DOI 10.15276/imms.v14.no3.123
УДК 004.04

Informatics and Mathematical Methods in Simulation
Vol.14 (2024), No.3, pp. 123-133

DEVICE AUTHENTICATION METHOD IN INTERNET OF THINGS NETWORKS

A.Ya. Davletova

West Ukrainian National University
11, Lvivska Str. Ternopil, 46009, Ukraine
Email: a7davletova@gmail.com

This work addresses the pressing issue of ensuring secure and reliable data transmission, along with efficient cryptographic key management in IoT networks. Typical authentication protocols, though providing secure communication, can be overly complex for many resource-constrained devices. This highlights the need to explore and identify efficient solutions suited to the capabilities of IoT devices, ensuring robust encryption and authentication. The paper presents a hybrid authentication algorithm, RHAA (RSA and Hamming Code-based Authentication Algorithm), designed to guarantee confidentiality, integrity, and authentication of data in IoT networks with numerous nodes. The key feature of the proposed algorithm is the combination of RSA-based asymmetric encryption with the error-correcting capabilities of Hamming code in finite fields. This approach ensures device authentication during information exchange through centralized key generation and management. Data protection from unauthorized access is achieved through re-encryption at each node. Data integrity is maintained by detecting and correcting errors at each transmission stage. This solution enhances IoT network security by reducing the risk of data leakage or loss. The work also includes an example of the RHAA algorithm implementation with real key values, demonstrating the system's response to errors and their correction. The proposed algorithm, optimized for resource-limited devices, can be applied to improve data protection in IoT networks.

Keywords: authentication, IoT networks, RSA algorithm, Hamming code in finite fields, encryption, data integrity, error detection and correction, privacy, data security.

Introduction. With the development of information systems and technologies, the importance of implementing effective methods for information protection and secure data exchange is becoming increasingly apparent. The active digitalization of key sectors of activity [1], the widespread use of Internet of Things (IoT) technologies, and the growing reliance on cloud services with remote access significantly increase the risks of misuse and cyberattacks [2].

The growing number of connected devices across various fields, from smart home automation to industrial systems, necessitates ensuring secure communication. The decentralized architecture complicates control over network devices and creates new opportunities for exploiting system vulnerabilities without the proper level of protection [3]. In such conditions, to prevent unauthorized access and avoid attacks aimed at spoofing or exploiting malicious devices, authentication mechanisms based on cryptography become crucial. The use of robust cryptographic methods ensures data confidentiality and the identification of communication participants. The properties of error-correcting codes guarantee the integrity of transmitted messages.

Many IoT devices are characterized by certain hardware limitations, such as low computational power, limited energy supply, and memory capacity [4]. This necessitates addressing the issue of managing security mechanisms, particularly cryptographic keys, including their generation, storage, and processing. A hybrid algorithm that combines various cryptographic methods and centralized security management could become an effective solution for enhancing the integrity of data transmission and the security of IoT device authentication.

Analysis of research and publications. IoT devices play an important role in data collection and processing across various fields, such as healthcare, industry, home automation, and more.

However, their widespread use exposes new vulnerabilities that can be exploited to gain unauthorized access to devices, intercept data, or perform other malicious actions [5]. Typically, IoT devices can be configured via remote control or through proprietary controllers developed by manufacturers [6].

This creates opportunities for sending malicious instructions directly to the device. Moreover, data from certain IoT devices are transmitted to edge servers for further processing, opening new channels for attacks [7].

By using specific types of attacks, such as radio attacks or reverse engineering attacks, malicious actors can compromise the security of the network [8]. Inadequate security levels can lead to serious consequences for data confidentiality and integrity. In response to these challenges, research in this field is focused on developing methods for securing data exchange and creating reliable authentication schemes.

In [9], a secure key exchange method and an authentication scheme are presented, aimed at enhancing communication security in the IoT environment. The proposed approach combines the security functions of elliptic curve cryptography (ECC) with the Elliptic Curve Diffie-Hellman (ECDH) key exchange mechanism. Research shows that the protocol is characterized by low energy consumption and computational costs, making it efficient for IoT nodes operating under resource constraints.

In [10], an enhanced authentication system for IoT is proposed, which includes the use of technologies such as blockchain, artificial intelligence, and biometrics. This approach provides a dynamic and adaptive authentication process, improving the security and accuracy of user and device identification. Research shows that the presented solution enhances security levels through the integration of multi-factor authentication, certificate-based authentication, robust encryption, identity management, and a zero-trust model, contributing to an increase in resilience against threats, including DDoS attacks, by up to 80%.

In [11], a re-encryption proxy server scheme for IoT environments is proposed. This approach utilizes a key update mechanism at specified time intervals, supporting user identity revocation. The proposed scheme ensures high functionality and security, demonstrating resilience against attacks based on the DBDH assumption, and does not require high computational costs, making it effective for fog-based cloud environments. However, the scheme has limitations regarding resilience to quantum attacks and may impose a burden on low-power or low-capacity IoT devices due to the complexity of encryption and decryption.

In [12], a security model for IoT networks is presented, which includes blockchain and a post-quantum secure identification scheme (PQ-IDS). The proposed digital signature mechanism, based on lattice-based cryptography, ensures their non-repudiation. Research shows that the proposed approach provides security properties such as unforgeability, non-repudiation, and non-transferability. Comparisons of efficiency and performance assessments indicate that the proposed PQ-IDS exhibits high effectiveness and practicality in IoT network applications.

