In-depth Study of Intelligent Healthcare Systems Using 5G Security

Bindu Ananthula[1], Niharika Budde[1], Oleksii Baranovskyi[1], Tetiana Babenko[2,3], Andrii Bigdan[3] and Rostyslav Lisnevskyi[2,3]

[1] *Blekinge Tekniska Högskola, 371 79 Karlskrona, Sweden*
[2] *International Information Technology University, 34/1 Manas St., Almaty, Kazakhstan*
[3] *Taras Shevchenko National University of Kyiv, 64/13 Volodymyrska Street, Kyiv, 01601, Ukraine*

Abstract

A promising approach to raising the caliber and accessibility of healthcare services is the development of Smart Healthcare Systems. However, the union of wireless networks and smart medical devices has created additional security issues, such as the possibility of identity theft, data breaches, and denial-of-service assaults. These flaws emphasize the significance of creating a safe and dependable smart healthcare system that can safeguard patient data and guarantee the confidentiality of private medical information.

This study suggests adopting 5G security standards to address the security issues with smart healthcare systems. The threat modeling approach, which includes six threat categories (spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege), is used in this study to investigate potential threats in smart healthcare systems. The report suggests using strong encryption protocols between smart healthcare equipment and 5G-AKA to reduce these potential threats.

The proposed approach showed appreciable advancements in data security and privacy. According to the findings, 5G security standards can be used to efficiently reduce security risks in smart healthcare systems and establish a trustworthy and secure platform for delivering medical services. The study emphasizes the significance of including strong security controls in such systems to secure patient information and raise the standard of treatment generally.

Keywords

Smart Healthcare, 5G, IoT, security, risk management

1. Introduction

A newly formed prototype is produced through technological advancement and adopted daily. Smart gadgets use Secure Internet protocols to transfer sophisticated data. These gadgets are used by many industries, including those in healthcare, business, administration, and education. The importance of Smart Healthcare Systems (SHS) is highlighted by the fact that some of them give users more control over their health data [1].

SHS uses networks and smart devices (such as smartphones, smartwatches, wireless smart glucometers, and wireless blood pressure monitors) to deliver healthcare services. Smart devices examine health information gathered from many sources, such as sensors and biological systems. In short, smart healthcare makes it possible for people from different backgrounds and professions (such as doctors, nurses, patient caregivers, family members, and patients) to access the right information and find the right solutions, which are primarily to reduce medical errors, increase efficiency, and cut costs when necessary. Improved patient care, lower healthcare costs, and greater provider efficiency are just a few advantages that SHS can give. These systems can aid medical professionals in optimizing their procedures, as well as lowering mistakes and improving patient outcomes. Furthermore, by giving patients real-time data and individualized treatment plans, smart SHS can enable individuals to play a more active role in their healthcare.

DTESI 2023: Proceedings of the 8th International Conference on Digital Technologies in Education, Science and Industry, December 06–07, 2023, Almaty, Kazakhstan

bian21@student.bth.se (B. Ananthula); nibu21@student.bth.se (N. Budde); oleksii.baranovskyi@bth.se (O. Baranovskyi); babenko.tetiana.v@gmail.com (T. Babenko); r.lisnevskyi@iitu.edu.kz (R. Lisnevskyi)

0009-0002-4657-3034 (B. Ananthula); 0009-0004-5450-9926 (N. Budde); 0000-0001-5629-5205 (O. Baranovskyi); 0000-0002-2940-6085 (T. Babenko); 0000-0003-1184-9483 (A. Bigdan); 0000-0002-9006-6366 (R. Lisnevskyi)

Overall, SHS have the potential to change the healthcare sector by improving access to, affordability of, and effectiveness of healthcare for everybody.

The most recent standard for cellular networks is 5G or fifth-generation wireless technology. It offers increased device connectivity, higher data rates, and reduced latency. A security mechanism known as 5G-AKA (5G Authentication and Key Agreement) is used in 5G networks to authenticate devices and provide secure communication channels. With improvements to satisfy the more stringent security needs of 5G networks, it is an evolution of the AKA protocol used in earlier cellular networks. Network slicing capability, which enables the creation of many virtual networks inside a single physical network infrastructure, is one of the main aspects of 5G-AKA. The 5G-AKA protocol imposes its own set of access control rules and security parameters on each network slice. Additionally, 5G-AKA enables improved privacy features such the temporary identifiers that shield a device's identity and stop tracking. The protocol also has safeguards against attacks like eavesdropping, replay assaults, and man-in-the-middle attacks [2].

