node, which will be responsible for being the root node, initializing the network, and establishing the transmission route. During deployment, the sink node will obtain its ID and key information for authentication purposes. Each button will have a unique ID specified by the manufacturer. Accordingly, the proposed solution is a lightweight digital signature based on ID, as shown in Figure 1.

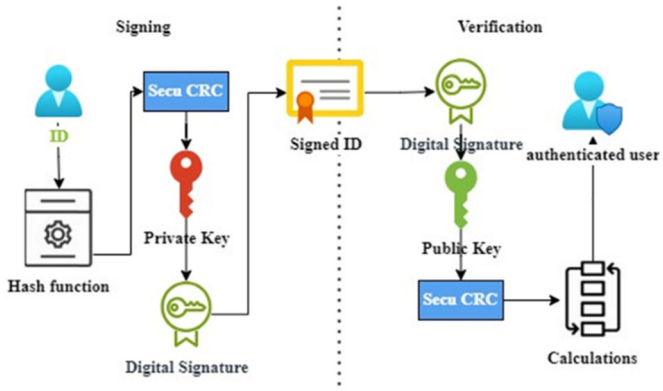

Figure 1. ID-based digital signature scheme

2.1. Generate key

- Choose an elliptic curve using a 128-bit key of the form $y^2 = x^3 + ax + b \bmod p$ with $0 \le x < p$. The constants a and b are non-negative integers smaller than the prime number p and must satisfy the following conditions:

$$4a^3 + 27b^2 \bmod p \neq 0 \quad (1)$$

- Choose an initial point G belonging to that elliptic curve, used for scalar multiplication on the curve. During network initialization, this point is made public throughout the network.

- Then each node (for example node A) when needing to forward the message will choose a private key by choosing a random number $n_A < p$. This private key is responsible for creating a digital signature to prove that data belongs to the owner of the private key.

- Unlike the private key, the public key is public to everyone and is calculated by $P_A = n_A G$. Elliptic curve multiplication is a "trap door" operation, meaning it is easy to calculate in one direction and impossible to calculate in the other direction. Therefore, the owner of the private key can easily create a public key and safely share it with everyone without worrying that someone can reverse the public key to take over their private key.

2.2. Create digital signature

- Each ID will be converted into binary bits and then hashed using CRC.

To perform CRC encryption [12], the message polynomial M(x) is first multiplied by x^n, where n is the degree of the generating polynomial p(x). Then, the result is divided modulo by the generating polynomial p(x). The coefficients of the result form the check bits of the CRC:

$$r(x) = M(x).x^n \bmod p(x) \quad (2)$$

These check bits are added to the message to form the CRC codeword:

$$M(x).x^n \oplus r(x) \quad (3)$$

where "⊕" is the XOR operation.

- ECC is then used to create a digital signature on the CRC check value (h).

To perform a digital signature, proceed as follows:

+ Choose a random number k in the range [1…p-1];

+ Calculate random point R = k * G and get its x coordinate:

$$r = R.x \quad (4)$$

+ Calculate digital signature by:

$$s = k^{-1} * (h + r * n_A) (\bmod p) \quad (5)$$

where k-1 is the modular inverse (also an integer) of k such that:

$$k * k^{-1} \equiv 1 \quad (6)$$

- Returns the digital signature {r,s}

The computed signature {r, s} is a pair of integers, each in the range [1…p-1]. It encrypts the random point R = k * G, together with the proof s, confirming that the signer knows the message h and the private key. The proof s can be verified using the corresponding public key.

2.3. Verify digital signature

Any node in the network can verify node A's digital signature using the shared public key P_A.

- Calculate the inverse of the signature proof module:

$$s1 = s^{-1} (\bmod p) \quad (7)$$

- Restore the random point used during the signing process by:

$$R' = (h * s1) * G + (r * s1) * P_A \quad (8)$$

- Get from R' its x coordinate:

$$r' = R'.x \quad (9)$$

- Calculate the authentication result by attaching the new CRC code obtained from calculating the coordinates of R' to the ID, the authentication process through CRC decoding is performed by dividing the modulus of the

received message by the initial polynomial p(x) and comparing the coefficients of the obtained remainder with the received CRC check bits. An error will appear if the test results are not the same.

