IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 19, 2020, accepted February 13, 2020, date of publication February 20, 2020, date of current version March 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2975208

A Secure Energy-Saving Communication and Encrypted Storage Model Based on RC4 for EHR

JINQUAN ZHANG [1,2], **HAORAN LIU** [1], **AND LINA NI** [1,2]

[1] College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China
[2] Key Laboratory of the Ministry of Education for Embedded System and Service Computing, Tongji University, Shanghai 201804, China

Corresponding author: Lina Ni (nln2004@163.com)

This work was supported in part by the National Key Research and Development Programs Project of China under Grant 2017YFC0804406, in part by the Training Program of the Major Research Plan of NSF of China under Grant 91746104, in part by the Open Project of Tongji University Embedded System and Service Computing of Ministry of Education of China under Grant ESSCKF 2019-06 and Grant ESSCKF 2019-08, in part by the Shandong Key Research and Development Project under Grant 2019GGX101066, and in part by the GIF of Shandong University of Science and Technology under Grant SDKDYC190336.

ABSTRACT The commercialization of 5G has greatly promoted the development of medical Internet of Things (IoT). More medical devices connected to the Internet may further increase the communication power consumption. Meanwhile, privacy protection technique in cloud computing cannot match the rapid development of medical applications. Therefore, exploring secure, balanced and energy-efficient data transmission between medical devices and cloud servers is extremely challenging. This paper focuses on the security and energy consumption of medical electronic health record (EHR) data transmission and storage between cloud server and IoT device users. We build a secure energy-saving communication and encrypted storage model by adding secure energy-saving communication scheme and encryption algorithm to the traditional medical cloud model. Specifically, we propose a communication authentication algorithm MedGreen based on elliptic curve and bilinear pair. In the algorithm, the two communication parties can complete the key establishment and identity authentication only after one communication, which effectively balances the resource overhead of the key center and the user, and resists the Man-in-the-middle attack. Aiming at the characteristics of large repetition and high sensitivity of medical data, we present a secure data storage algorithm MedSecrecy based on Huffman compression and RC4. The algorithm not only maintains the RC4 encryption efficiency, reduces the amount of cipher text data, but also improves confidentiality, randomness and security of the key stream. Comprehensive analysis and simulations show that our system is secure, energy-saving and highly efficient for EHR.

INDEX TERMS Electronic health records (EHR), Internet of Things (IoT), secure communication, data encryption, privacy protection.

I. INTRODUCTION

With the commercialization of 5G wireless networks [1], [2], the speed of medical Internet of Things (IoT) [3]–[5] devices accessing Internet will be faster in the future. This will cause further expansion of medical big data, and more medical applications or Internet+ medical products will appear. Due to the lack of local computing and storage capabilities of medical institutions, storing medical data in cloud is considered the most ideal solution [6]. Cloud service providers also provide users and medical institutions with diverse medical management platform [7].

Medical data is commonly stored in the form of Electronic Health Records (EHR). Generally, these data can be collected directly through hardware devices and uploaded to cloud after being encrypted. However, access to more IoT devices may further increase communication power consumption and cause more privacy leakage issues in access control, sharing and storage [8], [9] etc.

For medical cloud access control, identity authentication and digital signature are two indispensable parts. Sharaf and Shilbayeh [10] proposed a multi-authority cipher text policy

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Khalil Afzal.

and attribute-based encryption EHR framework for the secure access control of government to citizens (G2C). The aim is to provide high secure EHR services for governments and users. Similarly, Riad *et al.* [11] seized the parallel characteristics of cloud computing, and built an EHR access control mechanism to deal with the issue of identity authentication in a cloud hosting environment.

It is noted that the sharing of medical data is inseparable from excellent authentication schemes and effective encryption algorithms. Aiming at the data security and privacy protection of data sharing between different medical institutions, Wang *et al.* [12] established a consortium blockchain-based EHR storage framework to improve the security of medical data sharing. Liu *et al.* [13] proposed a medical data sharing and protection scheme via blockchain to clarify the security features in medical treatment.

Reliable storage solution can provide a solid foundation for the sharing of medical data [14]. Seol *et al.* [15] made the traditional fine-grained access control more flexible via constructing an EHR cloud storage model based on extensible access control markup language. Zhang *et al.* [16] outsourced the shared EHR reconstruction task to a cloud service provider, solving the problem of resource reconstruction for medical centers and patients.

It is worth noting that user's cloud operation of medical data is inseparable from a secure communication solution [17]. The most important part of communication is the encryption algorithm, and RC4 [18] and Elliptic curve cryptography (ECC) [19] are both integral parts of it. RC4 algorithm is simple to implement, fast in encryption speed, and has good adaptability to hardware devices. While ECC algorithm has the characteristics of short key length and high security. To this end, Zheng and Imai [20] first proposed a new paradigm combining signature and encryption to greatly reduce the computational cost and communication overhead. Hwang *et al.* [21] presented a signcryption scheme with forward secrecy based on elliptic curve to further reduce the cost of the sender and confirm the authentication function of the third party. Nayak [22] proposed a message encryption scheme based on Elliptic Curve Discrete Logarithm Problem (ECDLP) and Elliptic Curve Diffie-Hellman (ECDH) problem to improve the security of communication authentication.

Although scholars have done numerous research on the security of EHR in cloud, we still observe two limitations of it. Firstly, in terms of communication, the communication overhead of existing methods is huge. Not only the sender and receiver need to pass messages multiple times, but the authentication of existing communication schemes often relies too much on third-party authentication. These may cause lots of data blocking and user waiting time. Meanwhile, existing models are not suitable for complex medical scenarios. Furthermore, the inadequacy of existing models will provide more cipher text cracking opportunities for attackers with strong computing power. Secondly, in the data storage stage, traditional data encryption scheme does not combine the characteristics of medical data as well. This may increase the time and space overhead required for encryption. Moreover, most encryption algorithms are difficult to implement in medical cloud. Therefore, designing a secure energy-saving communication and secure storage system suitable for EHR remains an open challenge.

In this paper, we focus on the security of EHR data transmission and storage. In terms of information transmission, we should balance and save overhead while ensuring security. In the information storage area, we ought to ensure that the algorithm has good compatibility with hardware devices and optimal encryption efficiency. Fortunately, ECC and RC4 algorithms can solve these issues very well.

Based on the traditional medical cloud model, we propose a secure, balanced and energy-saving communication scheme and a confidential compression encryption algorithm for EHR. The communication scheme combines with the more secure elliptic curve bilinear pairing algorithm to balance the authentication between user and third-party, reducing the number of traditional verified information transmissions and resource overhead, and improving the security of the scheme. Furthermore, our encrypted storage algorithm combines the characteristics of medical data to reduce duplicate data. The proposed algorithm improves the security of cipher text data while ensuring the encryption efficiency. Our main contributions are summarized as follows:

1) Based on the traditional medical cloud model, we propose a secure, balanced energy-saving communication and encrypted storage model for EHR. Our model integrates low-overhead communication establishment schemes with high security encryption algorithms.
2) We propose a communication authentication scheme MedGreen based on elliptic curve and bilinear pairing. This solution breaks the situation that traditional medical models need to calculate the key first and then verify it, which combines the identity verification and key calculation as a whole. MedGreen can effectively balance the operations of third-party authentication centers and users and reduce communication overhead, and can resist man-in-the-middle attacks.
3) We present an EHR data encrypted storage algorithm MedSecrecy based on RC4 algorithm and Huffman compression algorithm. While maintaining the encryption efficiency of RC4, MedSecrecy enhances the randomness of the key stream and the confidentiality of the algorithm. In combination with Huffman compression algorithm, the amount of repetition in the data is reduced.

The rest of this paper is organized as follows. Section 2 introduces the related work in this paper. Section 3 presents preliminaries. A secure energy-saving communication and encrypted storage model is presented in Section 4. Section 5 presents protection technology in the model. Section 6 gives the simulation experiment. Finally, conclusions are made in section 7.

II. RELATED WORK

In order to tackle the medical cloud's security problems in data transmission and storage, researchers have proposed numerous security solutions.

For the storage security of data in cloud environment, in [23], Bao *et al.* proposed a new signal scrambling method to protect sensitive medical data and solve the security problems in complex network environments and sensitive data. In [24], Xue *et al.* presented a method combining cloud-side and data owner-side access control in encrypted cloud storage to resist DDoS/ESS attacks, provide resource consumption statistics, and reduce the system's construction overhead. In [25], Hu *et al.* proposed an enhanced Number Theory Research Unit (NTRU) cipher system to solve decryption failure problem in NTRU system. They also developed a new access policy based on the enhanced and verifiable access control scheme of NTRU.