The conducted analysis confirms that research and development of authentication methods and secure communication in IoT device networks are crucial for ensuring the confidentiality, integrity, and availability of data. Developing solutions that not only meet efficiency and adaptability requirements but also provide robust protection against potential threats is an urgent task.

The aim of this work is to develop a hybrid authentication algorithm that combines the properties of encryption and error-correcting codes, aimed at enhancing the reliability and security of data transmission, as well as the effective management of cryptographic keys in conditions of limited hardware resources.

Research on IoT Device Authentication Algorithms. Authentication for IoT devices is a crucial step and an integral part of security in cyber-physical systems. It ensures the protection of data and devices, as well as guarantees the integrity and reliability of the overall system. The goal of authentication is to establish a unified security policy, minimize risks, and ensure secure interactions between devices, users, and servers within the IoT environment.

Among the common technologies and protocols that ensure security and efficiency in the IoT environment, MQTT, CoAP, DTLS, and PUF stand out. Their use facilitates secure communication between resource-constrained devices, enhancing the reliability and protection of IoT systems.

The MQTT (Message Queuing Telemetry Transport) messaging protocol is specifically designed for low-power devices and networks with limited resources [13]. It utilizes a "publish-subscribe" architecture (Fig. 1), allowing devices to exchange data without a constant connection. MQTT does not include built-in security mechanisms. For secure communication, external mechanisms such as TLS/SSL are used, which require additional computational overhead.

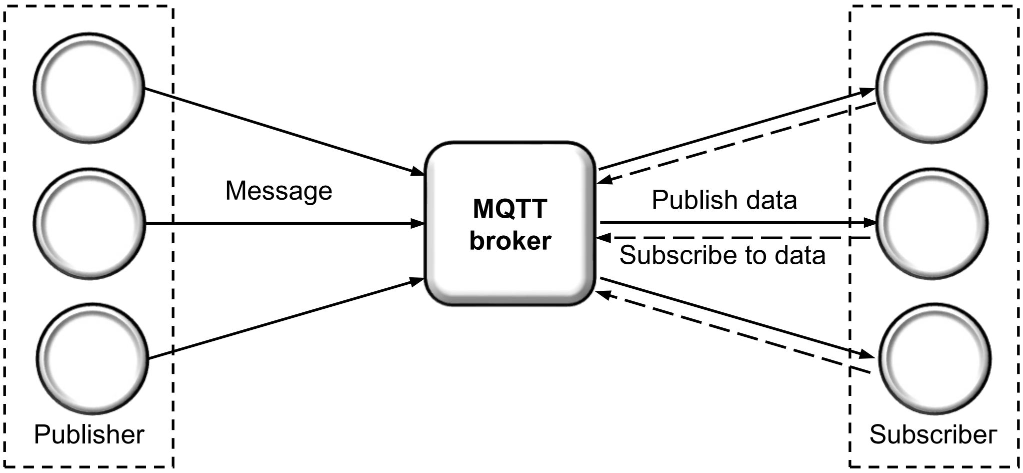


Fig. 1. MQTT Structure

Currently, the strategy of using a single broker is considered inefficient, making it advisable to implement a distributed strategy with multiple brokers in IoT networks, as this enhances the reliability and scalability of the system [14]. The widespread use of MQTT is attributed to its capabilities and features, including connectionless communication, high scalability, low energy consumption, and fast and reliable message delivery [15]. The protocol is applied in monitoring and management systems, as well as for effective routing in IoT [16].

The CoAP (Constrained Application Protocol) is used for interaction between IoT devices and servers. It provides unique addressing for each device, simplifying communication. CoAP is specifically designed for resource-constrained devices [17] and utilizes a simple request/response architecture (Fig. 2).

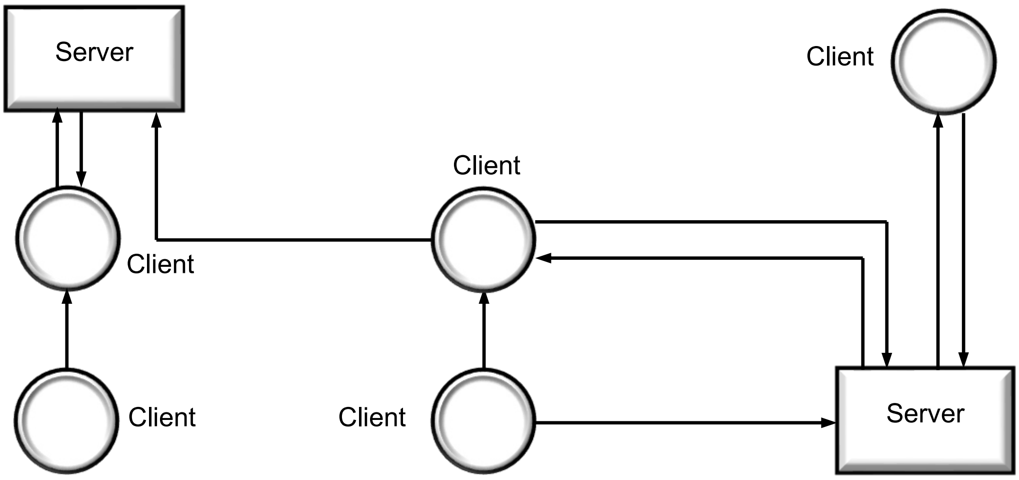


Fig. 2. CoAP Structure

CoAP is an adaptation of HTTP for use in IoT, as standard web communication protocols are considered "heavy" for IoT devices due to their memory, computational power, and energy consumption requirements. Despite its advantages, such as low energy and bandwidth requirements, CoAP is vulnerable to distributed denial-of-service (DDoS) attacks. To ensure security, such as authentication or encryption, the protocol is often integrated with DTLS (Datagram Transport Layer Security), which provides transport security for CoAP [18].