3. NETWORK MODELS, ROUTING ATTACKS, AND NETWORK PERFORMANCE MEASUREMENTS

3.1. Network model and assumptions

WMSN is considered to be a fixed set of sensor nodes that can be represented as a graph G(V, E), where $V = \{v_1, v_2, \ldots, v_n\}$ is a finite set of nodes. sensor node (vertex) and $E = \{e_1, e_2, \ldots e_n\}$ is a finite set of links (edges). Each sensor node has a transmission radius of TR and M 1-hop neighboring sensor nodes. The sink node is reliable and not resource-constrained. It is assumed that each node can maintain a certain amount of time before it is compromised. Sensor nodes are untrusted, which is a common assumption in WSNs because an adversary can capture and compromise sensor nodes relatively easily. Finally, we use the proposed solution to increase the effectiveness of the network under attack.

3.2. Routing attacks

Adversary nodes present inside and outside the network interfere with the routing protocol. In this section, the article will discuss possible attacks during routing, including:

- Spoofed Routing Attacks: Attackers can spoof routing information to redirect data traffic or attack sensor nodes in the WSN. This can lead to routing errors or data integrity disruption.

- Spoofing Attacks: An attacker can spoof the identity of a sensor node or transmit fake information to the WSN. This can lead to routing errors or disrupt data integrity and authenticity.

- Sybil attack: An attacker creates multiple fake identities (called Sybil nodes) and injects them into the WSN with the intention of cheating or disrupting the routing process.

- Wormhole attacks: The attacker creates a virtual communication channel (wormhole link) by quickly transferring packets from one node location to another location in the WSN, causing a time delay.

- Flooding attack: The attacker sends a large number of messages or packets to the WSN, causing overload and disrupting the normal operation of the WSN.

- Selective Forwarding attack: The attacker selects some specific packets to forward or send, while other packets are filtered or sent incorrectly. This causes information loss and affects the integrity and reliability of data in the WSN.

3.3. Network performance evaluation parameters

Among the parameters for evaluating network performance, the lifetime parameter is one of the parameters of primary concern to researchers of conventional WSNs, however, with WMSN networks, the end-to-end delay is) and path length are often considered to evaluate the performance of routing algorithms [3]. These parameters can be described as follows:

- End-to-end delay is the time required to transmit information from the source node to the sink node. The average delay of each hop is $D_{hope} + D_{otherfactors}$.

$$D_{e2e} = k \times \left(D_{hope} + D_{otherfactors}\right) \quad (10)$$

where k is the number of transmission hops, D_{hop} is the delay during transmission and $D_{otherfactor}$ is the delay based on other factors. For each transmission hop, the average delay ($D_{hop} + D_{otherfactors}$) is a fixed value. Then:

$$D_{e2e} \alpha k \quad (11)$$

From (5), the end-to-end delay is proportional to the number of hops k. If the number of hops is less, the end-to-end delay is reduced, meaning the time needed to transmit information will also be reduced.

- The path length is calculated as the sum of the weights associated with each visited link. Some routing protocols use hop counts to determine the number of relay nodes a packet must pass through from the source node to the destination node.

$$P_{Length} = k(\text{number of hops}) \quad (12)$$

4. SECURE ROUTING FOR WMSNS USING LIGHTWEIGHT IDENTITY-BASED DIGITAL SIGNATURE

Our proposed solution includes three phases: (i) network setup; (ii) discovering safe 1-hop nodes; (iii) communication via secure 1-hop nodes.

4.1. Network setup

The WSN manager, who has the authenticating authority (base station), deploys the network and performs the initialization process using its own infrastructure to minimize the power consumption of other nodes. After deploying the sensor network, the identification (ID) of each sensor node will be processed by the base station. The process of creating a lightweight ID-based digital signature by combining CRC and ECC is as discussed in section 2. The generated digital signature is stored as an attribute in the sensor node as shown in Figure 2.