The popularity of IoT mobile devices [26] has a profound impact on data transmission security and resource overhead. In [27], Hamid *et al.* proposed a bilinear pairing-based communication scheme built between cloud facilities and fog facilities, where the user-side key pairing has less overhead than other models. In [28], Sciancalepore *et al.* proposed a Key Management Protocol (KMP) based on Elliptic Curve Qu-Vanstone, which combined the certificate with the Elliptic Curve Diffie-Hellman exchange to shorten the maximum access time of both parties. In [29], Cai *et al.* proposed an open function handshake protocol based on improved ECDH, which is more lightweight than TLS and SSL. In [30], Rahman *et al.* proposed a Peer-to-Peer Data Management Systems (P2PDMS) data exchange security protocol based on paired encryption and data exchange policies. The protocol dynamically calculates the session key during data exchange and proves that it can resist man-in-the-middle attack, replay attack and masquerade attack through verification tools.

No matter in the field of data storage or data transmission, symmetric encryption is still the most widely used, most convenient and fastest data protection method. Generally, in medical IoT, RC4 algorithm is used for encryption at first, and then it is gradually replaced by AES algorithm due to the constant security vulnerabilities. However, RC4 algorithm has higher encryption efficiency and better adaptability to the encryption and decryption in the real scenario. Therefore, the research on RC4 algorithm is still proceeding in an orderly manner in recent years.

In [31], Maitra *et al.* proposed an RC4+ algorithm based on RC4. This algorithm introduces *IV* and zigzag replacement in the KSA phase, and increases the *i* and *j* pointer shift operations in the PRGA phase, which improves the safety of traditional RC4. In [32], Jindal *et al.* proposed three improved RC4 algorithms based on RC4+, which improve the randomness of key stream and shortened the encryption time. In [33], Orumiehchiha *et al.* summarized the merits and demerits of RC4(n, m) algorithm and proposed a key recovery attack method to crack the RC4(n, m) state.

In [34], Hammood *et al.* designed an RC4-2S algorithm based on RC4 to improve the randomness of the key stream by dividing the S-box into two equal parts. In [35], Weerasinghe proposed a modified RC4 algorithm to improve the confidentiality of RC4 algorithm. In [36], Xie *et al.* proposed an improved RC4 algorithm to further shorten the encryption time of the RC4 algorithm and expand the internal state of the S-box. In [37], Weerasinghe *et al.* proposed a double S-box effective RC4 based on the modified RC4 algorithm [35], and proved that effective RC4 has a better encryption effect under the shorter key length.

However, in terms of medical data communication, there are still limitations of unbalanced resource overhead and inadequacy of the medical cloud environment of existing methods. Moreover, since the tremendous amount of repeated medical data and high data security requirements are ignored in the encrypted storage stage of medical information, there exists the risk of large amounts of encrypted data and user privacy leakage during the implementation of encryption policies. In this paper, we design a secure energy-saving communication and encrypted storage model based on traditional medical cloud model, and propose the MedGreen communication authentication algorithm and MedSecrecy encryption algorithm to solve these issues.

III. PRELIMINARIES

A. ELLIPTIC CURVE AND ITS PROBLEM

Definition 1 (Elliptic Curve) [19]: An elliptic curve on $\mathbb{Z}_p (p > 3)$ is the set of all $(x, y) \in \mathbb{Z}_p$ that satisfy the conditions $y^3 \equiv x^3 + ax + b \bmod p$ and an infinite imaginary point ∞, where $a, b \in \mathbb{Z}_p$. Meanwhile, a and b satisfy the condition $4a^3 + 27b^2 \neq 0 \bmod p$.

The Elliptic Curve Discrete Logarithm Problem (ECDLP) and the Elliptic Curve Diffie-Hellman Problem (ECDHP) are extremely difficult mathematical problems to solve and very important for key protocols. Their definitions are as follows:

Definition 2 (Elliptic Curve Discrete Logarithm Problem) [38]: Let P and Q be two points of order n on the elliptic curve, and n is a prime number. When $k < n$, point $Q = k \times P$. Given these two points P and Q, find the correct k.

Definition 3 (Elliptic Curve Diffie-Hellman Problem) [39]: Let G be the base point of an elliptic curve of order n, n is a prime number, and $P = c \times G$, $Q = d \times G$. Given two points P and Q without c and d, solve $K = c \cdot d \times G$.

B. BILINEAR PAIRING AND ITS PROBLEM

Bilinear mappings [40] can be described by five-tuples (p, G_1, G_2, G_T, e), where G_1, G_2, G_T are cyclic groups whose orders are prime p, G_1, G_2 are additive groups and G_T is a multiplicative group. One mapping relationship e on three groups: $G_1 \times G_2 \rightarrow G_T$, satisfying the following properties:

1) Bilinearity: For any $P \in G_1$, $Q \in G_2$, $a, b \in \mathbb{Z}_p^*$, there exists $e(aP, bQ) = e(P, Q)^{ab}$.

2) Non-degeneracy: If P is a generator of G_1 and G_2, then $e(P, P)$ is a generator of G_T. In other words, $e(P, P) \neq 1$.
3) Computable: For any $P \in G_1, Q \in G_2$, there is an algorithm that can efficiently calculate $e(P, Q)$.

Computational Bilinear Diffie-Hellman problem and Decision Bilinear Diffie-Hellman problem [27] are difficult problems to solve in bilinear pairing. Computational bilinear Diffie-Hellman problem means that it is extremely difficult to calculate $e(P, P)^{abc}$ when $(P, aP, bP, cP \in G_1)$ are given, and where $(a, b, c) \in \mathbb{Z}_p^*$, P is a generator of G_1. The decision bilinear Diffie-Hellman problem refers to the problem of determining $e(P, P)^{abc} = r$ when the $(P, aP, bP, cP) \in G_1$ and $r \in G_T$ are known, and $(a, b, c) \in \mathbb{Z}_p^*$, P is a generator of G_1.

C. HASH ALGORITHM AND HMAC TECHNIQUE

In 1993, the US National Institute of Standards and Technology(NIST) and National Security Agency (NSA) set a new encryption standard, which called Secure Hash Algorithm (SHA) algorithm [41], [42]. Due to its irreversibility and uniqueness, it is widely used in the field of security information for file verification, digital signature, and identity authentication functions.

The importance of hash algorithm is that it can not only generate a fixed-length message code, such as SHA-1 [43] which can generate a 160-bit hash value message verification code, but also become part of the Hash-based Message Authentication Code (HMAC) [44]. The biggest advantage of HMAC message code over other message codes is that it can resist key prefix attacks and key suffix attacks. Its formula is:

$$HMAC_k(x) = h[(k^+ \oplus opad) \parallel h[(k^+ \oplus ipad) \parallel x]] \quad (1)$$

where h is the hash algorithm, k is the password, M is the input message, *opad* repeats the same number of times as the block length with $0x5c$, and *ipad* repeats the same number of times with the block length with $0x36$.

D. RC4 SYMMETRIC ENCRYPTION ALGORITHM

RC4 encryption algorithm [18] is a variable key length stream encryption algorithm designed by Ron Rivest in 1987. Because it has many characteristics such as simple implementation and fast encryption speed, RC4 has become the basis of Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA).

RC4 algorithm composes of two phases, which are the Key Scheduling Algorithm (KSA) phase and the Pseudo Random number Generation Algorithm (PRGA) phase. KSA phase first fills the S-box to reach 256 bytes. Subsequently, the order of the elements of S-box is disturbed using the key. Finally, PRGA phase generates a pseudo-random key stream and XOR encryption with the plain text to form a cipher text.

The two phases of RC4 algorithm are shown in Algorithm 1. Its time complexity is related to the length of the plain text. The larger the plain text, the longer the encryption time.

Algorithm 1 RC4 Symmetric Encryption Algorithm

```
Input: Key, Plain text.
Output: Cipher text.
    1. Key scheduling algorithm
for $i = 0$ to $n - 1$ do
    $S[i] = i$
end for
$j = 0$
for $i = 0$ to $n - 1$ do
    $j = (j + S[i] + Key[i]) \bmod n$
    swap($S[i], S[j]$)
end for
    2. Pseudo random number generation algorithm
$i = 0$
$j = 0$
while Plain text do
    $i = (i + 1) \bmod n$
    $j = (j + S[i]) \bmod n$
    swap($S[i], S[j]$)
    output $= S[(S[i] + S[j]) \bmod n]$
end while
```

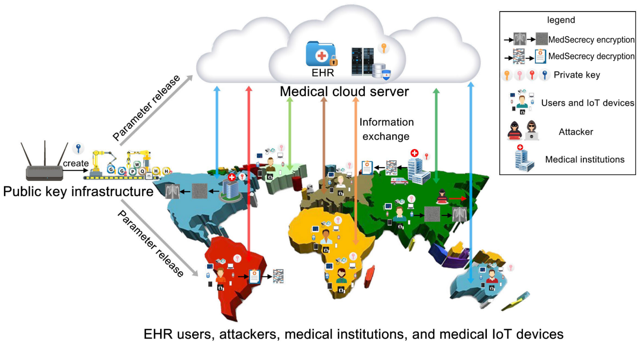


FIGURE 1. Secure energy-saving communication and encrypted storage model.