DTLS is a security protocol that provides encryption and authentication for data transmitted over unreliable networks (Fig. 3). It is an adaptation of TLS (Transport Layer Security) designed to work with data transmitted in datagrams [19].

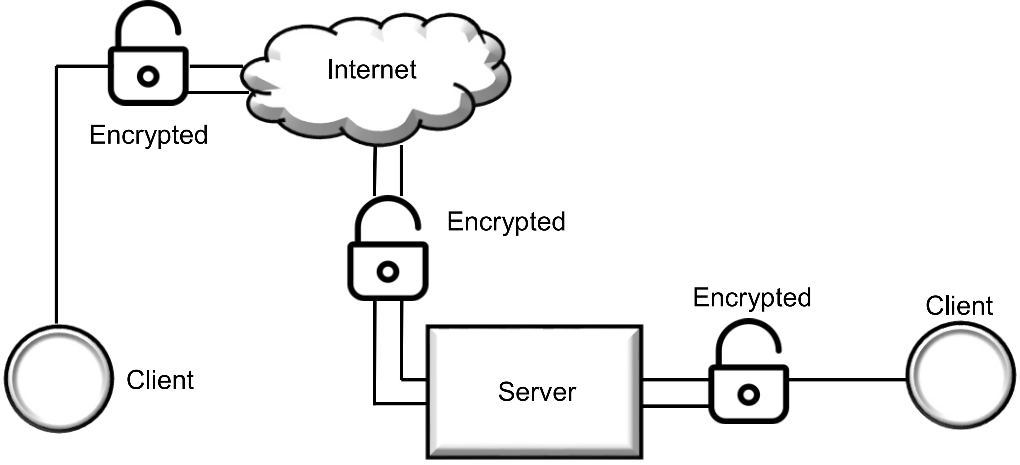


Fig. 3. DTLS Security Protocol

Although DTLS is optimized for constrained devices, it still requires significant computational resources to perform cryptographic operations, particularly the Diffie-Hellman key exchange and certificate verification, which ensure the authenticity of communications.

Authentication of IoT devices based on Physically Unclonable Functions (PUF) is a modern approach to ensuring security in IoT networks. A PUF is a hardware mechanism that utilizes the unique physical characteristics of each device to create a cryptographic identifier that cannot be copied or forged. Based on these unique characteristics, it can generate distinct responses to specific challenges (challenge-response pairs, CRP), which are used for device authentication [20].

The steps of authentication using PUF (Fig. 4) include an initial setup phase, during which each device generates a set of CRP that are stored in a secure repository. For authentication, the server sends a challenge to the IoT device, which uses its unique PUF to compute the response to the challenge. The response is sent back to the server, where it is compared with the stored data.

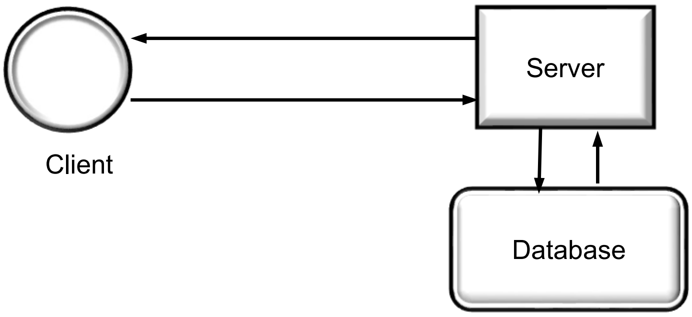


Fig. 4. PUF-Based Authentication Protocol in IoT

The advantages of this method include high resistance to spoofing and cloning, along with low computational resource requirements. However, it is sensitive to changes in the physical characteristics of the devices and vulnerable to challenge-response replay attacks [21].

The use of the discussed protocols enables secure communication between resource-constrained devices, enhancing the reliability and security of IoT systems. For IoT devices, traditional protocols may be excessive. The high computational requirements necessitate significant resources and energy, which limits their use for low-power IoT devices. Often, IoT devices lack reliable mechanisms for securely storing cryptographic keys, creating a threat of security compromise.

Despite the fact that protocols like MQTT, CoAP, and DTLS can provide device authentication in IoT, they do not address the issue of data integrity during transmission. These protocols focus on ensuring secure communication and authentication; however, they often leave the protection of data from modifications or loss during transmission unaddressed. This creates potential risks for IoT systems, as data integrity breaches can lead to malfunctioning devices or vulnerabilities within the network.

Operation of the Proposed Hybrid Authentication Algorithm. The proposed hybrid authentication algorithm RHAA (RSA and Hamming Code-based Authentication Algorithm), by combining encryption and coding, contributes to ensuring the confidentiality and integrity control of data in systems with a large number of nodes. This approach enhances the security and reliability of transmitted data and can be implemented to protect IoT device networks from various cyber threats.