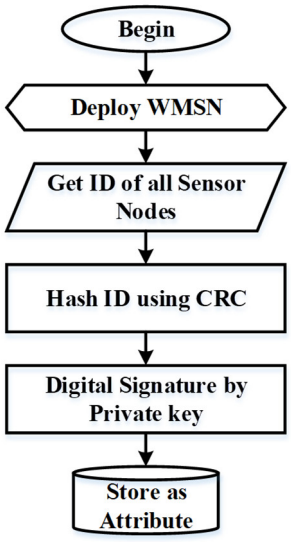

Figure 2. Flow graph of network setup

After completing the deployment and setup phase, the next phase, the discovery of secure 1-hop nodes, is initiated by the source node of the network.

4.2. Discover secure 1-hop nodes

By discovering secured 1-hop nodes, adversary nodes are prevented from joining the WSN; only authenticated nodes are allowed to join the network in the first stage. Digital signatures are used to encrypt the sent message (Figure 3). This process is described as follows:

After deploying sensor nodes in the network, each node tries to discover its one-hop nodes by broadcasting a message consisting of the identity (ID) information and the attached digital signature, Geolocation (GL) (as shown in Figure 3). Then, it waits for each 1-hop neighbor to respond.

The algorithm for discovering secure 1-hop nodes is as follows:

Step 1: Each node will broadcast a broadcast message to the nodes in the network. For example, Node A will send a message like this:

$$a \rightarrow * : \mathrm{HELLO}(\mathrm{ID_A} + \mathrm{Sig_A}, \mathrm{GL_A}) \quad (13)$$

where $\mathrm{ID_A}$ is the identity of node A and has an additional $\mathrm{Sig_A}$ digital signature as described in Section 4.1. $\mathrm{GL_A}$ is the location of the relay node.

Step 2: Neighbor node B will use PA to decrypt the digital signature and verify whether node A is in the storage list by dividing this message containing IDA+CRC by p(x) (discussed above). If this value is 0, it means that IDA + CRC matches the value in the archive. Then node B will send a message containing node B's ID and location to A.

$$B \rightarrow A : (\mathrm{ID_B} + \mathrm{CRC_B}, \mathrm{GL_B}) \quad (14)$$

Step 3: Node A receives this message and also confirms and stores it as a one-hop neighbor of each other.

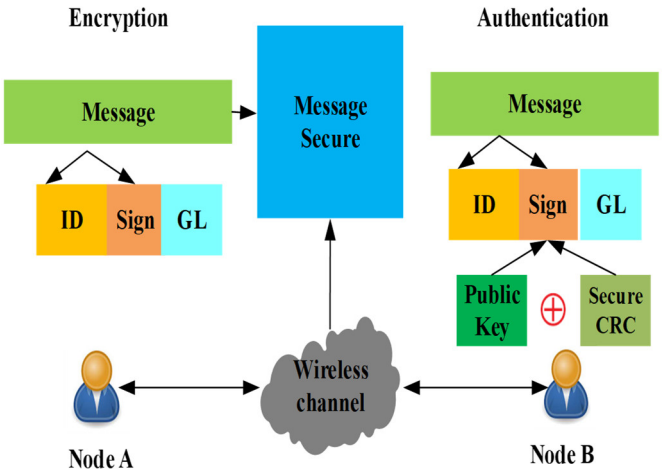

Figure 3. Proposed solution model

Every sensor node in the WSN verifies that the neighbor is a secure 1-hop node, establishes a secure link, and adds the node to its secured 1-hop neighbor list, as shown in Figure 4.