2. Literature review

In [3] authors suggest a 5G-based design for the healthcare dedicated network and other smart healthcare information infrastructure. The design uses the iGW as its central component and optimizes the most recent technical architecture regarding MEC 5G Integration that has been specified by 3GPP and ETSI for use in a smart healthcare scenario. Additionally, the network architecture supports intra-hospital, inter-hospital, and extra-hospital application scenarios for smart healthcare. Different apps can be launched and maintained in the 5G healthcare cloud by utilizing the new 5G-based medical information infrastructure that is made up of the dedicated network and cloud. where the 5G basic application clearly shows how 5G improves hospital production efficiency and 5G transmission capabilities. The goal of the medical applications is to create a full-scenario smart medical service ecosystem for telemedicine, emergency rescue, smart hospitals, and other applications in a variety of medical situations, including in-hospital, inter-hospital, and out-of-hospital.

In this [4] work, to accomplish the following objectives: rapid and accurate context-aware health situation identification; a secure data-sharing mechanism based on blockchain; and responsive and low-latency services for urgent patients, authors have proposed a framework for 5G-secure-smart healthcare monitoring.

The use of Advanced Encryption Standard (AES) in healthcare systems to ensure the confidentiality of patient data has been explored in [5] and proposed a secure data-sharing scheme based on AES for cloud-based electronic health records. Their scheme allowed patients to share their records with healthcare providers while ensuring that the data remained confidential. Similarly, [6] proposed an AES-based data encryption scheme for mobile health systems. Their scheme ensured that patient data remained confidential during transmission between the mobile device and the healthcare provider.

The STRIDE (acronym of Spoofing, Tampered, Reputation, Information disclosure, Denial of service, Elevation of privilege) model has been proposed as a way to ensure the confidentiality, integrity, and availability of data in healthcare systems. In the context of smart healthcare systems, the STRIDE model has been used to ensure that patient data is transmitted securely between the mobile device and the healthcare provider.

A comparison of the encryption protocols used in SHS [7] reveals that DHE (Diffie–Hellman key exchange) is vulnerable to man-in-the-middle attacks, which can compromise the confidentiality and integrity of the communication. In addition, DHE can be computationally expensive, particularly for resource-constrained devices used in smart healthcare systems. These factors make DHE less suitable for use in SHS that require high security and efficient communication. These challenges include the vulnerability to man-in-the-middle attacks, the computational overhead, and the complexity of managing and distributing keys. In contrast, research on implementing AES with ECDH (Elliptic Curve Diffie-Hellman) in healthcare systems has shown that it is a more efficient and secure solution for key exchange. Case studies of implementing AES with ECDH in healthcare systems have demonstrated its effectiveness in securing data transmission. However, challenges still exist, such as key management and distribution, and there is a need for further research to address these challenges [8].

3. Problem statement

The purpose of the study is to explore and assess the potential security risks in SHS using Threat Modelling, and to suggest a new encryption strategy to improve the security of SHS employing the capabilities of 5G technology. By investigating the security risks already present in SHS, suggesting the addition of a new encryption method using AES-CCM with ECDH using 5G.

- Explanation of the implementation of 5G in the context of a smart healthcare system.
- Assess the risk for integrity, confidentiality, and privacy.
- Attenuation plans for the problems identified by threat modeling.
- Propose improvement to the current security measures by introducing additional encryption between SHS and 5G.

4. Methods and technologies

4.1. Smart Healthcare Systems

This architecture of SHS gives the main knowledge about the efficient, effective, and secure delivery of healthcare services made possible through SHS. A decision can be made by the SHS based on the patient's observed conditions and body temperature, heart rate, and pulse. Since it does not constantly switch on all the sensors, its architecture is also an energy-efficient solution. The system's algorithm will manage the utilization of the sensors and regulate their price and lifespan [9]. The study's suggested smart healthcare monitoring and patient management system includes communication channels, embedded internal and exterior sensors, an IoT server, cloud storage, and gateway support. Body temperature, pulse rate, and heartbeat sensors make up the system's three sensors. The Arduino board connects these three sensors to gather and classify medical data. Devices for networking and communication control data transport. The fuzzy logic system is employed in this arrangement to enable decision-making, with data analytics providing decision-making capabilities. The doctor's view gives hospital professionals the ability to observe and speak with the patient at a distance [10].

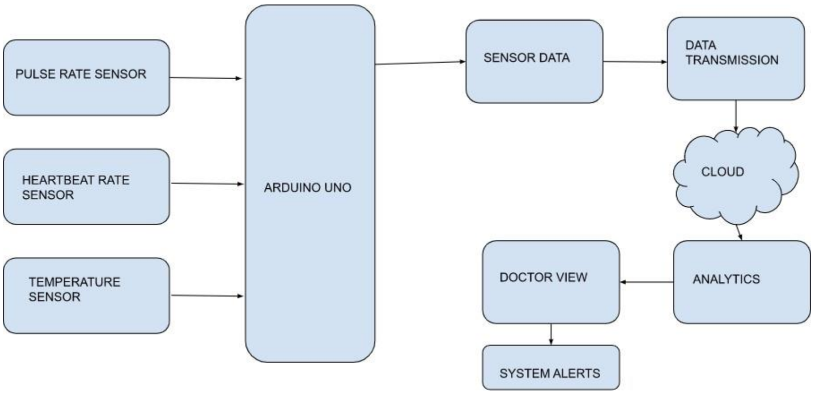

Figure 1: Architecture of Smart Healthcare Systems

Numerous advantages, including better patient outcomes and more effective healthcare delivery, are possible with SHS. They also present special privacy and security issues, which demand attention if patient data is to be kept secure [11-13]. To handle security and privacy in SHS, consider the following: secure data storage, access controls, education and training for users, regulation compliance, regular security audits and testing.