By using the RSA algorithm at each encryption stage, the data remains protected from unauthorized access [22]. RSA allows for the use of small key sizes, which reduces memory requirements for storage and ensures authentication and secure data transmission between IoT devices. The construction of the Hamming code $GF(n)$ provides the ability to detect and correct errors that may occur during transmission, thereby reducing the risk of data loss or corruption [23]. The error-correcting properties of the code offer advantages for IoT device networks, as the quality of communication may be unstable. The Hamming code is computationally simple, using minimal memory resources. Its implementation in finite fields GF(p) does not require additional overhead. The RHAA is flexible and scalable, allowing for the collection, processing, and transmission of large volumes of data from various sources.

In Figure 5, the structure of the proposed RHAA algorithm is presented.

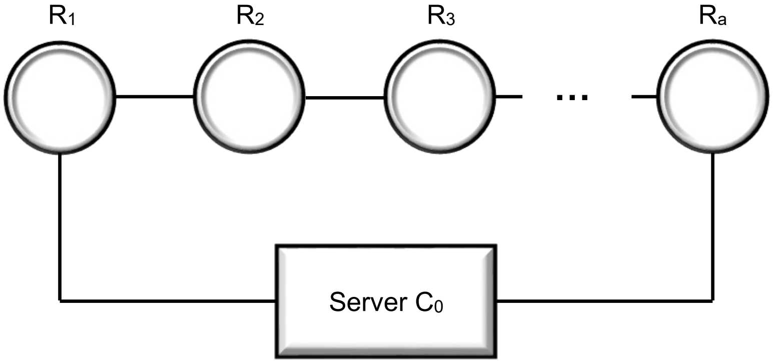


Fig. 5.– Structure of RHAA

The principle of operation of the proposed algorithm:

1. Setup Stage:

- Key Generation - for each node$R_1, R, \ldots, R_a$ a pair of RSA keys is generated: public keys (e_i, n) and private keys (d_i, n). The public keys are transmitted to each corresponding node, while the private keys are stored securely on the central server C.

2. Message Preparation:

- Verification Initiation - at defined time intervals, the central server C initiates a verification process by sending a control message containing specially formatted data that must sequentially pass through all nodes.

- The outgoing message m is encrypted on the central server C using the public key of R_0 with the RSA algorithm

$$c_0 = m^{e_0} \, mod \, n.$$

- The encrypted message c_0 is transformed into a codeword x_0 by constructing a Hamming code in finite fields $GF(n)$, which ensures the detection and correction of errors during transmission.

3. Transmission of the codeword through intermediate nodes $R_1, R_2, \dots, R_a$. Each node R_i receives the codeword x_i for which the following occurs:

- Data integrity check and, if necessary, correction.

- Decoding of the message by removing the parity symbols, resulting in the encrypted message m_i.

- Re-encryption of m_i using the public key of the next node R_{i+1}

$$c_{i+1} = m_{i+1}^{e_{i+1}} \, mod \, n.$$

- Converting c_{i+1} to codeword x_{i+1}, which is then passed to the next node R_{i+1}. The last node R_a transmits the code word x_a to the center C.

4. Decryption and data integrity verification:

- The central server C receives the codeword x_a, checks and corrects errors, removes the parity symbols, resulting in the encrypted message m_a.

- The central server C sequentially decrypts the message using the corresponding private keys for each node (d_i, n)

$$m_i = c_i^{d_i} \, mod \, n.$$

- If all decryption stages are successful, the central server C restores the original message m.

5. Intrusion Detection:

- If the original message cannot be restored during the verification process, it indicates a potential compromise or intrusion. This means that at least one of the intermediate nodes may have been breached, resulting in data loss or tampering, which jeopardizes the integrity and confidentiality of the transmitted information.

Since the data passes through several intermediate nodes, ensuring its confidentiality and integrity is crucial. The integration of the RSA algorithm provides data encryption and node authentication, preventing unauthorized access to the information. The construction of Hamming codes in $GF(n)$ ensures additional encryption and guaranteed detection and correction of a single error at all stages of data transmission. The central server C performs regular checks to detect potential security threats. In the event of an intrusion detection, it is recommended to change access keys immediately, which will help prevent further access by malicious actors and assist in restoring the integrity of the system.

The algorithm for the operation of RHAA is shown in Fig. 6.

To illustrate the operation of the proposed RHAA algorithm, let's consider an example.

Suppose the following values are chosen for key generation: $p = 71, q = 101$.

Consequently, we obtain the values $\phi(n) = 7000, n = p * q = 7171$.

The number of nodes is $R = 4$.

The control message is $m = 12, 35, 151, 569, 74, 84, 357$.

As a result of the key generation phase, we obtain the following key pairs:

R_0: $e_0 = (9,\ 7171)$; $d_0 = (3889,\ 7171)$.

R_1: $e_1 = (11,\ 7171)$; $d_1 = (5091,\ 7171)$.

R_2: $e_2 = (13,\ 7171)$; $d_2 = (1077,\ 7171)$.

R_3: $e_3 = (17,\ 7171)$; $d_3 = (5353,\ 7171)$.

R_4: $e_4 = (19,\ 7171)$; $d_4 = (2579,\ 7171)$.