E. HUFFMAN COMPRESSION

Huffman coding [45], proposed in 1952, is a special type of best prefix coding and a variable length coding. It is usually used for lossless data compression.

Huffman compression consists of two phases, namely the statistical character frequency calculating phase and the tree building phase. At the beginning of encoding, it first calculates the frequency and weight of characters respectively. Then, it encodes both the characters with higher frequency near the root node and the characters with lower frequency near the leaf node to achieve the best compression encoding effect. The algorithm is shown in Algorithm 2.

IV. SYSTEM MODEL

This section will propose a secure energy-saving communication and encrypted storage model based on bilinear pairing and RC4 algorithm, the model is shown in Fig. 1.

Algorithm 2 Huffman Compression Algorithm

```
Input: Uncompressed file.
Output: Compressed file.
Count all symbol probabilities and sort them in ascending order
Create a leaf node for each symbol and add its frequency of occurrence
while there are multiple nodes in the queue do
    i. Remove the two nodes with the lowest probability or frequency from the queue
    ii. Assign 0 to the second smallest element and 1 to the smallest element
    iii. Create a new node with these two nodes as children and set their probabilities equal to the sum of the probability of these two nodes
    iv. Add this new node to the queue
end while
When the remaining node is the root node, the tree building is completed
Calculate entropy, average length and redundancy (Round-2 Ends)
After getting the code for each symbol/alphabet, connect them
Calculate the probability of 0 and 1 again
Repeat steps 1 to 10
```

A. MODEL OVERVIEW

As can be seen from Fig. 1 that our model consists of 6 components, that is, EHR users and IoT devices, medical cloud servers, public key infrastructure server (PKI), medical institutions, EHR and attacker. The detailed functions of these 6 components are elaborated in the following.

1) **EHR:** EHR contains the user's basic information and medical information. It is not only the carrier of user data in the scheme, but also the data object protected in our model.
2) **EHR User:** The provider of user's basic information in EHR is called EHR user who is the source of EHR data. All data in EHR is related to him. EHR users have the right to control their own data.
3) **Medical Institution**: Medical institutions provide not only authoritative medical data for EHR, but also necessary disease diagnosis and treatment support. They are medical data providers and data users in our model.
4) **Medical Cloud Server:** Medical cloud server is semi-trusted, where the medical data is encrypted and stored. Our proposed MedSecrecy encryption algorithm completes the encryption of medical data.
5) **Key Center PKI:** Key center PKI, which is an independent object in the model, releases related parameters of the communication key in the MedGreen communication authentication scheme. It supervises the key of the entity, and provides part of the identity authentication function.
6) **Attacker:** The attacker in the model has the ability to monitor messages and can obtain user data through man-in-the-middle attacks and password cracking.

B. MODEL PRINCIPLE

The architecture of system model can be divided into 3 layers. The bottom layer includes EHR users, medical IoT devices, attackers and medical institutions. The middle layer is the key infrastructure, while the top layer is the medical cloud server. The running process of the model is as follows:

1) Establishment of model communication. PKI publishes the basic parameters of the key, and transmits the private key to the EHR user, medical institution, or medical cloud server. EHR users, medical institutions and medical cloud servers employ MedGreen communication authentication scheme proposed in this paper to calculate communication keys and complete authentication of identity information. After the communication key is established, the communication information is encrypted and transmitted by the key. These information are mainly operation instructions with a small amount of data.
2) Transmission of medical data in the model. User's basic information in EHR and the information collected through IoT device are encrypted and uploaded to the medical cloud server through MedSecrecy algorithm. It is worth noting that the key required by the algorithm is set by user and the encrypted information is identified by the account number. Meanwhile, medical data of users in medical institutions is encrypted and uploaded to the cloud server by medical institution through MedSecrecy algorithm as well.
3) Storage of medical data in the model. After receiving the cipher text from EHR user and medical institution, medical cloud server matches them based on the account information, and finally combines the two parts into a complete EHR for data storage.

V. PROTECTION TECHNIQUE IN THE MODEL

This section begins with a description of the application technique in our model. First, we propose the MedGreen communication authentication scheme. Then, we present the MedSecrecy symmetric encryption algorithm for EHR. We also analyze the performance and security of both algorithms.

A. MEDGREEN COMMUNICATION AUTHENTICATION SCHEME

Our MedGreen communication authentication scheme includes MedGreen communication authentication algorithm, algorithm correctness and security analysis, communication resource overhead analysis, and communication establishment method.

TABLE 1. Symbolic representation of MedGreen communication authentication algorithm.

Symbol	Explanation
PKI	Public key infrastructure server
$A = \{a_0, a_1, a_2, \cdots, a_n\}$	A collection of random numbers belonging to $\mathbb{Z}_q^*$
$P=\{P_0=a_0Q,\ P_1=a_1Q,$ $P_2=a_2Q, \cdots, P_n=a_nQ\}$	Public parameter set calculated by random secrets
$f(x)=a_0+a_1x+a_2x^2$ $+\cdots+a_nx^n \in \mathbb{Z}_q[x]$	N-degree random polynomial with PKI's random numbers as coefficient
G_1	Additive group of order q
G_2	Multiplicative group of order q
H_1: $\{0,1\}^* \rightarrow G_1$	Mapping strings of arbitrary length into G_1
H_2: $\{0,1\}^* \rightarrow \mathbb{Z}_q^*$	Mapping strings of arbitrary length into $\mathbb{Z}_q^*$
e	Bilinear mapping
Q	G_1 generator
q	G_1, G_2 prime order
S_i	Private key of entity i
W_i	Public key of entity i
ID_i	The identity of entity i
ξ	$\mathbb{Z}_q^*$ random number used to calculate the public key of entity i
ψ_i	Public parameters of entity i
T	Random verification information
$Auth$	Identity and key authentication information
V	Random verification code
K	Communication key

1) MEDGREEN COMMUNICATION AUTHENTICATION ALGORITHM

The transmission of medical data is inseparable from a secure communication scheme. A low-overhead and balanced communication scheme will be more conducive to the transmission and interaction of medical information. Inspired by Al Hamid model [27] and ID-based Signcryption Scheme [22], we design the secure communication algorithm applied in the scheme.

For better elaboration, the algorithm symbols and their explanations are presented in Table 1.

Now, we put forward the MedGreen communication authentication algorithm based on elliptic curve and bilinear pairing and then apply it to the communication of the scheme. The pseudocode of MedGreen communication authentication algorithm is elaborated in Algorithm 3.

Algorithm 3 MedGreen Communication Authentication Algorithm

```
Input: Key base element.
Output: Communication key element.
if $ID$ is PKI then
    Take the set of random numbers $A = \{a_0, a_1, a_2, \cdots, a_n\}$
    Calculate the public parameter set $P = \{P_0 = a_0Q, P_1 = a_1Q, P_2 = a_2Q, \cdots, P_n = a_nQ\}$
    Determine the calculation polynomial $f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n \in \mathbb{Z}_q[x]$
    Determine the hash algorithm $H_1$: $\{0, 1\}^* \rightarrow G_1$, $H_2$: $\{0,1\}^* \rightarrow \mathbb{Z}_q^*$
    Calculate PKI's public key $W_{PKI} = H_1(ID_{PKI}^{\xi})$
    Calculate PKI's private key $S_{PKI} = f(W_{PKI}) = a_0 + a_1W_{PKI} + a_2W_{PKI}^2 + \cdots + a_nW_{PKI}^n \bmod q$
    Calculate PKI's public parameter $\psi_{PKI} = \sum_{i=0}^{n} W_{PKI}^i P_i = S_{PKI}Q$
    if Received receiver's public key $W_R$ then
        Calculate receiver's private key $S_R = f(W_R) = a_0 + a_1W_R + a_2W_R^2 + \cdots + a_nW_R^n \bmod q$
        Calculate receiver's public parameter $\psi_R = \sum_{i=0}^{n} W_R^i P_i = S_RQ$
        Send $(S_R, \psi_R, \psi_{PKI})$ to receiver through the private channel
        Release key related parameters $(G_1, G_2, e, H_1, H_2, Q, P, \psi_R, \psi_{PKI})$
    else {Received sender's public key $W_S$}
        Calculate receiver's private key $S_S = f(W_S) = a_0 + a_1W_S + a_2W_S^2 + \cdots + a_nW_S^n \bmod q$
        Send $(S_S, \psi_R, \psi_{PKI})$ to sender through the private channel
    end if
else if $ID$ is sender then
    Calculate the public key $W_S = H_1(ID_S^{\xi})$
    Send $W_S$ to PKI via private path
    if Received $Auth'$ then
        Authentication fails
    else {Received $(S_S, \psi_R, \psi_{PKI})$}
        Calculate $\psi_S = \sum_{i=0}^{n} W_S^i P_i = S_SQ$
        Calculate $K_{SR} = e(\psi_R, \psi_{PKI})^{S_S} = e(Q, Q)^{S_S S_R S_{PKI}}$
        Generate random number $r$
        Calculate $T = r \cdot \psi_R$
        Calculate $Auth = H_2(\psi_S || \psi_R || T || K_{SR})$
        Calculate $V = (r - Auth \cdot S_S)\ mod\ q$
        Send $(\psi_S, Auth, V)$ to receiver
    end if
else {$ID$ is receiver}
    Calculate the public key $W_R = H_1(ID_R^{\xi})$
    Send $W_R$ to PKI via private path
    if Received $(\psi_S, Auth, V)$ and $(S_R, \psi_R, \psi_{PKI})$ then
        Calculate $K_{RS} = e(\psi_S, \psi_{PKI})^{S_R} = e(Q, Q)^{S_S S_R S_{PKI}}$
        Calculate $T' = V \cdot \psi_R + Auth \cdot S_R \cdot \psi_S = r \cdot \psi_R$
        Calculate $Auth' = H_2(\psi_S || \psi_R || T' || K_{RS})$
        if $Auth = Auth'$ then
            Verification succeeded
        else
            Verification failed, send $Auth'$ to sender
        end if
    end if
end if
```