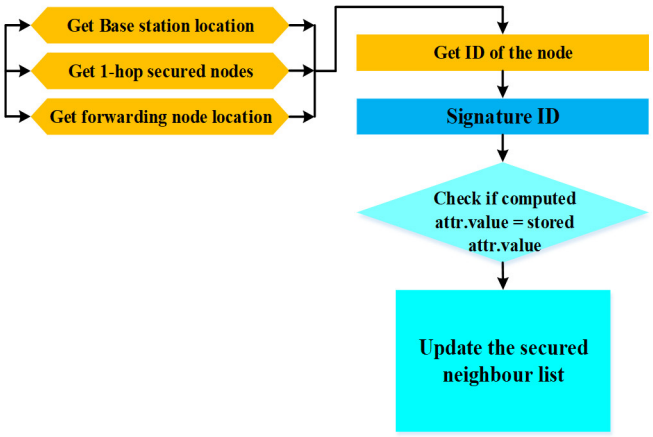

Figure 4. Flow graph of discovering secured 1-hop nodes

4.3. Transmission through secured 1-hop nodes

The source node initiates the routing process and forwards a request to the nearest secure 1-hop node among all identified secured 1-hop neighbors to the base station. When the relay node receives the request, it verifies that it has a secured 1-hop node to transmit. If it has a secured 1-hop node to transmit, it forwards a request to the next relay node or base station. If multiple one-hop security nodes are identified, it proceeds to select the one-hop node closest to the base station; otherwise, it is marked as a 'blocking' situation, and it returns to the previously secured 1-hop node and marks itself and is ignored. As shown in Figure 5, backtracking and marking are performed iteratively to determine the next secured 1-hop node for greedy forwarding. The number-based label is assigned to the specified 1-hop security node along with the path number.

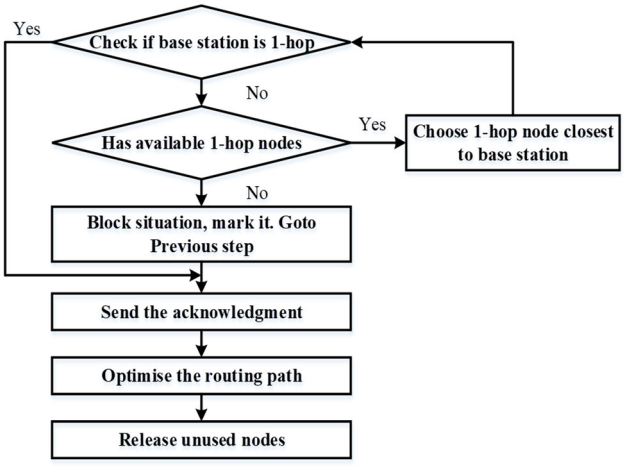

Figure 5. Flow graph of secured forwarding and transmission

An acknowledgment is returned to the source node from the base station when a routing path is determined. Acknowledgments are sent via 1-hop secure nodes with the largest number of nodes and the same number of paths. During backpropagation on the defined path, optimization is performed at each intermediate node to eliminate path circles. After receiving the confirmation with the pre-assigned path number, the source node starts transmitting

multimedia data. At the same time, a release command is executed for all other 1-hop nodes that are not participating in the transmission.

5. SIMULATION AND EVALUATION

5.1. Simulation setup

To evaluate and analyze the proposed ECDSA-TPGF protocol, we chose the Nettopo simulator, designed specifically for the TPGF protocol by the research group [17, 18]. The ECDSA-TPGF routing algorithm is built on the basis of the TPGF routing protocol. ECDSA-TPGF generates ID-based digital signatures using two algorithms, CRC and ECC, to provide security. Its performance is compared with the previous SecuTPGF protocol, built with user security algorithms determined based on various network metrics such as the number of routing paths and average path length.

Table 1. Simulation parameters ([3, 4 ,5])

Parameters	Value
Network size	640 x 400m
Number of sensor nodes	100 - 1000
Number of base station	1
Number of source nodes	1
Initial Energy of sensor nodes	10J
Transmission radius	60 - 120m
Expected lifetime	1 - 14h

Accordingly, the simulated network size is fixed at 640 × 400. The average number of hops and average number of paths are calculated by varying the number of nodes (from 100 to 1000) to obtain different values. The simulation parameters are set in the table below:

Sink button: (ID: 1; Location: 12.56; Max TR: 60; Bandwidth: 1); The power button is a red button with parameters: (ID: 2; Energy: 10J; Location: 620,211; Max TR: 60; Bandwidth: 1; Expected lifetime: 1); Purple sensor nodes and attack nodes (25% of sensor nodes) are randomly arranged in the network, as shown in Figure 6.