After encrypting the output message using the public key $e_0 = (9,\ 7171)$, we can calculate the encrypted message c_0 using the RSA encryption formula:

$$c_0 = 2038, 6300, 4516, 5539, 1223, 1711, 2145.$$

The result of converting c_0 into a codeword in the field $GF(7171)$ is:

$$x_0 = 2903, 1607, 2038, 2013, 6300, 4516, 5539, 5079, 1223, 1711, 2145.$$

The transmission of the codeword through intermediate nodes R_1 to R_4 may be accompanied by the occurrence of errors. For example, the message received by node R_1 will look like this:

$$x'_0 = 2903, 16, 2038, 2013, 6300, 4516, 5539, 5079, 1223, 1711, 2145.$$

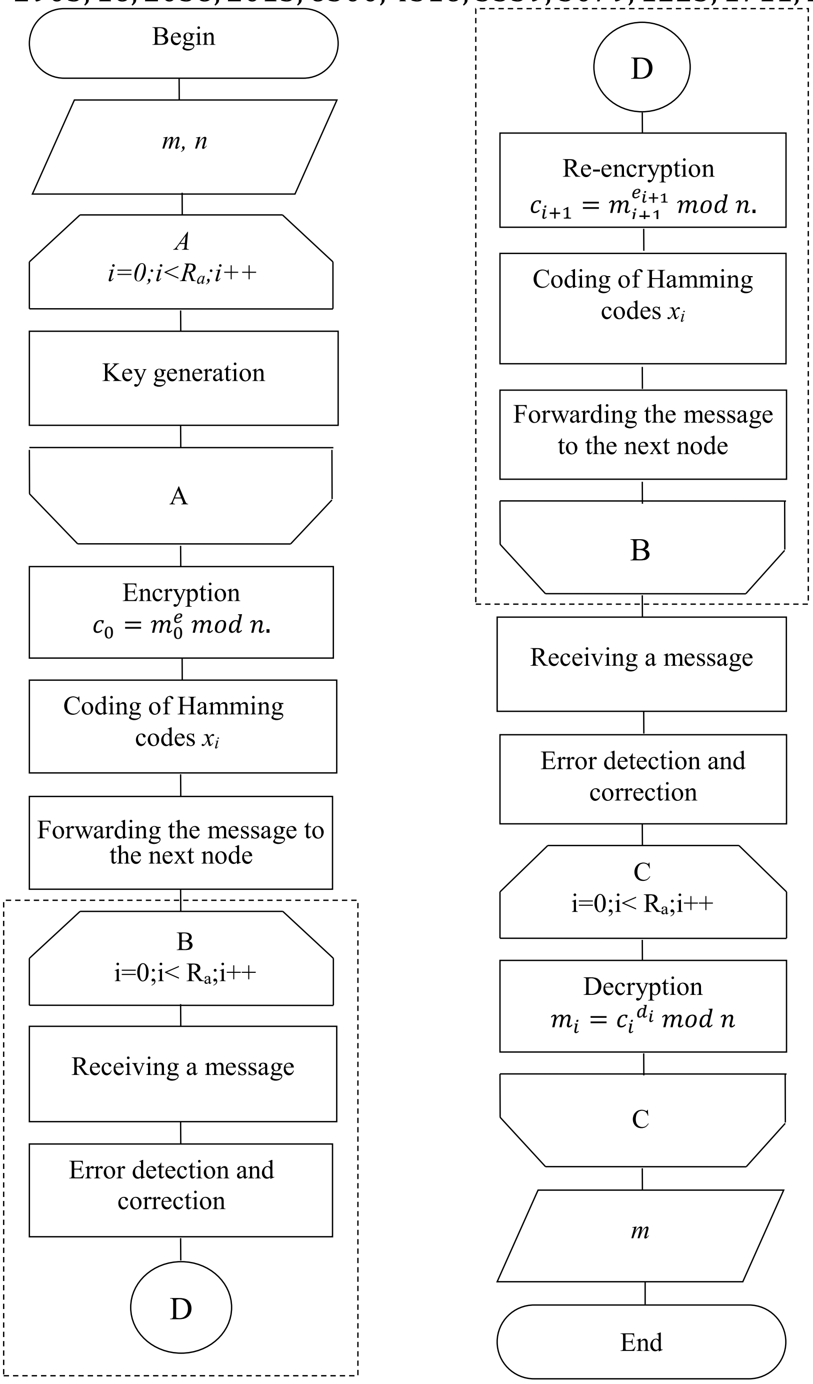
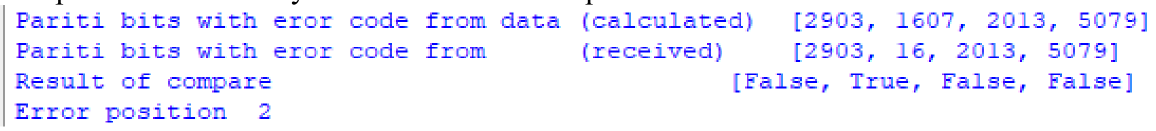

Begin
m, n
A
i=0;i<Ra;i++
Key generation
A
Encryption
$c_0 = m_0^e \ mod\ n.$
Coding of Hamming codes x_i
Forwarding the message to the next node
B
i=0;i< Ra;i++
Receiving a message
Error detection and correction
D
D
Re-encryption
$c_{i+1} = m_{i+1}^{e_{i+1}} \ mod\ n.$
Coding of Hamming codes x_i
Forwarding the message to the next node
B
Receiving a message
Error detection and correction
C
i=0;i< Ra;i++
Decryption
$m_i = c_i{}^{d_i} \ mod\ n$
C
m
End