The algorithm first judges the identity of the user, and then determines the user's algorithm operation mode according to the identity. Note that we adopt the elliptic curve construction method [22], [27] and the HMAC technique which is based on SHA-1 [43]. Here, the key base elements and communication key elements include $(W_i, S_i, \psi_i, \psi_{PKI}, \xi, Auth, V, \ K)$, where i represents the entity (receiver or sender).

PKI runs the key initialization part, and the sender or receiver runs the key establishment and authentication part. The process of MedGreen communication authentication algorithm is as follows:

- The algorithm first judges the user's identity according to the ID. If the identity is confirmed as PKI, the calculation of key parameters and the release of related data are performed. PKI selects a set of $n+1$ random numbers A, then uses A to calculate the corresponding set P, and determines the private key to calculate the polynomial $f(x)$. Next, determine the hash algorithms H_1, H_2, confirm the identity ID_{PKI}, and calculate the public key W_{PKI} and public parameter ψ_{PKI} through calculation. If the public key information W_i of entity i is received, information processing will be performed, and the processing result $(S_i, \psi_R, \psi_{PKI})$ will be transmitted through the private channel. Finally, the information $(G_1, G_2, e, H_1, H_2, Q, P, \psi_R, \psi_{PKI})$ is published.
- After the identity of the entity is confirmed as the sender, key calculation and identity verification are performed. The sender first calculates the public key information W_S and sends it to PKI, and obtains $(S_S, \psi_R, \psi_{PKI})$ through the key data published by PKI. The sender then calculates the communication key K_{SR}, selects the random number r and obtains $(\psi_S, Auth, V)$, finally sends it to the receiver.
- After the identity of the entity is confirmed as the receiver, the key calculation and identity verification are also performed. The receiver first calculated the public key information W_R and sent to PKI. After receiving the messages $(\psi_S, Auth, V)$ and $(S_R, \psi_R, \psi_{PKI})$, the receiver calculates the key K_{RS}, and then calculates T' and $Auth'$. If the verification is successful, the key is correct and the identity authentication is passed.

2) ALGORITHM CORRECTNESS ANALYSIS

First, in the key calculation phase, both the sender and receiver obtain the key construction information through calculation, as shown in equation 2.

$$\begin{aligned}\psi_S &= \sum_{i=0}^{n} W_S^i P_i = S_S Q \\ \psi_R &= \sum_{i=0}^{n} W_R^i P_i = S_R Q \end{aligned} \tag{2}$$

Subsequently, both parties calculate the same pairing to obtain the key, as shown in equation 3.

$$\begin{aligned} K_{SR} &= e(\psi_R, \psi_{PKI})^{S_S} = e(Q, Q)^{S_S S_R S_{PKI}} \\ K_{RS} &= e(\psi_S, \psi_{PKI})^{S_R} = e(Q, Q)^{S_S S_R S_{PKI}} \end{aligned} \tag{3}$$

Thus, we get $K_{SR} = K_{RS}$.

Finally, when the receiver verifies the identity of the sender, the verification information is calculated as shown in equation 4.

$$\begin{aligned} T' &= V \cdot \psi_R + Auth \cdot S_R \cdot \psi_S \\ &= (r - Auth \cdot S_S) \cdot \psi_R + Auth \cdot S_R \cdot \psi_S \\ &= r \cdot \psi_R - Auth \cdot S_S \cdot \psi_R + Auth \cdot S_R \cdot \psi_S \\ &= r \cdot \psi_R - Auth \cdot S_S \cdot \psi_R + Auth \cdot S_R \cdot S_S \cdot Q \\ &= r \cdot \psi_R - Auth \cdot S_S \cdot \psi_R + Auth \cdot S_S \cdot \psi_R \\ &= r \cdot \psi_R \end{aligned} \tag{4}$$

Then, $T' = T$. This confirms the correctness of the authentication.

3) ALGORITHM SECURITY ANALYSIS

First, we randomly generate large prime numbers n, r, and ξ. The private key S_i is transmitted from PKI to entity i through a private channel. The private channel is a unique channel provided to the user exclusively by the medical cloud service provider. Compared with the traditional public network access, this channel is unique for each user in the network. Because it provides independent physical leased lines, leased line channels, and leased line gateways for each access user, it has extremely high security. Since the order n is random, $f(x)$ is also random. ξ is used to calculate the public key W_i and is also random. Therefore, the random generation mechanism makes the private key S_i and the public key W_i have a good security effect.

Subsequently, the authentication information is mainly generated by verifying random number r, and r is hidden by calculation, T is not transmitted in this phase. Meanwhile, the *Auth* code incorporates the generated key K which contains the public parameter information of the receiver and the sender and determines the uniqueness of the message code.

Here, if the attacker tries to obtain the two-party key K by using a man-in-the-middle attack. According to the algorithm, we know that all the information the attacker can obtain is $(\psi_S, Auth, V)$ and (ψ_R, ψ_{PKI}). Because these two parts are transmitted through the public path or published by PKI. According to the calculation formula of K, the attacker cannot crack the key without knowing the physical private key S_i. Therefore, the attacker cannot grasp the key K information.

Secondly, the attacker also needs to forge the authentication information *Auth* of the sender, and must first forge T and K. Since r is randomly generated by the sender, it is hidden in T, that is, $T = r \cdot \psi_S$. When the receiver verifies V, T' is obtained through a complex calculation, and then $Auth'$ is obtained through a hash operation, the final authentication is completed by comparing *Auth* and $Auth'$. T is not directly

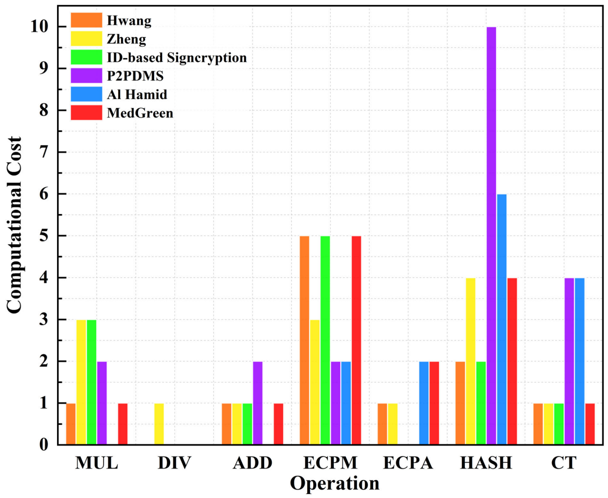


FIGURE 2. Comparison of communication resource overhead.

TABLE 2. Full name of communication resource overhead.

Abbreviation	Full name
MUL	Modular multiplication operation
DIV	Modular division operation
ADD	Modular addition operation
ECPM	Elliptic Curve point multiplication operation
ECPA	Elliptic Curve point addition operation
CT	Communication times

transmitted, but is calculated, so this ensures the irreplaceability of the number of messages. It is known from the above that K is obtained by bilinear pairing. If the attacker wants to completely decrypt the sender's identity information, he will face ECDLP. Therefore, the attacker cannot forge the identity. Thus, our algorithm can avoid man-in-the-middle attacks.

4) COMMUNICATION RESOURCE OVERHEAD ANALYSIS

The time complexity of MedGreen communication authentication algorithm is $O(n)$, which has good operating efficiency. The algorithm complexity is mainly related to the input data and response efficiency. We make statistics on the resource overhead of the sender and receiver. The resource overhead includes basic operations and communication times. That is, the algorithm performs 1 modular multiplication operation, 1 modular addition operation, 5 elliptic curve point multiplication operations, 2 elliptic curve point addition operations, 4 hash operations and 1 communication transmission.

