# Remote ECG System Data Encryption Scheme

Liu Xin
College of Information Science and Engineering
Ocean University of China
Qingdao, China
e-mail: liuxinouc@126.com

Wei Zhiqiang
College of Information Science and Engineering
Ocean University of China
Qingdao, China

***Abstract*—The remote ECG communicates over the public Internet, and the public Internet security is not sufficient to ensure the information security of the remote ECG system. User data disclosure poses a security threat to the system. The ECG information security solution proposed in this paper adopts the method of preset multi-key-per-machine at the factory to avoid leakage caused by personnel factors, and uses the hash function to generate a secondary password for communication encryption during network transmission. Experiments show that this program can effectively resist the existing various attack methods and fully protect patient data security.**

***Keywords-Remote ECG System; Multi-Key-Per-Machine***

## I. INTRODUCTION

In the remote ECG system, because the terminals are distributed in the vast number of families, clinics, community hospitals, etc., there are problems in the management of physical equipment. Since the ECG data is communicated through the Internet, there is a problem that the communication data packet can be intercepted and monitored. Due to the diversification of commercial channels and construction teams, there is also a risk of technical confidentiality leaks.

This paper proposes a remote ECG system data encryption mechanism, which generates unable-to-remember keys by means of a random number generator, establishes a multi-key model, and implements the encryption storage and secure communication technology for periodic replacement of keys.

The core design concept of this program is: human beings as a confidential subject is also a key factor in generating leaks. If every link of encryption is implemented by an automatic algorithm and the artificial factors are completely eliminated, the patient data security can still be effectively protected in the presence of the above security management vulnerability.

## II. DESIGN OF INFORMATION SECURITY SCHEME FOR REMOTE ECG SYSTEM

The key protocol developed in this paper contains mechanisms for key generation, storage, deployment, and verification. This scheme takes into account various attacks such as brute force cracking by attackers and interception of network communication packets.

### *A. Hardware risk control scheme*

The hardware of the device may be disassembled, and the data on the storage device can be taken out. If the attacker can obtain two devices of the same model, the content of the key can be known through the content comparison.

A common method of hardware attacks is the wiretap method. For example, a logic analyzer records and analyzes the timing of signals on the memory bus of the memory device. Practice has proved that ordinary memory devices, such as Nor Flash chip, Nand Flash chip or I2C bus serial memory cannot resist the wiretap method attack.

Therefore, we must use the Tamper-Resistant Security Module (TRSM) to store the encryption key. In theory, pure software encryption cannot guarantee the security of the key when there is a possibility that the hardware is disassembled. This solution uses a TRSM with a capacity of 1k bytes, which costs about $0.50, and the cost increase for the entire device is negligible.

### *B. Research on Key Agreement of multi-key-per-machine*

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

A multi-key-per-machine encryption algorithm is used to encrypt patient data in the terminal device memory. Patient data can be sent to the cloud very quickly, but we need to consider the device features in the off-network state. So local storage is still an indispensable feature.

Considering that the terminal equipment needs to deal with many tasks such as write memory, waveform curve drawing, network communication, alarm and other emergency processing, the encryption algorithm should not adopt an overly complex algorithm. The key point of prevention is to be able to prevent attackers from implementing bulk cracking on same model devices. Therefore, the key issue of encryption is the key design and the key storage.

DOI: 10.21307/ijanmc-2019-018

### C. Key design

User data (patient ECG data) is encrypted using a block cipher encryption scheme. A block cipher is also called a symmetry cipher. When encrypting plaintext by using a block cipher, firstly, the plaintext needs to be grouped, each group has the same length, and then each set of plaintexts is separately encrypted to obtain a ciphertext of the same length. A block cipher is characterized by the same encryption key as the decryption key. This scheme adopts the DES encryption standard, the packet length is 64 bits, and the key length is also 64 bits. There are 8 parity bits in the key, and the actual key length is 56 bits. Although DES has been replaced by AES, the encryption strength is sufficient for this application and the computational complexity is lower than AES. The modular multiplication that AES needs to use is too computationally intensive for embedded processors and is not suitable for encrypting real-time collected ECG data streams.

### D. Key security strength

Considering the possibility that the device-side key is forcibly cracked by an attacker, a mechanism for periodically changing the key can be adopted. In general, the life of an electrocardiograph is 10 years. If you change the key once a month, you need 120 keys for a total of 960 bytes. The cloud server's cost of storing 1 million groups of lifetime keys is 960M, which is still affordable.