Fig. 6. Algorithm of RHAA Operation

As a result of the integrity check (Fig. 7), an error was detected in position 2, which corresponds to a check symbol that does not require correction.



```
Pariti bits with eror code from data (calculated)  [2903, 1607, 2013, 5079]
Pariti bits with eror code from      (received)    [2903, 16, 2013, 5079]
Result of compare                                [False, True, False, False]
Error position  2
```

Fig. 7. Error Detection Process

The result of further decoding and encryption using the public key $e_1 = (11,\ 7171)$ will be the encrypted message:

$$c_1 = 2281, 6894, 6260, 4474, 86, 3226, 4199.$$

As a result of constructing the Hamming code in the field $GF(n)$ from the message c_1, we have:

$$x_1 = 3592, 6098, 2281, 3286, 6894, 6260, 4474, 340, 86, 3226, 4199.$$

The message received by node R_2, i considering potential distortion during transmission, will appear as follows:

$$x_1' = 3592, 6098, 2281, 3286, 105894, 6260, 4474, 340, 86, 3226, 4199.$$

As a result of the data integrity check, an error was detected in position 5. After correction, we obtain:

$$x_1 = 3592, 6098, 2281, 3286, 6894, 6260, 4474, 340, 86, 3226, 4199.$$

After decoding and re-encrypting using the public key $e_2 = (13, 71791)$, we obtain:

$$c_2 = 5010, 1022, 5241, 6036, 3528, 1905, 322.$$

In the message received by node R_3, no modifications were detected during the integrity check, so no correction is needed. Therefore, we will use the received message for further processing:

$$x_2 = 1576, 4172, 5010, 5128, 1022, 5241, 6036, 5755, 3528, 1905, 322.$$

As a result of decoding and decrypting using the public key $e_3 = (17,\ 7171)$, we obtain

$$c_3 = 714, 3865, 5567, 5752, 1927, 5448, 371.$$

The codeword for transmission to node R_4 will be as follows:

$$x_3 = 5458, 3510, 714, 842, 3865, 5567, 5752, 575, 1927, 5448, 371.$$

During transmission, the data were corrupted; therefore, the codeword received by node R_4 appears as follows:

$$x_3' = 5458, 3510, 714, 842, 3865, 5567, 1, 575, 1927, 5448, 371.$$

An error is detected in position 7, corresponding to the information symbol, which requires correction. This correction process is carried out similarly to the previous stages.

As a result of encrypting with the key $e_4 = (19,\ 7171)$, we obtain::

$$c_4 = 3345, 3469, 858, 569, 1084, 4528, 2478.$$

The central server C receives the codeword:

$$x_4' = 3774, 4607, 3345, 4896, 3469, 858, 9503, 919, 1084, 4528, 2478.$$

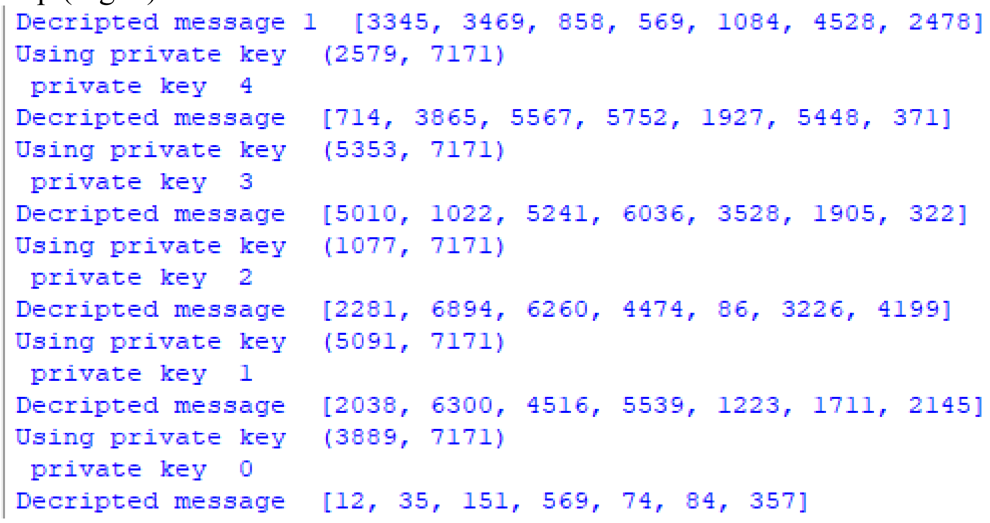
The integrity check detected an error in position 7, and after correction, we obtain:

$$x_4 = 3774, 4607, 3345, 4896, 3469, 858, 569, 919, 1084, 4528, 2478.$$

The check bits are no longer needed, and by discarding them, we obtain the message for decryption:

$$c_4 = 3345, 3469, 858, 569, 1084, 4528, 2478.$$

In order to detect an intrusion, data is decrypted step by step using private keys $d_0 - d_4$ at each step (Fig. 8).



```
Decripted message 1  [3345, 3469, 858, 569, 1084, 4528, 2478]
Using private key  (2579, 7171)
 private key  4
Decripted message  [714, 3865, 5567, 5752, 1927, 5448, 371]
Using private key  (5353, 7171)
 private key  3
Decripted message  [5010, 1022, 5241, 6036, 3528, 1905, 322]
Using private key  (1077, 7171)
 private key  2
Decripted message  [2281, 6894, 6260, 4474, 86, 3226, 4199]
Using private key  (5091, 7171)
 private key  1
Decripted message  [2038, 6300, 4516, 5539, 1223, 1711, 2145]
Using private key  (3889, 7171)
 private key  0
Decripted message  [12, 35, 151, 569, 74, 84, 357]
```