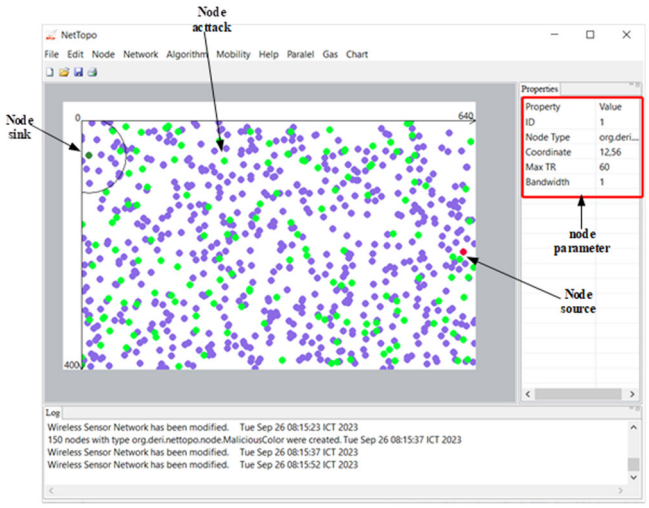

Figure 6. Network setup

Run the SecueTPGF and ECDSA-TPGF algorithms in turn and compare.

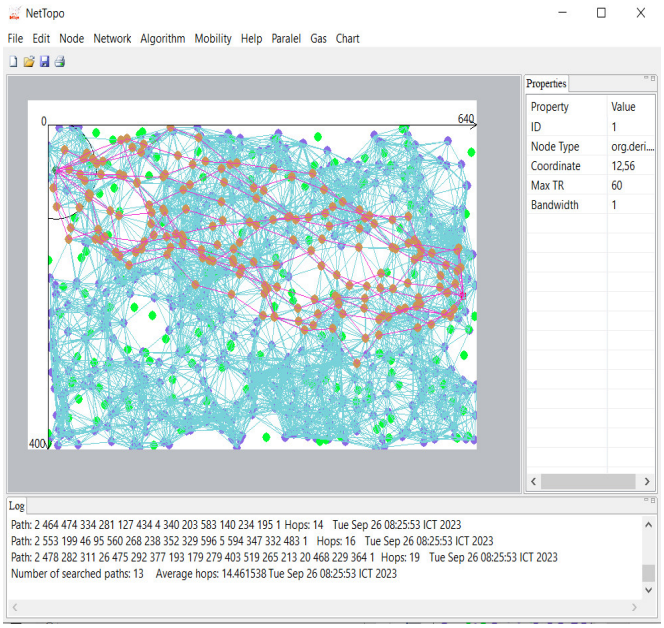

Figure 7. Results of simulation execution

Figure 7 shows the implementation of ECDSA-TPGF in NetTopo; attack nodes are not included in the transmission paths. When the intermediate nodes receive the routing request, it checks whether the base station is in 1-hop; if so, it constructs the route and sends an acknowledgment. If it is an intermediate 1-hop node, it simply forwards it to the next secure 1-hop node. This is repeated until the base station is reached.

5.2. Evaluate

Table 2 compares the simulation results of the calculated average number of hops before and after optimization in finding routing paths using the SecuTPGF and ECDSA-TPGF algorithms.

Table 2. Average number of hops

Number of nodes	Before optimization		After optimization	
	SecuTPGF	ECDSA-TPGF	SecuTPGF	ECDSA-TPGF
100	0	17	0	14
200	23	22	18	17
300	24	21	17	14
400	22	20	17	15
500	20	17	16	13
600	19	18	16	15
700	18	15	16	13
800	18	17	16	15
900	20	17	15	12
1000	19	19	14	12

The average number of stops before and after optimization is obtained as shown in Figures 8 and 9.