### E. Key generation

The composition of the key on each terminal device: 120 random numbers are stored as a key sequence, and the total length is 960 bytes. The server can generate a sequence of random numbers with a length of 960M at a time, and then sequentially intercept 960 bytes for each device. The key of the device and the offset of the key sequence in the server-side master key are recorded in the device's secure memory. The server records the product serial number of each device and the corresponding random number sequence index value. As long as the random number is guaranteed not to be copied by internal spies during the manufacturing process, the security of the key can be ensured.

### F. Key deployment

The device should have a 1K capacity of secure memory(TRSM). The 120 user data encryption keys (960 bytes long random number) of the full life are loaded into the machine's secure memory at one time when the machine is shipped from the factory. If the device is out of service, you can use these keys in a rolling loop. The device does not update the key after it leaves the factory, preventing the key from being intercepted by the attacker during delivery.

After the device leaves the factory, it is no longer necessary to transfer the user data encryption key under any condition, which eliminates the possibility of the key data packet being intercepted.

### G. Key verification

The main purpose of key verification is to test whether the encryption-related hardware and software systems are defective.

After the machine is assembled, read out the 120 key values in the secure memory, and the MD5 value of the keys is calculated and sent to the cloud server along with the product serial number. If the server-side verification is successful, the key can be considered to have been deployed correctly. If you are worried about the collision of MD5 values, you can pre-calculate all device keys on the server side in advance, and delete the key segments with MD5 duplicates. The probability of such a problem in theory is extremely small.

### H. Key synchronization

Through the above steps, we deployed the same user data encryption key sequence on the terminal device and the cloud server, but how to use the two is still a problem. Ideally, using timestamp as a synchronization parameter is the best solution. The two parties can agree that, from the date of leaving the factory, the natural month is used as the measurement granularity, and a new key is replaced every natural month.

The problem is that in the actual use environment, the time accuracy of the device side is completely unsecured. On the one hand, the RTC (Real Time Clock) devices based on quartz crystal oscillators do not achieve such precision, and some devices run faster than others. On the other hand, the RTC backup power failure is difficult to avoid, and the device cannot be abandoned because the clock is inaccurate. Furthermore, if the user is allowed to adjust the device time, the attacker can take advantage of this adjustment opportunity to detect the subsequent ciphertext form in advance, which will be a significant security hole.

In the case of networking, the server can know the number of key-used-times on the terminal device. When the server finds that the consumption speed of the terminal key is too slow, it is expected that the pre-stored 120 keys will not be used in the entire product life cycle, and the maximum-key-used-times value can be modified by the key agreement protocol (change the maximum-key-used-times to be small).

First, a temporary communication key for modifying the maximum-key-used-times value is negotiated using the Diffie-Hellman key exchange protocol. The generator element p of the large prime number Q and the rational number domain FQ is required to be pre-agreed between the server and the device. Q and p are written into the device's TRSM at the factory. The algorithm for generating the temporary communication key K is as follows:

- 1, The server generates a random number x, calculates A = px mod Q, and sends it to the terminal;
- 2, The terminal generates a random number y, calculates B = py mod Q, and sends it to the server;
- 3, The terminal calculates K = Ay mod Q, and the server calculates K = Bx mod Q, which proves that the two K values are equal.

In this way, the server and the terminal share the random number K as a temporary communication key for modifying the maximum-key-used-times, and the attacker cannot detect the communication content. Even if the content can be cracked by the attacker, it is only an index value, and the key cannot be derived.

The server should record the timestamp of modify the maximum-key-used-times value to avoid modifying it too frequently. Modify it 1 to 3 times a month is enough. In the case where the hardware can ensure the security of the key, it is not necessarily meaningful to change the key too frequently.

## III. DATA COMPRESSION UPLOAD SCHEME DESIGN

Channel eavesdropping is a common attack method. In the Internet environment, it is easy to intercept communication packets on the network line by means of route bypass. The scheme adopts data compression and then encrypts and transmits, as shown in Fig. 1, which further increases the attacking difficulty of the attacker while reducing the load of the network. After the cracker obtains the message, the compressed file can be restored by the exhaustive cracking method; since the key used for the two encryptions is different, the original data obtained from the compressed file needs to be cracked again. A legitimate cloud server stores the original key from the factory. It is easy to calculate the compression key from the original key. Conversely, since the hash function is irreversible, it is not feasible to derive the original key from the compression key, and the attacker can only crack it by the exhaustive method.