Fig. 8. Decryption process

The proposed RHAA algorithm, as a component of a comprehensive information protection system, ensures the confidentiality and integrity of data, providing a high level of protection and accuracy of information, as well as detecting anomalies, which helps to detect potential attacks. Using the proposed approach provides protection against the following types of attacks:

- Data Integrity Attacks - coding protects against data changes that occur when attempts are made to replace or modify data, for example, when traffic parameters are changed or part of the information is destroyed.
- Man-In-The-Middle (MITM) Attacks - encrypted data using the RSA algorithm is difficult to forge. Even if an attacker intercepts the message, altering the ciphertext will lead to errors that reveal the characteristics of the correction codes.;
- Data Interception - thanks to multiple encryption layers, an attacker will not be able to decrypt the intercepted data without the appropriate keys, which reduces the risk of losing confidential information.
- Unauthorized Access - in the case of node compromise, the system may detect a discrepancy in the keys, which signals a breach.
- Replay Attacks - regular checking ensures the detection of attempts to reuse encrypted messages.
- Confidentiality Attacks - encryption and coding ensure a high level of confidentiality, protecting data from unauthorized access during transmission.

The proposed RHAA is easy to adapt or expand according to growing requirements or changes in the environment. The proposed approach is more adaptive, compared to traditional authentication methods that focus only on identifying and confirming devices or users. RHAA actively monitors the quality of data transmission and provides solutions to eliminate problems. The combination of error detection and correction mechanisms with re-encryption improves efficiency in ensuring data integrity and security in cyber-physical systems with a large number of nodes, where the risk of errors is significantly increased.

Conclusions. The proposed RHAA algorithm provides detection of device compromise thanks to key loss monitoring and node authentication functions. This allows timely response to potential threats and unauthorized access attempts. The use of correction codes ensures the detection of errors of any size in the symbols of the code word and performs the correction of single errors in the data block. Re-encryption and coding make it possible to increase the security level by R times, which makes RHAA more reliable and adaptable to modern data transmission conditions.

The combination of RSA and Hamming code construction in finite fields provides an effective authentication scheme for IoT devices, with the correct choice of encryption parameters, for example, using keys of a given length, to minimize IoT resources.

References

1. Shopina I. Information security of digital transformation. *Scientific Bulletin of the Lviv State University of Internal Affairs. Legal series*. 2023. No. 1. P.28-35. doi: https://doi.org/10.32782/2311-8040/2023-1-4.
2. Pöhn D., Hommel W. Towards an Improved Taxonomy of Attacks related to Digital Identities and Identity Management Systems. *Security and Communication Networks.* 2024. doi: 10.48550/arXiv.2407.16718.
3. Mumin A., Hammoudeh M., Alrawashdeh R., Alsulaimy B. A Survey on Security, Privacy, Trust, and Architectural Challenges in IoT Systems. *IEEE Access*. 2024. PP.(99): 1-1. doi: 10.1109/ACCESS.2024.3382709.
4. Srivastava N., Pandey P. Internet of things (IoT): Applications, trends, issues and challenges. *Materials Today: Proceedings.* 2022. V. 69/2. P. 587-591. doi: 10.1016/j.matpr.2022.09.490
5. Ibibo J.T. IoT Attacks Countermeasures: Systematic Review and Future Research Direction. *BDTA 2023, LNICST 555.* 2023. P. 95–111. doi: 10.1007/978-3-031-52265-9_7.