Threats to SHS can jeopardize both their performance and security. which may endanger the lives of patients and cost healthcare providers money in downtime. The widespread use of SHS might also raise challenges with data privacy and moral considerations with the gathering, storing, and utilization of personal health information: user impersonation attack, device impersonation attack, forward secrecy attack, man-in-the-middle attack, untraceable attack, replay attack, password change attack, eavesdropping.

4.1.1 Threat Analysis

There are so many methodologies that have been used while threat modeling (STRIDE, DRIDE, PASTA, VAST, CVSS, Attack Trees, etc.). The right threat model for your needs depends on what types of threats you're trying to model and for your purpose. [14] Upon considering the model of SHS architecture, STRIDE threat modeling is taken into view, which explains the workflow to analyze the threats and to find the mitigations for the threats.

Table 1
STRIDE classification

Threat Classification	Cause	Desired Property	Affected Attribute of SHS
Spoofing	By looking at the authentication processes in place, such as username/password systems, to make sure that only authorized users are given access, spoofing threats can be found in SHS	Authenticity	Connectivity, Data privacy and security, Integration with Healthcare systems, Health data privacy and security, Remote Healthcare support and User-friendly Interface.
Tampering	By checking the accuracy of the information and software used to manage patient data and healthcare services, tampering hazards can be found in SHS.	Integrity	Connectivity, Data accuracy and Reliability, Data security and privacy, Integration with the healthcare system, Health data privacy and security, Health Insights and Analysis, Remote Healthcare support and User-friendly Interfaces.
Repudiation	By adding tools to detect and document user activities, like as audit trails and digital signatures, to make sure that actions taken by users within the system can be traced back to them, repudiation threats can be identified in SHS.	Non-Repudiation	Data Accuracy and Reliability
Information Disclosure	By regularly doing vulnerability assessments and penetration tests to find any potential security holes that an attacker could exploit, Information Disclosure hazards can be found in SHS.	Confidentiality	Connectivity, Data privacy security, Integration with the Healthcare Systems, Health data privacy security, Health Insights and Analysis, Remote Healthcare and support, Specific medical applications.
Denial of Service	By monitoring network traffic and system performance for any unusual behavior or spikes in traffic that could signify a DoS attack, Denial of Service threats can be found in SHS.	Availability	Connectivity and Remote Healthcare support.
Elevation of Privilege	Adopting stringent access controls and role-based permissions, SHS can detect threats involving the elevation of privilege by making sure that users only have access to the privileges required to carry out their job duties.	Authorization	Connectivity, Data privacy and security, Integration with Healthcare systems, Remote Healthcare support, User-friendly Interfaces

4.1.2 Limitations of Smart Healthcare Devices

The battery life and processing power of smart medical devices, such as wearable health monitors, are similarly constrained. Some of these restrictions consist of limited battery life, computation power, technical difficulties, cost, security and privacy.

In summary, additional encryption is a critical component of securing medical data transmitted through smart healthcare devices and 5G networks. It protects sensitive data from unauthorized access, ensures confidentiality, integrity, and availability of the data, maintains patient privacy, and helps prevent cyberattacks and data breaches.

4.2. 5G-AKA Architecture

For the 5G core network, a service-based architecture (SBA) has been suggested. In light of this, 5G also defines new entities and service requests [15].

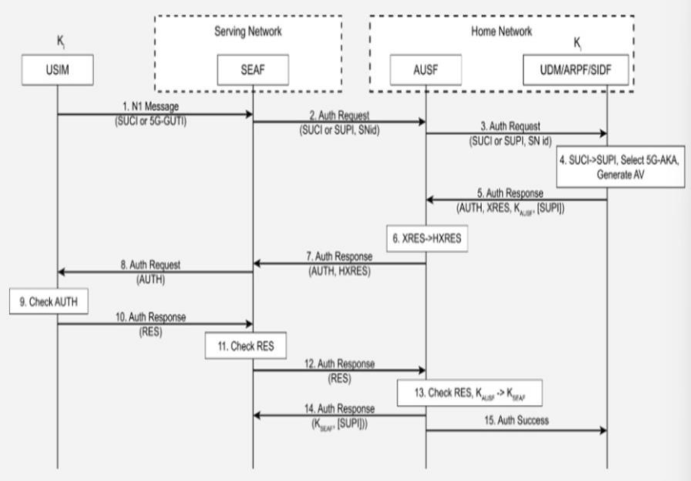
Serving Network
Home Network
K_i
K_i
USIM
SEAF
AUSF
UDM/ARPF/SIDF
1. N1 Message (SUCI or 5G-GUTI)
2. Auth Request (SUCI or SUPI, SNid)
3. Auth Request (SUCI or SUPI, SN id)
4. SUCI->SUPI, Select 5G-AKA, Generate AV
5. Auth Response (AUTH, XRES, K_{AUSF}, [SUPI])
6. XRES->HXRES
7. Auth Response (AUTH, HXRES)
8. Auth Request (AUTH)
9. Check AUTH
10. Auth Response (RES)
11. Check RES
12. Auth Response (RES)
13. Check RES, K_{AUSF} -> K_{SEAF}
14. Auth Response (K_{SEAF}, [SUPI])
15. Auth Success