We also perform statistics on other models. Fig. 2 depicts the comparison of communication resource overhead of some communication authentication models. For ease of description, Table 2 presents the full operation name in Fig. 2. As can be seen from the figure that Hwang's model [21] has a total cost of 11, Zheng's model [20] has a total cost of 14, ID-based Signcryption model [22] has a total cost of 12, P2PDMS model [30] has a total cost of 20, and Al Hamid's model [27] has a total cost of 14.

In this regard, our algorithm focuses on a comparative analysis of the number of communications and the size of the computational overhead. Generally, the consumption of communication resources, in addition to local computing operations, has a direct or indirect relationship with the transmission distance, the number of access devices, communication delays and the amount of data in transmission resources. This is more prominent in 5G wireless networks [46], [47].

Compared with the local calculation of the key, the overhead of key transmission in the intermediate medium often accounts for a large part. Therefore, reducing the number of communications during the key establishment process is also part of reducing the bilinear pairing overhead. In the process of establishing secure communication, we reduce the transmission of communication information to one. The size of the transmitted packet is $|G_1|$ element (ψ_S) + 160 bits (HMAC generation value *Auth*) + $|\mathbb{Z}_q^*|$ element(V). MedGreen communication authentication algorithm only needs the sender and receiver to transmit the key data once, and the amount of data is small. Under the premise of reducing the number of communications, the overall resource overhead is more energy efficient.

In the medical scenario, the sender and receiver correspond to the user and the cloud server, so the number of users' access to the device is much larger than that of the cloud server. When a large number of devices are connected at the same time, the traditional model with PKI as the main calculation device may cause data blocking, large computational overhead, network congestion and user waiting. For these reason, we hand over some simple calculations to users to balance the resource consumption of users and PKI, which relieves the pressure of PKI during busy hours. Thus, the model can have better robustness.

5) MEDICAL SECURE COMMUNICATION

In this section, we will apply MedGreen communication authentication algorithm to construct a secure key communication scheme. First of all, we make the necessary description of the secure private channel for key transmission. The private channel in the model is jointly built by the medical cloud server and the network operator. The operator first provides the physical line and the dedicated line gateway, then configures the private network on the basis of the physical line, and finally sets the network dedicated line channel. At the same time, the medical cloud server needs to configure relevant data interfaces.

The basic parameters for constructing communication in the scheme are provided by PKI. PKI selects basic parameters for establishing the communication and then publishes them. Fig. 3 shows the process of PKI communication information release. As can be seen from the figure that the specific operation steps of PKI are:

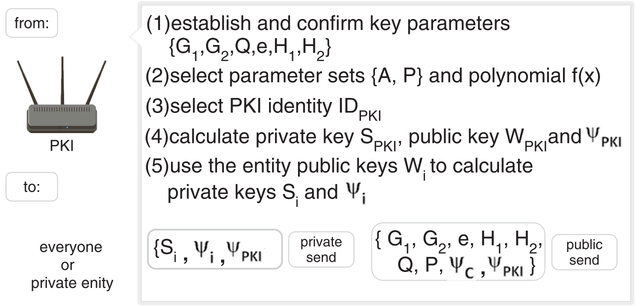

from:
PKI
to:
everyone
or
private enity
(1)establish and confirm key parameters
$\{G_1,G_2,Q,e,H_1,H_2\}$
(2)select parameter sets {A, P} and polynomial f(x)
(3)select PKI identity ID_{PKI}
(4)calculate private key S_{PKI}, public key $W_{PKI}$and Ψ_{PKI}
(5)use the entity public keys W_i to calculate
private keys S_i and Ψ_i
$\{S_i, \Psi_i, \Psi_{PKI}$
private send
$\{G_1, G_2, e, H_1, H_2, Q, P, \Psi_C, \Psi_{PKI}\}$
public send

FIGURE 3. PKI communication information release.

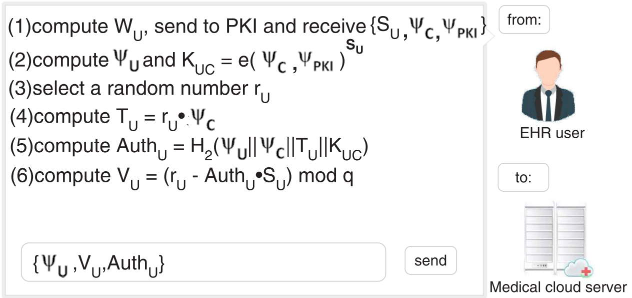

(1)compute W_U, send to PKI and receive $\{S_U,\Psi_C,\Psi_{PKI}\}$
(2)compute Ψ_U and $K_{UC} = e(\Psi_C, \Psi_{PKI})^{S_U}$
(3)select a random number r_U
(4)compute $T_U = r_U \bullet \Psi_C$
(5)compute $Auth_U = H_2(\Psi_U || \Psi_C || T_U || K_{UC})$
(6)compute $V_U = (r_U - Auth_U \bullet S_U) \bmod q$
$\{\Psi_U, V_U, Auth_U\}$
send
from:
EHR user
to:
Medical cloud server

FIGURE 4. EHR user communication information processing.

- Establish and confirm the key parameters $\{G_1, G_2, Q, e, H_1, H_2\}$.
- Confirm the polynomial function $f(x)$ and the parameter sets A, P.
- Choose the identity ID_{PKI}.
- Calculate private key S_{PKI}, public key W_{PKI}, and public parameters ψ_{PKI}.
- If receives entity public key W_i, then calculate entity private key S_i and public parameter ψ_i, where i represents user U or medical institution M.
- Transmit S_i, ψ_i, ψ_{PKI} to the entity through a secure path, and publish G_1, G_2, Q, P, e, H_1, H_2, ψ_C, ψ_{PKI}, where C is the medical cloud server.

After that, the EHR user will get the corresponding key parameters, then calculates the communication key K_{UC}, and simultaneously generates authentication information $Auth_U$ and V_U. In the end, user sends it to the medical cloud server. The specific operation steps of the EHR user in Fig. 4, which shows the workflow of EHR user communication information processing, are:

- Calculate the public key W_U and send it to PKI to get $\{S_U, \psi_C, \psi_{PKI}\}$.
- Obtain the public parameter ψ_U through calculation, and adopt the bilinear pairing to obtain the key K_{UC}.
- Select the random number $r_U \in \mathbb{Z}_q^*$.
- Employ calculation and hash algorithm to get T_U, $Auth_U$.
- Calculate the verification information V_U.
- Send ψ_U, V_U, $Auth_U$ to medical cloud server.

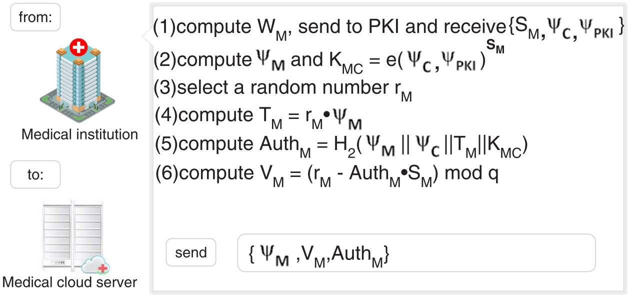

from:
Medical institution
to:
Medical cloud server
(1)compute W_M, send to PKI and receive $\{S_M,\Psi_C,\Psi_{PKI}\}$
(2)compute Ψ_M and $K_{MC} = e(\Psi_C, \Psi_{PKI})^{S_M}$
(3)select a random number r_M
(4)compute $T_M = r_M \bullet \Psi_M$
(5)compute $Auth_M = H_2(\Psi_M || \Psi_C || T_M || K_{MC})$
(6)compute $V_M = (r_M - Auth_M \bullet S_M) \bmod q$
send
$\{\Psi_M, V_M, Auth_M\}$

FIGURE 5. Medical institutions communication information processing.

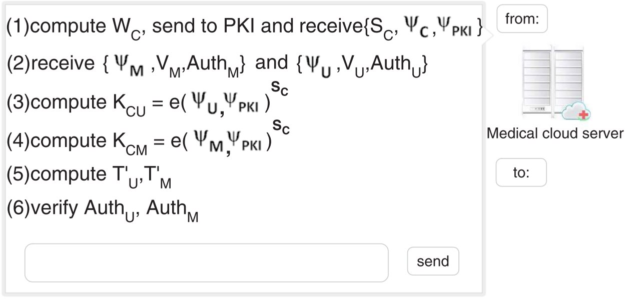

(1)compute W_C, send to PKI and receive $\{S_C, \Psi_C, \Psi_{PKI}\}$
(2)receive $\{\Psi_M, V_M, Auth_M\}$ and $\{\Psi_U, V_U, Auth_U\}$
(3)compute $K_{CU} = e(\Psi_U, \Psi_{PKI})^{S_C}$
(4)compute $K_{CM} = e(\Psi_M, \Psi_{PKI})^{S_C}$
(5)compute T'_U, T'_M
(6)verify $Auth_U$, $Auth_M$
send
from:
Medical cloud server
to:

FIGURE 6. Medical cloud server verification.