P-ISSN 1859-3585 | E-ISSN 2615-9619

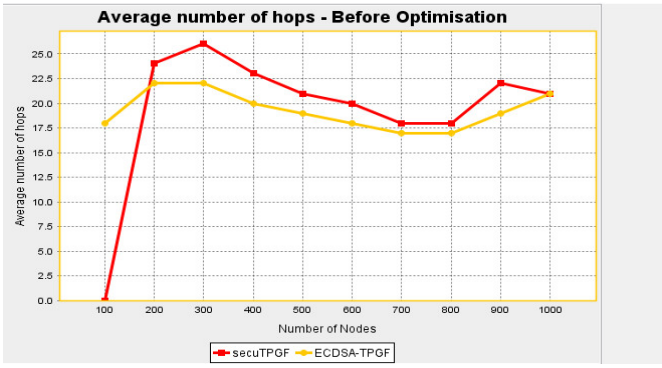

Figure 8. Average number of hops using SecuTPGF and ECDSA-TPGF before optimization

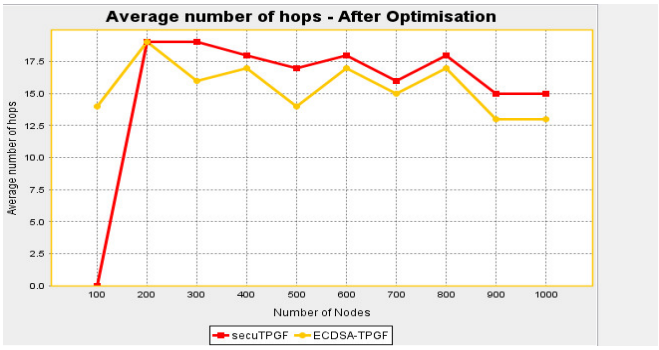

Figure 9. Average number of hops using SecuTPGF and ECDSA-TPGF after optimization

Expanding on a similar setup with the GSTP and GSR protocols, the results are obtained as shown in Figures 10 and 11.

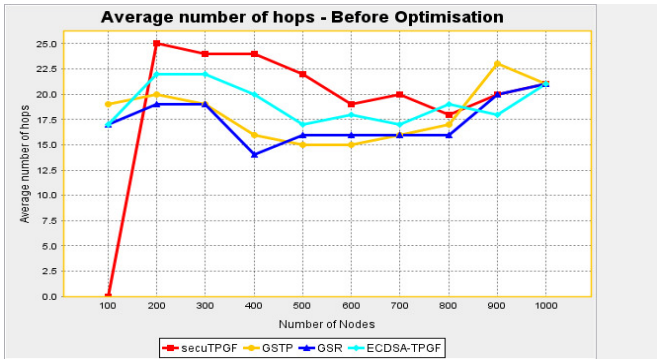

Figure 10. Average number of hops using SecuTPGF; GSTP, GSR and ECDSA-TPGF before optimization

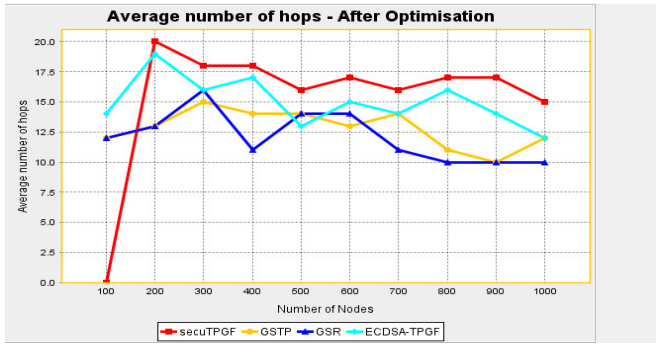

Figure 11. Average number of hops using SecuTPGF; GSTP, GSR and ECDSA-TPGF after optimization

Simulation results show that the average number of hops found for the ECDSA-TPGF protocol tends to decrease compared to SecuTPGF thanks to the application of lightweight algorithms while still ensuring attack prevention capabilities. However, due to applying two different algorithms, the ECDSA-TPGF protocol is still slower than GSTP and GSR.