Figure 2: 5G-AKA Architecture

The 5G-AKA (Authentication and Key Agreement) protocol is used in 5G mobile communication networks for authentication and key agreement between a mobile device and the network [16,17]. The protocol involves several steps and can be explained using the following schema:

- Initialization: the mobile device and the network exchange their identities and initial parameters to begin the authentication process. In 5G, this step involves the exchange of a "SUPI" (Subscription Permanent Identifier) and "Ks-NN" (a network-specific key) between the mobile device and the network.
- Random Challenge: the network sends a random challenge, called "RAND", to the mobile device, which the device must use to generate a response. This challenge is used to prevent replay attacks.
- Response: the mobile device generates a response to the challenge using its secret key, called "Ks", and sends it back to the network. The response also includes the "SUPI", a temporary identifier called "SUCI" (Subscription Concealed Identifier), and a "MAC" (Message Authentication Code) to ensure the integrity of the response.
- Authentication: the network authenticates the mobile device based on the response it received. If the authentication is successful, the network sends the mobile device a shared secret key, called "Kseaf" (Ks for the Encryption Algorithm with Freshness).
- Key Agreement: the mobile device and the network use the shared secret key "Kseaf" to generate a session key for secure communication. The session key is used to encrypt and decrypt all subsequent messages between the mobile device and the network.
- Security Mode: once the session key is established, the mobile device and the network enter into a security mode where they use the session key to encrypt and decrypt all subsequent messages.

The 5G-AKA protocol is designed to be more secure than previous versions of the AKA protocol used in older mobile communication networks. It includes additional security features, such as the use of "SUCI" and "MAC", to ensure the privacy and integrity of the authentication process. Additionally, the use of "Kseaf" allows for more secure key agreement between the mobile device and the network.

4.2.1 Limitations of usage for 5G-AKA in SHS

The AKA protocol can be used in a healthcare system to provide secure authentication and key agreement between a patient's mobile device and the healthcare network [18,19]. This can be useful in several scenarios, such as when a patient needs to securely access their medical records or when transmitting sensitive medical data.

The 5G-AKA protocol is a security protocol used in 5G networks to provide mutual authentication and establish a secure communication channel between the user equipment (UE) and the core network [20]. However, like any other security protocol, the 5G-AKA protocol has some limitations, including:

- Lack of forward secrecy: the 5G-AKA protocol does not provide forward secrecy, which means that if an attacker gains access to the long-term keys used for authentication, they can decrypt past and future communication.
- Vulnerability to man-in-the-middle (MITM) attacks: the 5G-AKA protocol is susceptible to Man-in-the-Middle (MITM) attacks, wherein an attacker intercepts the communication between the User Equipment (UE) and the network. By modifying the messages, the attacker can gain unauthorized access to the system.
- Potential for denial-of-service (DoS) attacks: the protocol is susceptible to DoS attacks, where an attacker can flood the network with fake authentication requests, causing a denial of service to legitimate users.
- Complexity: the 5G-AKA protocol is complex and requires a significant amount of computational resources, which can increase the processing time and delay the authentication process.

Overall, the 5G-AKA protocol is a robust security protocol, but it has some limitations that need to be addressed to ensure the security and privacy of the 5G networks.

4.3. Additional encryption

4.3.1 Lightweight Encryption

The term "lightweight encryption" refers to cryptographic algorithms and protocols that are specifically created to function effectively on devices with limited resources, such as memory, processing power, and energy. Low-power embedded systems, Internet of Things (IoT) gadgets, and wireless sensor networks are a few examples of these gadgets. Security, performance, and resource utilization are frequently traded off during the design of lightweight encryption methods. To protect sensitive data, they must nevertheless offer a high enough level of protection [21].

Using Lightweight Encryption in SHS offers several benefits: low computational cost, low memory usage, fast encryption and decryption, small code size, standardization.