Similarly, medical institution will obtain the corresponding key parameters, then calculate the communication key K_{MC}, and generate public parameter verification information $Auth_M$ and V_M. Finally, medical institution sends it to the medical cloud server.

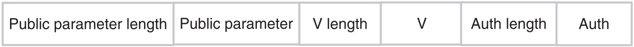
Fig. 5 shows the workflow of medical institutions communication information processing. As can be seen from the figure that the operating steps of medical institution are:

- Calculate the public key W_M and send it to PKI to get S_M, ψ_C, ψ_{PKI}.
- Obtain the public parameter ψ_M by calculation, and adopt the bilinear pairing to obtain the key K_{MC}.
- Select the random number $r_M \in \mathbb{Z}_q^*$.
- Employ calculation and hashing algorithms to get T_M, $Auth_M$.
- Calculate the verification information V_M.
- Send ψ_M, V_M, $Auth_M$ to medical cloud server.

After medical cloud server obtains the key information of EHR user and medical institution, it calculates the communication key and verifies these. If the verification is successful, the key is successfully established.

Fig. 6 shows the workflow of medical cloud server verification. It can be seen from Fig. 6 that the operation steps of the medical cloud server are:

- Calculate the public key W_C and send it to PKI to get S_C, ψ_C, ψ_{PKI}.
- Receive key information ψ_U, V_U, $Auth_U$ and ψ_M, V_M, $Auth_M$ from user and medical institution.
- Calculate keys K_{CU} and K_{CM}.
- Calculate T'_U and T'_M.
- Verify $Auth_U$ and $Auth_M$.



Public parameter length	Public parameter	V length	V	Auth length	Auth

FIGURE 7. Message transmission format.

So far, the establishment and verification of the communication key is complete. In order to improve the verification efficiency of the server, we adopt the message format shown in Fig. 7. We set the length of the message before the message. When the device detects the length of the message, it will be easy to determine the location of the data.

B. MEDSECRECY ENCRYPTION TECHNIQUE

In this section, we design the EHR encryption algorithm. Our encryption algorithm holds the characteristics of high data security requirements, high data repetition and large data volume in medical big data. Furthermore, it balances the speed, efficiency, and difficulty of implementing the algorithm. Finally, the algorithm is verified from the security, time complexity and randomness of the generated key stream.

1) MEDSECRECY ALGORITHM

RC4 algorithm is a secure symmetric encryption algorithm with the advantages of simple implementation, fast encryption. It has been widely used in wireless network encryption (WEP, WPA). Unfortunately, many security issues [48] about RC4 algorithm have arisen.

We have analyzed the current problems of RC4 algorithm and other variations of RC4. In order to provide stronger protection for EHR data, we redesign the algorithm with the original structure of RC4, named MedSecrecy algorithm, to further enhance the security and encryption efficiency.

MedSecrecy algorithm draws on the advantages of the RC4+ [31] algorithm, RC4-M2 [32] algorithm, and other RC4 algorithms to improve the security on this basis. Compared with RC4 and its improved algorithms, the main improvements of MedSecrecy algorithm are as follows:

1) In the initialization phase of the algorithm, a double S-box and double-key structure are used. To improve the KSA phase of the traditional double S-box RC4 algorithm, we propose a hybrid key scrambling strategy and also employ an IV with the same key length at the second layer.
2) We combine the Huffman compression algorithm [45] with the PRGA phase to reduce the amount of data output in cipher text.
3) In the PRGA phase, we improve the element output mode and complicate the output of key elements. Moreover, we improve the interaction mode of the double S-box, enhancing the randomness of the key stream and the security of encryption.

As shown in Algorithm 4, MedSecrecy algorithm consists of two sub-algorithms, named key scheduling algorithm (KSA) and Huffman compression and pseudo random generation algorithm (HCPRGA). KSA uses key_1 and key_2 to initialize the S-box, while HCPRGA encrypts the plain text data through the scrambled S-box and finally obtains the cipher text.

Algorithm 4 MedSecrecy Algorithm

```
Input: $Key_1$, $Key_2$, Plain text.
Output: Cipher text.
Call Algorithm 5 (Key Scheduling Algorithm)
Call Algorithm 6 (Huffman Compression and Pseudo Random Generation Algorithm)
```

Then, we divide each part of MedSecrecy algorithm which based on Huffman compression and RC4. Algorithm 5 elaborates the pseudocode of key scheduling algorithm (KSA).

In the initialization phase of KSA, the first thing to do is to fill the elements of double S-box. Then in the key scrambling phase, two keys are combined to scramble the elements in boxes. At the same time, j pointer is used to disrupt the initial order of the elements to generate a new element arrangement order. Next is the IV of KSA phase, using IV and keys to scramble the S-box. The i pointer first points to the middle of the S-box, then traverses down to 0 in order, and finally traverses the entire S-box. Also in the scrambling process, the two keys are XOR with j pointer together. In the zigzag disturbance of the third phase, the construction method of j is changed, and j is performed by using the exchange key scramble method.

In MedSecrecy algorithm, the Huffman compression and pseudo random generation algorithm (HCPRGA) consists of Huffman compression algorithm [45] and PRGA algorithm. The pseudocode of HCPRGA is shown in Algorithm 6. Huffman compression algorithm, which is in front of PRGA algorithm, can eliminate duplicate data in EHR and reduce the amount of data in the encryption and decryption process.

2) ALGORITHM SECURITY ANALYSIS

The current attacks faced by RC4 algorithm are weak key attack and key collision attack, IV attack, key recovery from state attack and state recovery attack, key recovery from key stream attack, brute force attack, biased bytes attack and distinguishers attack [48]. We will analyze these attacks of our MedSecrecy algorithm in this section.

a: WEAK KEY ATTACK AND KEY COLLISION ATTACK

Weak key attack refers to the traces left by the key when it disrupts S-box or generates the key stream. The attacker can further infer the key from these traces. Key collision attack is that two different keys generate nearly the same or similar key stream in the RC4 algorithm. In response to such attacks, the randomness, complexity, and length of the key have become the essential to strengthening the security of the key.

MedSecrecy algorithm's key is different from the traditional pseudo-random key generation methods. Instead, it combines the concept of random key generation proposed by Weerasinghe [37], and expands the output of the key from 128 bytes to 512 bytes. While ensuring the randomness

Algorithm 5 Key Scheduling Algorithm (KSA)

```
Input: $Key_1$, $Key_2$.
Output: Scrambled $S$-$box$.
    Initialization phase
for $i = 0$ to $n - 1$ do
    $S_1[i] = i$
    $S_2[i] = i$
end for
$j_1 = j_2 = 0$
    Layer 1: Basic scrambling phase
for $i = 0$ to $n - 1$ do
    $j_1 = (j_1 + S_1[i] + Key_1[i \bmod Key_1.length] + Key_2[i \bmod Key_2.length]) \bmod n$
    swap($S_1[i]$, $S_1[j_1]$)
    $j_2 = (j_2 + S_2[i] + Key_1[i \bmod Key_1.length] + Key_2[i \bmod Key_2.length]) \bmod n$
    swap($S_2[i]$, $S_2[j_2]$)
end for
    Layer 2: Scrambling with IV phase
for $i = n/2 - 1$ to $0$ do
    $j_1 = (j_1 + S_1[i])$ XOR $(Key_1[i \bmod Key_1.length] + Key_2[i \bmod Key_2.length] + IV[i])$
    $j_2 = (j_2 + S_2[i])$ XOR $(Key_1[i \bmod Key_1.length] + Key_2[i \bmod Key_2.length] + IV[i])$
    swap($S_1[i]$, $S_1[j_1]$)
    swap($S_2[i]$, $S_2[j_2]$)
end for
for $i = 0$ to $n - 1$ do
    $j_1 = (j_1 + S_1[i])$ XOR $(Key_1[i \bmod Key_1.length] + Key_2[i \bmod Key_2.length] + IV[i])$
    swap($S_1[i]$, $S_1[j_1]$)
    $j_2 = (j_2 + S_2[i])$ XOR $(Key_1[i \bmod Key_1.length] + Key_2[i \bmod Key_2.length] + IV[i])$
    swap($S_2[i]$, $S_2[j_2]$)
end for
    Layer 3: Zigzag scrambling phase
for $y = 0$ to $n - 1$ do
    if $y \equiv 0 \bmod 2$ then
        $i = y/2$
    else
        $i = n - (y + 1)/2$
    end if
    $j_1$=$(j_1 + S_1[i] + Key_2[i \bmod Key_2.length]) \bmod n$
    swap($S_1[i]$, $S_1[j_1]$)
    $j_2$=$(j_2 + S_2[i] + Key_1[i \bmod Key_1.length]) \bmod n$
    swap($S_2[i]$, $S_2[j_2]$)
end for
```

of the keys, the complexity of key generation is enhanced. Therefore, MedSecrecy algorithm can effectively avoid weak key attack and key collision attack.

b: IV ATTACK

IV attack is often caused by improper initialization vectors. We adopt the same KSA structure as the RC4+ algorithm, applying the *IV* in the middle layer. Meanwhile, the initialization state of the pointer is changed. *i* pointer moves from $2/n$ to the left and then from 0 to the right. Through the double-key scrambling operation, the movement of *j* pointer is affected by the double keys. Thus, it is difficult to infer the replacement state of the *S*-box. Finally, zigzag is used for further scrambling and the scrambling adopts a crossover method, providing more effective scrambling for the *IV* replacement phase. Therefore, it is believed that MedSecrecy algorithm can resist *IV* attack.