6. Arora R., Muqeem M., Saxena M.. Developing a Comprehensive Security Framework for Detecting and Mitigating IoT device Attack. 2024. doi: 10.21203/rs.3.rs-5165811/v1.
7. Yilmaz S., Dener M. Security with Wireless Sensor Networks in Smart Grids: A Review. *Symmetry*. 2024. 16(10): 1295. doi: 10.3390/sym16101295
8. Kalaria R., Kayes A.S.M., Rahayu W., Pardede E., Salehi S. A. IoTPredictor: A security framework for predicting IoT device behaviours and detecting malicious devices against cyber attacks. *Computers & Security*. 2024. V. 146. doi: 10.1016/j.cose.2024.104037.
9. Peivandizadeh A., Haitham Y. A., Molavi B., Mohajerzadeh A., Al-Badi H. A. A Secure Key Exchange and Authentication Scheme for Securing Communications in the Internet of Things Environment. *Future Internet* 2024. V.16. No. 16(10). P. 357. doi: 10.3390/fi16100357
10. Maiwada U.D., Danyaro K.U., Janisar A.A., Abdullahi M. Enhancing Security of 5G-Enabled IoT Systems through Advanced Authentication Mechanisms: A Multifaceted Approach. *UMYU Scientifica*. 2024. V. 2. No. 4. P. 201-2011. doi: 10.56919/usci.2324.025.
11. Lin H.-Y., Chen P.-R. Revocable and Fog-Enabled Proxy Re-Encryption Scheme for IoT Environments. *Sensors*. 2024. V.24. No. 19: 6290. doi: 10.3390/s24196290
12. Zhang Y., Tang Y., Li C., Zhang H., Ahmad H. Post-Quantum Secure Identity-Based Signature Scheme with Lattice Assumption for Internet of Things Networks. *Sensors* 2024. V.24, No. 13: 4188. doi: 10.3390/s24134188
13. Abdulelah H., Mohammad I. Effective Feature Engineering Framework for Securing MQTT Protocol in IoT Environments. *Sensors*. 24. 1782. 2024. doi: 10.3390/s24061782.
14. Hmissi F., Ouni Se. TD-MQTT: Transparent Distributed MQTT Brokers for Horizontal IoT Applications. *2022 IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecommunicatio*ns (SETIT), Hammamet, Tunisia. 2022. P. 479-486. doi: 10.1109/SETIT54465.2022.9875881.
15. Lakshminarayana S., Praseed A., Thilagam P. Securing the IoT Application Layer from an MQTT Protocol Perspective: Challenges and Research Prospects. *IEEE Communications Surveys & Tutorials*. 2024. P. 1-1. doi: 10.1109/COMST.2024.3372630.
16. Sonam, Johari R., Garg S., Bawa P., Aggarwal D. MIAWM: MQTT based IoT Application for Weather Monitoring. *Journal of High Speed Networks*. 2024. V.30. P.1-22. doi: 10.3233/JHS-230008.
17. Tariq M.A., Khan M., Khan M.T.R., Kim D. Enhancements and Challenges in CoAP-A Survey. *Sensors*. 2020. V.20. P.6391. doi: 10.3390/s20216391.
18. Westphall J., Loffi L., Merkle Westphall C., Martina J. CoAP + DTLS: A Comprehensive Overview of Cryptographic Performance on an IOT Scenario. *IEEE Sensors Applications Symposium (SAS)*. 2020. P.1-6. doi: 10.1109/SAS48726.2020.9220033.
19. Restuccia G., Tschofenig H., Baccelli E. (2020). Low-Power IoT Communication Security: On the Performance of DTLS and TLS 2020. 1.3. doi: 10.48550/arXiv.2011.12035.
20. Tun W.N., Mambo M. Secure PUF-Based Authentication Systems. *Sensors*. 2024. 24(16): 5295. doi: 10.3390/s24165295.
21. Rajput S., Dofe J. Secure Dynamic PUF for IoT Security. Internet of Things. *Advances in Information and Communication Technology*. 2023. P.454-462. doi: 10.1007/978-3-031-45878-1_33.
22. Rivest R.L. Shamir A., Adleman L.M.A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*. 1978. V. 21. No. 2. P. 120-126. doi: 10.7551/mitpress/12274.003.0047
23. Davletova A. Побудова кодів Хеммінга в скінченних полях Галуа. *Herald of Khmelnytskyi National University. Technical Sciences. 2024.* V.333(2). P.28-34. doi: 10.31891/2307-5732-2024-333-2-4.

МЕТОД АВТЕНТИФІКАЦІЇ ПРИСТРОЇВ В МЕРЕЖАХ ІНТЕРНЕТ РЕЧЕЙ

А.Я. Давлетова

Західноукраїнський національний університет
11, Львівська, м. Тернопіль, 46009, Україна
Email: a7davletova@gmail.com

Робота присвячена вирішенню актуальної задачі забезпечення безпечної та надійної передачі даних та ефективного управління криптографічними ключами в мережах IoT. Типові протоколи автентифікації, хоча й забезпечують захищену комунікацію, можуть бути занадто складними для багатьох пристроїв з обмеженими ресурсами. Це підкреслює необхідність дослідження та пошукуефективних рішень, які відповідають ресурсним можливостям пристроїв IoT і дозволять забезпечити надійне шифрування та автентифікацію. Представлено гібридний алгоритм автентифікації RHAA (RSA and Hamming Code-based Authentication Algorithm), розроблений для забезпечення конфіденційності, цілісності та автентифікації даних у мережах IoT з великою кількістю вузлів. Особливістю запропонованого алгоритму є поєднання асиметричного шифрування на основі RSA з використанням корегуючих властивостей коду Хеммінга в скінченних полях. Такий підхід гарантує автентифікацію пристроїв, які беруть участь у процесі обміну інформацією, за рахунок централізованої генерації та управління ключами. Захист даних від несанкціонованого доступу досягається повторним шифруванням інформації на кожному з вузлів. Цілісність даних забезпечується шляхом виявлення та виправлення помилок під час кожного з етапів передачі. Таке рішення підвищує рівень безпеки даних в мережах IoT , знижуючи ризики витоку чи втрати інформації. У роботі наведено приклад реалізації алгоритму RHAA із застосуванням реальних значень ключів та показано, як система реагує на появу помилок і виконує їх корекцію. Запропонований алгоритм, оптимізований для обмежених пристроїв, може бути використаний для покращення захисту даних у мережах IoT.

Ключові слова: автентифікація, мережі IoT, алгоритм RSA, код Хеммінга в скінчених полях, шифрування, цілісність даних, виявлення та корекція помилок, конфіденційність, безпека передачі даних.