To assess the trade-offs between security, performance, and resource utilization of the two encryption algorithms, AES-CCM and SPECK were compared. Using the unique needs and limitations of the system, this comparison aids in choosing the best encryption method for a certain application [22,23]. In conclusion, AES-CCM is a better choice for lightweight encryption in smart healthcare SHS, including the healthcare industry. While SPECK may have advantages in terms of performance and smaller code size, the security and reliability of AES-CCM make it a more reliable choice for securing medical data.

4.3.2 Key exchange protocol

Understanding the security and performance aspects of the two key exchange protocols requires a comparison of ECDH and DHE. Based on elements like security requirements, computational capabilities, and network conditions, this comparison aids in choosing the most suitable key exchange protocol for a given system [4,26].

ECDH is a more suitable choice for key exchange in SHS due to its ability to provide strong security with shorter key lengths, which is essential for resource-constrained devices used in smart healthcare systems. While DHE may provide greater security, its higher computational complexity can significantly disadvantage these systems.

Using ECDH key exchange in SHS offers several benefits: high security, forward secrecy, small key size, fast computation, key agility.

ECDH provides strong security, smaller key sizes, increased efficiency, interoperability, and forward secrecy in SHS. These benefits make ECDH an ideal choice for securing communication between healthcare providers and patients in SHS.

5. Implementation

The security of SHS can be enhanced by advanced encryption mechanisms such as AES-CCM and ECDH. The Advanced Encryption Standard with Counter with CBC-MAC (AES-CCM) is a block cipher encryption algorithm that provides both encryption and authentication of messages. AES-CCM operates on 128-bit blocks of data and supports 128-bit, 192-bit, and 256-bit keys.

By implementing AES-CCM with ECDH in an SHS with 5G-AKA, the security and privacy of patient information can be enhanced, ensuring that sensitive data is protected from unauthorized access and tampering.

The algorithm for the AES-CCM and ECDH encryption over SHS and 5G-AKA.

1. Generate an ECDH key pair using the SECP256R1 curve.
2. Generate a private key for the server using the same curve.
3. Generate the public key for the server from its private key.
4. Exchange the private key with the server's public key using ECDH to get a shared secret.
5. Use HKDF (HMAC (hash-based message authentication code) Key Derivation Key) with SHA256 to derive a 256-bit symmetric key from the shared secret with salt value as None and info value as b'smart healthcare'.
6. Generate a random 12-byte nonce for AES-CCM encryption.
7. Create a dictionary named data containing a patient ID, heart rate, and temperature as key-value pairs.
8. Serialize the data dictionary into a JSON string and encode it as bytes.
9. Generate a random 32-byte session key for 5G-AKA.
10. Concatenate the nonce, JSON data, and session key to form the 5G-AKA token.
11. Use AES-CCM with the derived symmetric key and the 12-byte nonce to encrypt the 5G-AKA token.
12. Decrypt and verify the 5G-AKA token using AES-CCM with the derived symmetric key and the same 12-byte nonce.
13. If the decrypted token matches the original token, print "Authentication successful".
14. Otherwise, print "Authentication failed".

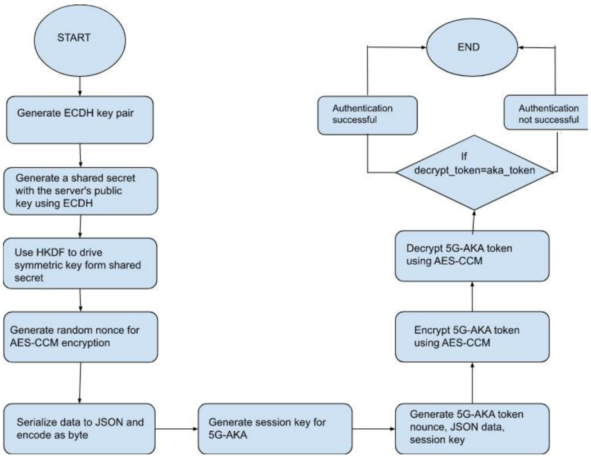
START
Generate ECDH key pair
Generate a shared secret with the server's public key using ECDH
Use HKDF to drive symmetric key form shared secret
Generate random nonce for AES-CCM encryption
Serialize data to JSON and encode as byte
Generate session key for 5G-AKA
Generate 5G-AKA token nounce, JSON data, session key
Encrypt 5G-AKA token using AES-CCM
Decrypt 5G-AKA token using AES-CCM
If decrypt_token=aka_token
Authentication successful
Authentication not successful
END

Figure 3: Flowchart of AES-CCM and ECDH encryption over SHS and 5G-AKA

This is the general flowchart of SHS using AES-CCM and ECDH over the 5G-AKA network. These are the steps that are followed in the implementation.

6. Results and Analysis

6.1 Experiment 1: Analyzing Encryption for various types of data sets

This experiment uses code to compare how well ECC and AES-CCM perform encryption on various data quantities.