Algorithm 6 Huffman Compression and Pseudo Random Generation Algorithm (HCPRGA)

```
Input: Plain text, Scrambled $S$-$box$.
Output: Cipher text.
    Call Huffman compression algorithm [45]
    Pseudo random number generation algorithm
while Plain text do
    $j_1 = j_1 + S_1[i]$
    $j_2 = j_2 + S_2[i]$
    swap($S_1[i]$, $S_1[j_1]$)
    swap($S_2[i]$, $S_2[j_2]$)
    $t_1 = (S_1[i] + S_1[j_1]) \bmod n$
    $t_1' = (S_1[i \gg 3$ XOR $j_1 \ll 5] + S_1[i \ll 5$ XOR $j_1 \gg 3])$ XOR $0xAA$
    $t_1'' = (j_1 + S_1[j_1]) \bmod n$
    $t_2 = (S_2[i] + S_2[j_2]) \bmod n$
    $t_2' = (S_2[i \gg 3$ XOR $j_2 \ll 5] + S_2[i \ll 5$ XOR $j_2 \gg 3])$ XOR $0x55$
    $t_2'' = (j_2 + S_2[j_2]) \bmod n$
    $out_1 = ((S_1[S_2[t_2]] + S_1[S_2[t_2']])$ XOR $S_1[S_2[t_2'']]) \bmod n$
    $out_2 = ((S_2[S_1[t_1]] + S_2[S_1[t_1']])$ XOR $S_2[S_1[t_1'']]) \bmod n$
    $out_1$ XOR Cipher text
    $out_2$ XOR Cipher text
end while
```

c: KEY RECOVERY FROM STATE ATTACK AND STATE RECOVERY ATTACK

These two attack modes analyze the entire state of the *S*-box in RC4, that is, $N! \times N^2$ ($N!$ is the *N* bytes space of the *S*-box, and N^2 comes from all possible combinations of indexes *i* and *j*). The reversibility of the state with the key stream estimates the internal state of the *S*-box to infer the initial key and the state of the *S*-box transformation. However, MedSecrecy algorithm is not generated by $S_1[S_1[i] + S_1[j]]$ during the generation of the PRGA key stream. It is combined with the elements in the *S*-box through the pointer *t* step-by-step transformation, reaching $S_1[S_2[t_2]]$, which can make the key stream generated by PRGA more random. Also, it can make the key stream generation method more complicated, disrupt the internal state change of the *S*-box law. Therefore, MedSecrecy algorithm can effectively resist key recovery from state attack and state recovery attack.

d: KEY RECOVERY FROM KEY STREAM ATTACK

Key recovery from key stream attack also uses the reversibility of the PRGA phase, which continuously tries to encrypt the plain text and cipher text to recover the original key. In the PRGA phase, MedSecrecy algorithm further expands the randomness of the pseudo-random key stream and truly combines the double S-boxes so as to make the output state more difficult to predict. In addition, in the first phase of KSA, S-box is scrambled by double keys at the same time, which can ensure the uncertainty of elements' state in S-box. In the second phase, deviation elimination process is performed by using the IV. Finally, in the third phase, the zigzag replacement scrambles the elements of S-box again by key swapping to prevent the formation of recursive equations, which can better resist secret key recovery from key stream attack.

e: BRUTE FORCE ATTACK

Brute force attack is the most time-consuming and laborious attack method under any circumstances. However, it is often more effective for simple keys. With the development of distributed technology, the efficiency of brute force attacks will be further improved. MedSecrecy algorithm adopts a 512 bytes key length which is 2 times longer than that of RC4 [18], RC4(n, m) [33], RC4+ [31], RC4-M2 [32], RC4-2S [34]. Note that our algorithm employs the double S-box, double key scrambling and double key cross zigzag replacement methods. Compared with improved RC4 [36], modified RC4 [35], effective RC4 [37], RC4-M1, RC4-M3, the internal state of the S-box is more complicated. Therefore, it can effectively resist brute force attacks in both length and complexity.

f: BIASED BYTES ATTACK AND DISTINGUISHERS ATTACK

In stream cryptography algorithms, the reason for the deviation is the non-random part of the pseudo-random key stream. Attackers often start the attack from the non-random key flow, and the deviation is largely related to the KSA and PRGA phases. And formation is related to the correlation or deviation of the key bytes. MedSecrecy algorithm eliminates the bias by adding an IV of the same length as the key in KSA phase, and improves the randomness of the key stream by adding more pointer operations to interact with the double S-box in PRGA phase. Thus, it effectively avoids biased bytes attack and distinguishers attack. Therefore, the security of the algorithm is enhanced and protected.

3) TIME COMPLEXITY ANALYSIS

Next, we analyze the time complexity of the algorithm, as shown in Theorem 1.

Theorem 1: The time complexity of MedSecrecy algorithm is $O(nlogn)$.

Proof: All operations of the algorithm in the KSA phase are only for the S-box, and the size of elements in S-box are known. Therefore, the time complexity of the KSA phase is $O(1)$. In the Huffman compression phase, we adjust the implementation method of the algorithm. The time complexity of this phase is $O(nlogn)$. In the PRGA phase, since the key stream is XOR with the plain text, and the amount of time is related to the length of the plain text. Since the algorithm can generate two XOR units in one cycle, the time complexity of this phase is about $O(l/2)$, where l is the length of the plain text. Therefore, the time complexity of MedSecrecy algorithm is $O(nlogn)$.

VI. SIMULATION

The simulation experiments will compare the confidentiality of the encryption algorithm, the randomness of key stream, the encryption time, and the amount of encrypted data. The comparison targets are RC4 algorithms and AES algorithm which widely used in wireless network and medical data encryption.

The hardware facilities of the simulation experiments are Intel(R) Core(TM) i5-8250U processor, 8G memory, equipped with 64-bit Windows 10 operating system, the programming language is Python, and the experimental data set is MIMIC-III(https://mimic.physionet.org/). MIMIC-III is a publicly available data set developed by the Massachusetts Institute of Technology's Computational Physiology Laboratory. These data include not only patient vital signs, test results, medication status, but also nursing staff records, imaging reports, etc. Since its content is consistent with EHR data, it is the best data set for EHR research.

A. ALGORITHM CONFIDENTIALITY ANALYSIS

We first use Shannon's privacy principle defined by Weerasinghe [37] to verify the relationship between cipher text output size and algorithm confidentiality when the key length is constant, and the relationship between key length and algorithm confidentiality when the cipher text output is unchanged.

The maximum key length of the RC4 algorithm for a single S-box is 256 bytes, so it is the safest for RC4 key lengths of 256. This is the same as RC4(n, m), RC4+, RC4-2S, modified RC4, RC4-M2, FJ-RC4. The encryption key of improved RC4, effective RC4, RC4-M1, RC4-M3 and MedSecrecy can be extended to 512 bytes, so when the key length is 512, its security is the highest.

Fig. 8 describes the change in the security value after increasing the amount of cipher text output when the key length is 256 bytes. As can be seen, MedSecrecy algorithm keeps a good confidentiality when the amount of cipher text output is increasing, the overall performance is stable, and it has the best security value at 577KB, 865KB, 1153KB. It can also maintain the advantage level under other data volumes, which is due to the improvement of MedSecrecy algorithm in the key stream. On the basis of the traditional double

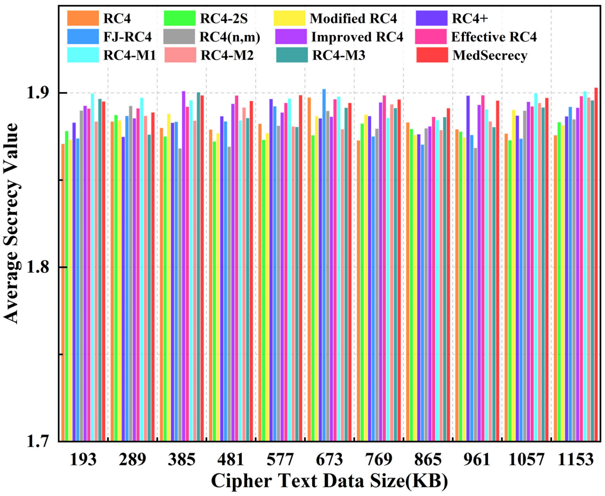


FIGURE 8. Average secrecy over cipher text data size.