When increasing the transmission distance of sensor nodes from 60 to 120, the average number of hops decreases proportionally (as shown in Figure 12).

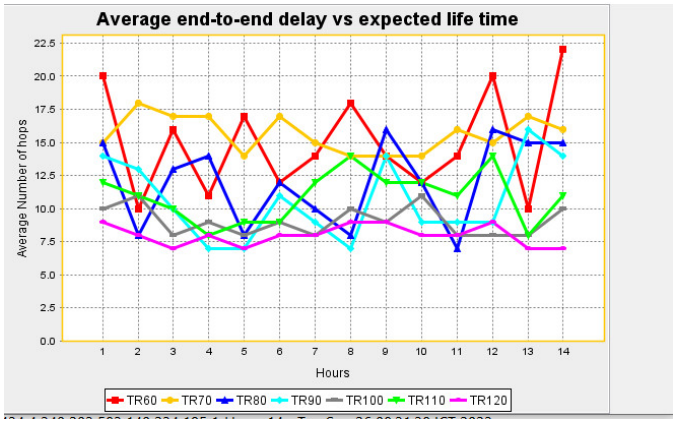

Figure 12. Average end-to-end latency compared to expected usage time

6. DISCUSSION

It is clear that the lightweight digital signature solution proposed in this article is strong enough to protect data from external attacks specifically:

- Digital signatures are established based on each node's ID, helping to prevent tampering and falsification of data, and providing an authentication method without affecting the security of the original data.

- ECDSA-TPGF is enough to ensure routing message security as proposed:

+ Short key length: Compared to traditional digital signature algorithms such as RSA, ECDSA requires a shorter key length. This means faster signature generation, data processing and transmission. ECDSA can provide the same level of security with shorter key lengths, which saves resources and increases processing speed.

+ High security: ECDSA provides data integrity and authenticity. ECDSA digital signatures are generated using a private key and a public key associated with an elliptic curve. This algorithm ensures confidentiality and resistance to intrusion and determines the origin of data.

+ High performance: ECDSA is a high-performance algorithm. Due to the special properties of elliptic curves, performing operations on elliptic curves is faster than operations in traditional cryptographic systems such as RSA. This helps increase processing speed, especially in resource-constrained network applications such as wireless sensor networks (WSN) and mobile devices.

+ Small size: ECDSA signature size is smaller than RSA. Therefore, transmitting and storing ECDSA signatures saves

space and bandwidth. This makes ECDSA suitable for applications with limited resources and small size requirements such as IoT (Internet of Things)/WSN and mobile applications.

7. CONCLUSION

This paper proposes a new secure routing protocol that uses digital signatures based on each node's ID, so each message sent from a sensor node can be digitally signed to provide information about the origin and status integrity of the message. ECDSA-TPGF can identify attack nodes using a combination of a lightweight CRC hash function and a lightweight and secure ECC asymmetric key algorithm. Simulation results confirm that ECDSA-TPGF is robust in detecting and excluding adversarial nodes.

In the future, we will continue to consider solutions such as key sharing and lightweight hash functions that help increase network lifetime while still ensuring WMSN security.

REFERENCES

[1]. L. Shu, Y. Zhang, L. Yang, Y. Wang, M. Hauswirth, 2008. *Geographic Routing in Wireless Multimedia Sensor Networks.* In Proceedings of Second International Conference on Future Generation Communication and Networking, FGCN '08, Hainan Island.

[2]. Lei Shu, Yan Zhang, Laurence T. Yang, Yu Wang, Manfred Hauswirth, Naixue Xiong, 2010. *TPGF: geographic routing in wireless multimedia sensor networks*. Telecommun Syst 44: 79–95.