- A public key is generated using an ECC private key. The public key is then converted into bytes and serialized. The code then generates random data that needs to be encrypted in various sizes. For each amount of data, it then creates an ephemeral ECC key pair and uses ECDH key agreement to derive a shared key.
- The shared key is then used to derive an encryption key using HKDF with a random salt. A random 96-bit nonce is also generated, and the data is encrypted using the AES-CCM cipher object. The encryption time is then calculated and stored in a list. Finally, the encryption times are plotted against the data sizes.

Overall, the plot sheds light on how the encryption method performs for various data volumes and can be used to fine-tune the method for certain use cases. For instance, optimizing the scheme for tiny data quantities may be more crucial than optimizing for bigger data sizes if the method is typically used to encrypt small amounts of data.

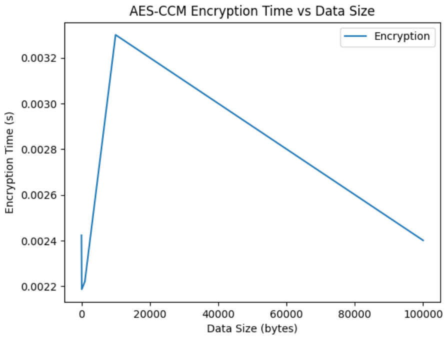

Figure 4: Encryption for various data sets

6.2. Experiment 2: Simulating Encryption for various types of data sets

This experiment uses code to compare how well ECC and AES-CCM perform encryption on various data quantities and simulate it 10 times.

- A public key is generated using an ECC private key. The public key is then converted into bytes and serialized. The code then generates random data that needs to be encrypted in various sizes. For each amount of data, it then creates an ephemeral ECC key pair and uses ECDH key agreement to derive a shared key.
- The shared key is then used to derive an encryption key using the HKDF with a random salt. A random 96-bit nonce is also generated, and the data is encrypted using the AES-CCM cipher object.
- The encryption times refer to the length of time needed to use the implemented encryption algorithm to encrypt a specific amount of data. The program estimates the encryption time for five different data sizes, including 10, 100, 1000, 10,000, and 100,000 bytes, and records the results in a list called encryption times.
- The overhead of the encryption process is the reason why small data sets require more time to encrypt than large data sets. An ephemeral ECC key pair is generated, ECDH is used to create a shared secret key, HKDF is used to derive an encryption key, a random nonce is generated, and an AES-CCM cipher object is created.

The time required to complete these procedures for small data sets is comparable to the time needed to encrypt the data itself, leading to a substantially longer encryption time. Contrarily, for big data sets, the time required to carry out these procedures is insignificant in comparison to the time required to encrypt the data, leading to a substantially shorter encryption time.

6.3 Experiment 3: Attack on the derived key

Using ECDH key exchange, HKDF, and Authenticated Encryption with Associated Data (AEAD) encryption using AES-CCM, the code is a Python script that illustrates a secure communication protocol between two participants. The protocol tries to protect the communication's integrity and secrecy.

- The code creates a shared secret using the server's public key using ECDH after first creating a private key with the SECP256R1 curve. To generate a symmetric key for encryption and decryption, the shared secret is fed as an input into the HKDF algorithm.
- A patient ID, heart rate, and temperature are then included in an example data dictionary that is created by the code. The information is encoded as bytes and serialized to JSON. The token-bytes() function of the secrets module is used to create a session key.
- The AES-CCM encryption key is then derived by the code utilizing HKDF and the 5G-AKA label. After that, the derived key, a random nonce, and no related data are used to encrypt the JSON data and session key using AES-CCM.
- The last byte of the encrypted data is then changed to a null byte to mimic an assault. Decryption will be unsuccessful because of this alteration since it compromises the integrity of the encrypted data. After the encrypted data, the decryption procedure anticipates receiving a tag length of 8 bytes. However, the assault has resulted in an erroneous tag length, which prevents the decryption from succeeding.
- The script then uses the resulting key and nonce to attempt to decrypt the changed encrypted data using AES-CCM. The JSON data and session key are taken from the decrypted data if the decryption is successful. The authentication is successful if the original data and session key match the retrieved JSON data and session key. If not, authentication is unsuccessful.

The decryption and authentication failed because of the attack, which involved changing the last byte of the encrypted data. To stop assaults on the communication protocol, it is crucial to guarantee the integrity of the encrypted data, as shown by this attack.

6.4 Experiment 4: Attack on the encryption data

Using ECDH key exchange, HKDF, and AEAD encryption using AES-CCM, the code is a Python script that illustrates a secure communication protocol between two participants. The protocol tries to protect the communication's integrity and secrecy.