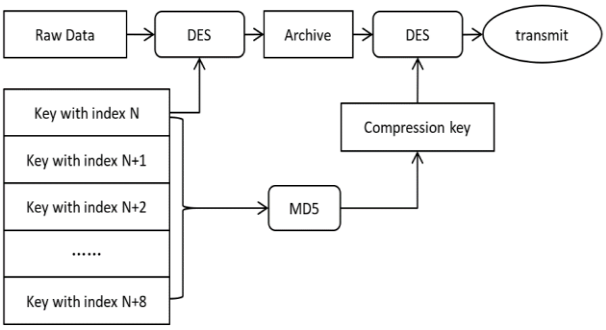


Figure 1. Two encryption processes

Both ciphers in Fig. 1 use the DES algorithm. The hash function uses the MD5 algorithm. MD5 requires a 64-byte input to generate a 16-byte output. In order to meet the input requirements of MD5, the consecutive 8 keys in the key queue are taken as a set of parameters. After the key expires and the next key is replaced, the parameter group is pushed back and used in turn.

Data compression can use Shannon coding or Huffman coding. But today there are many mature methods that can be used, and there is no need to reinvent the wheel for a little performance boost. We use two classical algorithms to compress a 100M size data file. The performance is shown in Table 1.

It can be seen that the data is de-duplicated and then compressed, and the compression ratio is the highest, which is beneficial to network transmission. The amount of data generated by the ECG machine per minute is about 0.7M, and the compression algorithm takes less than 2 seconds, which is equivalent to 1/30 processor occupancy, and the system can fully withstand this burden.

TABLE I. CLASSIC COMPRESSION ALGORITHM PERFORMANCE TEST

| ***Algorithm*** | ***Compression Ratio*** | ***Time Consuming (Seconds)*** |
|---|---|---|
| gzip (Compression) | 36.2% | 452 |
| dedup (Remove Duplicates) | 51.5% | 78 |
| First gzip and then dedub | 26.8% | 468 |
| First dedub and then gzip | 14.6% | 203 |

## IV. EXPERIMENT AND VERIFICATION

In order to verify the effectiveness of the security system, the following practical application scenarios are designed to carry out crack attacks on the remote ECG system.

Attack target: obtain the patient raw data.

Experimental conditions: Provide 10 attack devices and an ECG signal generator to the attack team, and informed the key length and encryption algorithm, the attacker tried on an Intel E5-2660 (ten core) server, perform no more than 100 hours of calculations on each machine to obtain its patient raw data.

Experimental results: After several times of improving the attack plan, the attacker violently cracked the data records stored in the device by exhaustive cryptography, and successfully cracked two terminal devices within the specified calculation time, and the cracking time was 75 hours and 40 hours respectively.

In the case that the device is not disassembled, the communication message received by the network cannot be successfully cracked.

Experiments prove that this system is not completely unbreakable in the case of hardware damage, but the cost of cracking is very expensive, and it is not worth the loss.

## V. CONCLUSION

The ECG information security solution proposed in this paper adopts the method of preset multi-key-per-machine at the factory to avoid leakage caused by personnel factors, and uses the hash function to generate a secondary password for communication encryption during network transmission. Experiments show that this program can effectively resist the existing various attack methods and fully protect patient data security.

## ACKNOWLEDGMENT

This work is supported by The Aoshan Innovation Project in Science and Technology of Qingdao National Laboratory for Marine Science and Technology (No.2016ASKJ07)

## REFERENCES


[1] Chen, Zhuang, F. Qi, and C. Ye. "Research on Cloud Data Encryption Scheme Based on Chinese Cryptographic Algorithms." Journal of Information Security Research (2018).

[2] Huang, Pei, et al. "A Robust and Reusable ECG-Based Authentication and Data Encryption Scheme for eHealth Systems." Global Communications Conference IEEE, 2017:1-6.

[3] Quang, Do Vinh, T. N. K. Hoan, and I. Koo. "Energy-Efficient Data Encryption Scheme for Cognitive Radio Networks." IEEE Sensors Journal PP.99(2018):1-1.

[4] Liang, Kaitai, et al. "An Efficient Cloud-Based Revocable Identity-Based Proxy Re-encryption Scheme for Public Clouds Data Sharing." European Symposium on Research in Computer Security Springer, Cham, 2014:257-272.

[5] Xu, Lei, X. Wu, and X. Zhang. "CL-PRE:a certificateless proxy re-encryption scheme for secure data sharing with public cloud." ACM Symposium on Information, Computer and Communications Security ACM, 2012:87-88.

[6] Gillies, Alan. "Improving the quality of information security management systems with ISO27000." Tqm Journal 23.4(2011):367-376.

[7] Publishing, It Governance. Iso27000 and Information Security: A Combined Glossary. It Governance Ltd, 2010.

[8] Wang, Chi Hsiang, and D. R. Tsai. "Integrated installing ISO 9000 and ISO 27000 management systems on an organization." 2009 International Carnahan Conference on Security Technology IEEE, 2009:265-267.