[3]. Taye Mulugeta1, Lei Shu, Manfred Hauswirth, Min Chen, Takahiro Hara, Shojiro Nishio, 2010. *Secured Two Phase Geographic Forwarding Protocol in Wireless Multimedia Sensor Networks.* 2010 IEEE Global Telecommunications Conference GLOBECOM 2010.

[4]. B. Prathusha Laxmi, A. Chilambuchelvan, 2016. *GSTP: Geographic Secured Two Phase Routing Using MD5 Algorithm*. Circuits and Systems, 7, 1845-1855.

[5]. B. Prathusha Laxmi, A. Chilambuchelvan, 2017. *GSR: Geographic Secured Routing using SHA-3 algorithm for node and message authentication in wireless sensor networks*. Future Generation Computer Systems 76, pp 98–105.

[6]. Piyush Gupta, S. K., 2014. *A comparative analysis of SHA and MD5 algorithm*. Int. J. Comput. Sci. Inf. Technol, p. 4492–4495.

[7]. Dan Kaminsky, 2004. *MD5 To Be Considered Harmful Someday.* Senior Security Consultant, Avaya.

[8]. Pankaj Kumar, Saurabh Kumar Sharma, 2021. *An Empirical Evaluation of Various Digital Signature Scheme in Wireless Sensor Network*. IETE Technical Review.

[9]. M Layvanya, V Natarajan, 2017. *LWDSA: light-weight digital signature algorithm for wireless sensor networks*. Indian Academy of Sciences, DOI 10.1007/s12046-017-0718-5.

[10]. A. Shamir, 1984. *Identity-based cryptosystems and signature schemes*. In Proc. CRYPTO '84, volume 196 of Lecture Notes in Computer Science, pages 47–53. Springer-Verlag.

[11]. Raghini Sharma, Umarani Chellapandy, 2022. *A Survey on Encrypted and Decrypted Text Algorithm Using CRC, SHA-256, MD5 and Caesar Cipher*, International Journal of Trend in Scientific Research and Development (IJTSRD) Volume 6 Issue 2.

[12]. Elena Dubrova, Mats Naslund, Goran Selander, Fredrik Lindqvist, 2015. *Cryptographically Secure CRC for Lightweight Message Authentication*. Computer Science, Mathematics IACR Cryptol.

[13]. Elena Dubrova, Mats Näslund, Goran Selander, Fredrik Lindqvist, 2018. *Lightweight Message Authentication for Constrained Devices*. WiSec '18: Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Network.

[14]. Yang Yu. 2017. *Evaluation of Cryptographic CRC in 65nm CMOS*. M. Sc. Thesis, Royal Institute of Technology (KTH), Sweden.

[15]. Elena Dubrova, Mats Naslund, Goran Selander, Fredrik Lindqvist, 2018. *Message Authentication Based on Cryptographically Secure CRC without Polynomial Irreducibility Test*. Cryptography and Communications 10, 383-399. Issue 2.

[16]. S. Vasundhara, 2017. *The Advantages of Elliptic Curve Cryptography for Security*. Global Journal of Pure and Applied Mathematics, ISSN 0973-1768 Volume 13, Number 9, pp. 4995-5011.

[17]. L. Shu, C. Wu, M. Hauswirth, 2008. *NetTopo: Beyond Simulator and Visualizer for Wireless Sensor Networks*. Technical Report of Digital Enterprise Research Institute.

[18]. L. Shu, M. Hauswirth, H.-C. Chao, M. Chen, Y. Zhang, 2011. *NetTopo: A framework of simulation and visualization for wireless sensor networks*. Adhoc Networks, vol. 9, p. 799–820.

THÔNG TIN TÁC GIẢ

Trần Huy Long[1], Trần Thiện Chính[1], Trần Hoài Trung[2], Phạm Văn Vĩnh[3]

[1]Học viện Công nghệ Bưu chính Viễn thông

[2]Trường Đại học Giao thông Vận tải

[3]Cục An toàn thông tin