- The code creates a shared secret using the server's public key using ECDH after first creating a private key with the SECP256R1 curve. To generate a symmetric key for encryption and decryption, the shared secret is fed as an input into the HKDF algorithm.
- A patient ID, heart rate, and temperature are then included in an example data dictionary that is created by the code. The information is encoded as bytes and serialized to JSON. The token-bytes() function of the secrets module is used to create a session key.
- The AES-CCM encryption key is then derived by the code utilizing HKDF and the 5G-AKA label. After that, the derived key, a random nonce, and no related data are used to encrypt the JSON data and session key using AES-CCM.
- The attacker may intercept the public key exchange and substitute their own public key, enabling them to establish a shared secret with the client and perform a man-in-the-middle attack.
- The attacker may be able to guess or brute-force the shared secret generated from the ECDH key exchange, allowing them to derive the symmetric key and decrypt the data.
- If the attacker can obtain the salt used in the HKDF key derivation function, they may be able to derive the symmetric key from the shared secret without performing the ECDH key exchange.
- If the attacker can intercept the encrypted data and modify it before it reaches the recipient, they can change the data or substitute it with their own data, causing authentication to fail. Alternatively, they may be able to modify the nonce or the tag length, causing the decryption to fail.

An attack scenario has been demonstrated where an incorrect nonce length is used in the encryption process, causing the encryption to fail. However, this attack does not compromise the security of the system, as the decryption will fail as well.

6.5 Experiment 5: Attack on the nonce (random stream)

Using ECDH key exchange, HKDF, and AEAD encryption using AES-CCM, the code is a Python script that illustrates a secure communication protocol between two participants. The protocol tries to protect the communication's integrity and secrecy.

- The code creates a shared secret using the server's public key using ECDH after first creating a private key with the SECP256R1 curve. To generate a symmetric key for encryption and decryption, the shared secret is fed as an input into the HKDF algorithm.
- A patient ID, heart rate, and temperature are then included in an example data dictionary that is created by the code. The information is encoded as bytes and serialized to JSON. The token-bytes() function of the secrets module is used to create a session key.
- The nonce, JSON data, and session key are encrypted using AES-CCM with the derived key. The encrypted data is then decrypted using AES-CCM with the wrong nonce, which leads to a decryption failure.
- The attack happens when the wrong nonce is used for AES-CCM decryption. The nonce is randomly generated and used for both encryption and decryption.
- If the wrong nonce is used for decryption, the decrypted data will be garbage. In this example, the decrypted data is expected to contain both the JSON data and the session key. If the decryption fails, the session key will not be correctly extracted, and authentication will fail.

This attack is an example of a replay attack, where an attacker captures the encrypted data and then replays it with a different nonce to try to gain access to the data. The use of a random nonce for each encryption ensures that a replay attack cannot be successful, assuming the nonce is kept secret.

7. Conclusions

Using AES-CCM and ECDH encryption in SHS and 5G-AKA can mitigate all STRIDE threats and can help mitigate a wide range of security threats related to the STRIDE model in SHS and 5G-AKA. By implementing these security measures, organizations can improve the confidentiality, integrity, and availability of their sensitive data and resources.

The integration of SHS with 5G technology brings forth additional benefits and advancements. The 5G network's high data speeds, low latency, reliability, and massive connectivity enable seamless and real-time communication between healthcare providers and devices. This facilitates remote healthcare support, telemedicine consultations, and the incorporation of emerging technologies such as augmented reality (AR) and virtual reality (VR) into healthcare practices.

The implementation of AES-CCM and ECDH encryption, combined with 5G technology, strengthens the security measures and privacy standards within the Smart Healthcare System. Patient information remains protected, ensuring that sensitive data is not vulnerable to unauthorized access or tampering. The enhanced communication capabilities provided by 5G enable healthcare professionals to deliver improved patient care, offer remote monitoring and diagnosis, and optimize resource utilization.

Moving forward, we suggest future work to investigate the scalability of the protocol. Healthcare systems often involve many devices and servers, and it is crucial to ensure that the proposed protocol can handle multiple devices and servers and a high volume of traffic. This will help ensure that the protocol can meet the demands of a large-scale healthcare system without compromising security or performance.

8. References

[1] R. Ahmed, M. H. Ahmed, S. U. Hassan, and A. H. Baig (2021). Internet of Things (IoT)-enabled healthcare system: A comprehensive review. IEEE Reviews in Biomedical Engineering, 14:284–298.

[2] Anjali Sharma and Ritu Arora (2017). A comparative analysis of symmetric key cryptography algorithms. International Journal of Computer Applications, 171(16):33–38.

[3] Xiaoyong Tang, Lijun Zhao, Jing Chong, Zhengpeng You, Lei Zhu, Haiying Ren, Yuxiang Shang, Yantao Han, and Gong Li (2022). 5G-based smart healthcare system designing and field trial in hospitals. IET Communications, 16(1):1–13.

[4] Baozhan Chen, Siyuan Qiao, Jie Zhao, Dongqing Liu, Xiaobing Shi, Minzhao Lyu, Haotian Chen, Huimin Lu, and Yunkai Zhai (2021). A security awareness and protection system for 5G smart healthcare based on zero-trust architecture. IEEE Internet of Things Journal, 8(13):10248–10263.

[5] Lei Yang, Shuai Zhang, Feng Tian, Shuai Ren, and Hongnian Yu (2020). A secure data sharing scheme based on AES for cloud-based electronic health records. IEEE Access, 8:124444–124452.

[6] Yuyu Zhang, Yu Zhu, Yufeng Cheng, Hongliang Zhang, and Jing Wang (2019). A secure data encryption scheme for mobile health systems. IEEE Access, 7:64391–64399.

[7] S. Singh and S. Kaur (2018). A comparative study of symmetric key encryption algorithms for secure communication in smart healthcare systems. International Journal of Computer Applications, 179(22):6–11.

[8] Alok Gupta and Amarjeet Singh (2019). A comparative study of cryptographic algorithms for securing smart healthcare systems. Journal of Medical Systems, 43(7):1–9.

[9] Yalong Tang, Xiang Chen, and Xinhua Wang (2021). Smart healthcare system architecture based on internet of things. International Journal of Distributed Sensor Networks, 17(1):1550147721994408.

[10] Naser Alsafi, Steven L Gao, and Majid Alshammari (2021). Smart healthcare system architecture using IoT and cloud computing. Wireless Communications and Mobile Computing, 2021:6647141.

[11] Ke Yang, Jinchang Ren, and Xiaochun Liu (2021). A privacy-preserving and secure IoT-based healthcare system with edge computing. IEEE Transactions on Industrial Informatics, 17(6):4065–4073.

[12] S. A. Al-Kaabi, M. M. Hassan, and M. Al-Qutayri (2020). Security analysis of smart healthcare system using blockchain technology. In 2020 International Wireless Communications and Mobile Computing (IWCMC), pages 2392–2397. IEEE.

[13] Mohammed H. B. Al Rawi and Mahmoud Al-Qutayri (2021). A blockchain-based framework for securing healthcare systems. IEEE Access, 9:48076–48090.

[14] Loren Kohnfelder, David Rine, and Gregory White (2005). Stride: A threat modeling tool for identifying security threats. Journal of Information Warfare, 4(2):31–42.

[15] Abdulrahman Al-Hezmi, Abdulaziz Al-Nahari, Mohammed Alqahtani, Ahmad Alkahtani, and Abdulrahman Alghamdi (2021). 5G security: analysis of potential threats and solutions. Wireless Networks, 27(6):4413–4429.

[16] Yutao Sun, Ang Li, Yong Liu, Lei Zhou, and Zhe Fang (2021). 5G core networks: security challenges and opportunities. IEEE Communications Magazine, 59(5):50– 56.

[17] Hongbing Song, Lei Lu, and Hongxiang Jiang (2020). 5G core network architecture and security. In 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pages 135–140. IEEE.

[18] S. Srinivasan and A. Ananth (2021). A novel authentication protocol for healthcare system using aka algorithm. International Journal of Engineering and Technology Innovation, 11(1):53–60.

[19] M. T. Dehghani and A. R. Sadeghnejad (2021). Secure authentication scheme for e-healthcare systems using aka protocol and blockchain technology. Journal of Medical Systems, 45(12):174.

[20] A. Al-Zoubi and S. Alkhraisat (2021). Secure authentication scheme for mobile healthcare using aka protocol and blockchain. Journal of Ambient Intelligence and Humanized Computing, 12(12):11851–11860.

[21] Abiodun Olusola Afolabi, Adedoyin Adeyinka Alaba, Olumide Olawale Olabiyi, and Olumide Oluwaseun Akintola (2021). Lightweight cryptography for the internet of things: A review. International Journal of Communication Systems, 34(12):e4817.

[22] Abdellah Benali and Abdelhakim Erritali (2021). Performance evaluation of lightweight encryption algorithms for IoT devices. In International Conference on Computational Science and Its Applications, pages 293–303. Springer.

[23] Muhammad Umer Rehman, Muhammad Umar Farooq, and Muhammad Asad Khan (2020). A comprehensive performance evaluation of lightweight block ciphers for IoT devices. IEEE Access, 8:100303–100319.

[24] Yanli Sun, Xiong Li, Yan Li, and Keqiu Zhang (2021). A lightweight authentication and data confidentiality scheme for industrial Internet of things. IEEE Transactions on Industrial Informatics, 17(6):4276–4285.

[25] Mohammed Al-Haidari, Mznah Al-Rodhaan, and Abdullah Al-Dhelaan (2021). A comparative study of AES-CCM and chacha20-poly1305 in IoT-based healthcare systems. Sensors, 21(15):5034.

[26] Adel Ammar, Ahmed Ben Abid, and Abdelfettah Belghith (2021). Performance evaluation of DHE and ECDH key exchange algorithms for secure communication in IoT. In 2021 IEEE International Conference on Consumer Electronics (ICCE), pages 1–6. IEEE.