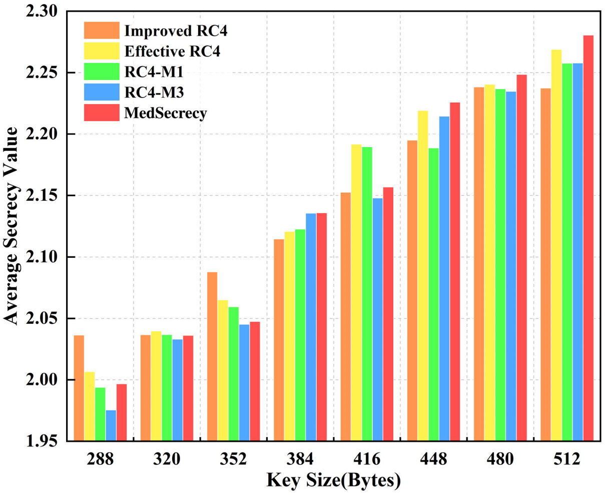


FIGURE 9. Average secrecy over key size.

S-box, we enhance the key scrambling method in the KSA phase and improve the key stream generation method in the PRGA phase, effectively strengthening the confidentiality of the algorithm.

So, when the amount of data is increasing, the average secret value of MedSecrecy algorithm is better than other RC4 algorithms in stability, therefore it has good algorithm security. So, when the amount of data is increasing, the average secret value of MedSecrecy algorithm is better than other RC4 algorithms in stability, therefore it has good algorithm security.

Fig. 9 illustrates the change in the average secret value when the length of the key is continuously increased with the amount of cipher text output being constant. Because RC4, RC4(n, m), RC4+, RC4-2S, modified RC4, RC4-M2, FJ-RC4 key length cannot continue to increase, we only choose improved RC4, effective RC4, RC4-M1, RC4-M3, MedSecrecy for comparison.

As can be seen in Fig. 9, with the increasing key length, the confidentiality of the five algorithms has continuously increased, and our MedSecrecy algorithm has grown significantly. When the key length reaches 480 bytes and 512 bytes, MedSecrecy algorithm reaches the best level of confidentiality. This is because MedSecrecy algorithm improves the key replacement strategy in the KSA phase by employing double key combination to scramble the S-box and improving the key usage in zigzag replacement. Thus, it further improves the key utilization rate and enhances the S-box replacement effect. In addition, 512 bytes is exactly the safest key length of each algorithm. Therefore, under the premise of increasing key length, MedSecrecy algorithm has better performance.

B. KEY STREAM RANDOMNESS ANALYSIS

The randomness test is the key to the success of the stream cipher algorithm design. Generally, the higher value of test result, the higher randomness of key stream. The higher security of the key algorithm, and the more it can withstand different degrees attack. The NIST randomness test is widely recognized as an internationally authoritative randomness test, it contains 15 tests as follows:

1) The frequency test is to test the proportion of "0" and "1" in the entire sequence. The number of 0 and 1 in the truly random sequence is the same.
2) The frequency test within a block is to divide the sequence into blocks of size M to calculate the proportion of "1" codes in the sub-blocks. Similarly, the sequence is random when the "1" code occupies half of the total number of codes.
3) The runs test is to test the size of the same sequence without interruption, and then compare the expected value of the ideal random sequence to determine whether it is random.
4) The test for the longest run of ones in a block is to test the longest uninterrupted "1"code run with a block length of M, and then determine whether it is the same as a random sequence.
5) The binary matrix rank test is to test the rank of the separated sub-matrix of the entire sequence, so as to determine the linear dependence between the fixed-length sub-chains.
6) The discrete fourier transform test is the peak height of the test sequence after stepwise fourier transform, so as to determine the degree of deviation between random signals.
7) The non-overlapping template matching test is a test to predict the number of occurrences of the target data string, and the purpose is to detect generators that generate too many aperiodic patterns.
8) The overlapping template matching test is to test the number of occurrences of known target modules. The process is generally the same as the non-overlapping module matching test. The only difference is the movement mode after the target module is found.
9) The maurer's "universal statistical" test is the information loss of the test sequence. The larger the loss, the worse the randomness.

10) The linear complexity test is to test the length of the linear feedback shift register to determine the linear complexity of the sequence.
11) The serial test is to check the frequency of all possible M-bits of the entire sequence. The sequence of the uniformity distribution is random.
12) The approximate entropy test is to test the frequency of the size of all possible overlapping M-bit sequences in the entire sequence, and then compare the randomness with the overlapping sub-blocks in a random case.
13) The cumulative sums test is the maximum deviation of the random walk of the test sequence, and then compared with the expected cumulative sum to obtain randomness.
14) The random excursions test is to test the number of loops with K nodes in the accumulation and random walk, to determine the number of special nodes in a loop state and the degree of deviation from the random state.
15) The random excursions variant test is to test the multiple states experienced during the random walk to determine the degree of deviation from the random state.

The test result is expressed by P-value. 0.01 is considered to be the lowest threshold for randomness. Under this condition, the higher the value, the better the randomness.

Therefore, we use the latest version of the NIST random test suite (https://csrc.nist.gov/projects/random-bit-generation/documentation-and-software) to test the key streams generated by each algorithm, and statistics their P-value, test key stream size is set at 6000000 bits, and the result is shown in Fig. 10.

As can be seen from the Fig. 10, all algorithms can pass 15 tests. MedSecrecy algorithm shows good randomness in frequency test, frequency test within a block, test for the longest run of ones in a block, overlapping template matching test, maurer's "universal statistical" test, approximate entropy test and cumulative sums test. This is due to the unique KSA algorithm of MedSecrecy, which improves the scheme of key pair S-box, IV and zigzag replacement, and improves the double S-box interaction of PRGA algorithm, making the output key stream more random.

At the same time, MedSecrecy algorithm is at a medium level in runs test, non-overlapping template matching test, serial test, random excursions test, and random excursions variant test.

Finally, MedSecrecy algorithm does not perform as well as other tests in the binary matrix rank test, discrete fourier transform test, and linear complexity test, but its P-value is still much greater than 0.01 and still has strong randomness. Therefore, it fully meets the requirements of the encryption algorithm for the randomness of the key stream.

In summary, MedSecrecy algorithm has the best key stream stability while passing all the tests. Compared with other algorithms, it exhibits better comprehensive performance in key stream randomness.

C. ALGORITHMS PERFORMANCE AND ENERGY CONSUMPTION ANALYSIS

Medical information is the largest amount of data in all attributes of EHR. It contains medical imaging information, doctor's orders, and other medical attributes. Both the image and the text have a high repetition rate, that is why the algorithm combines Huffman compression. Huffman compression shows excellent compression efficiency in both image and text compression. Therefore, we first compare the RC4 algorithms with a data volume of 128KB to 768KB, and then compare the AES-128 algorithm.

The test result of the amount of cipher text data is shown in Fig. 11. In order to highlight the change in the amount of encrypted data, we set the amount of encrypted data to bytes.

In Fig. 11, RC4 algorithms have no obvious difference in the amount of cipher text data, but MedSecrecy algorithm significantly shows better results. The algorithm reduces the amount of cipher text data, which has obvious advantages over other algorithms in this respect. This is because MedSecrecy algorithm integrates Huffman lossless compression algorithm, which has a good compression efficiency for medical data. Therefore, MedSecrecy algorithm can output less cipher text when the amount of plain text data is the same.

Fig. 12 is the comparison of the amount of cipher text data between AES-128 algorithm and MedSecrecy algorithm. Thanks to our optimized use of the Huffman compression algorithm, MedSecrecy algorithm is in an advantageous position compared with the AES-128 algorithm, and the gap will further increase as the amount of data increases.

Fig. 13 shows the encryption time comparison of each algorithm under different data volumes. MedSecrecy algorithm has better encryption efficiency when the data size is small. When the amount of data is 320KB to 640KB, the increase in encryption time is larger than other algorithms. This is because MedSecrecy algorithm includes two processes including compressed data and data encryption. Moreover, the continuous increasing amount of plain text data increases the time that the algorithm uses to compress data. But compared to other RC4 improved algorithms with the same amount of data, the difference is still in milliseconds. The time after 640KB tends to be stable and still has good encryption efficiency.

Similarly, we compare the encryption time of AES-128 algorithm with MedSecrecy algorithm about the increasing amount of data. It can be seen from Fig. 14 that MedSecrecy algorithm has obvious advantages over AES-128 algorithm in terms of encryption time, and has excellent encryption efficiency.

Finally, we analyze the energy efficiency of MedSecrecy algorithm. Firstly, in terms of the amount of cipher text data, MedSecrecy algorithm has obvious advantages over other algorithms, so it saves energy consumption for data transmission and storage. Secondly, in terms of encryption time, MedSecrecy algorithm has a faster encryption speed than AES algorithm. Compared with RC4 algorithms, although MedSecrecy algorithm does not perform well after the amount of

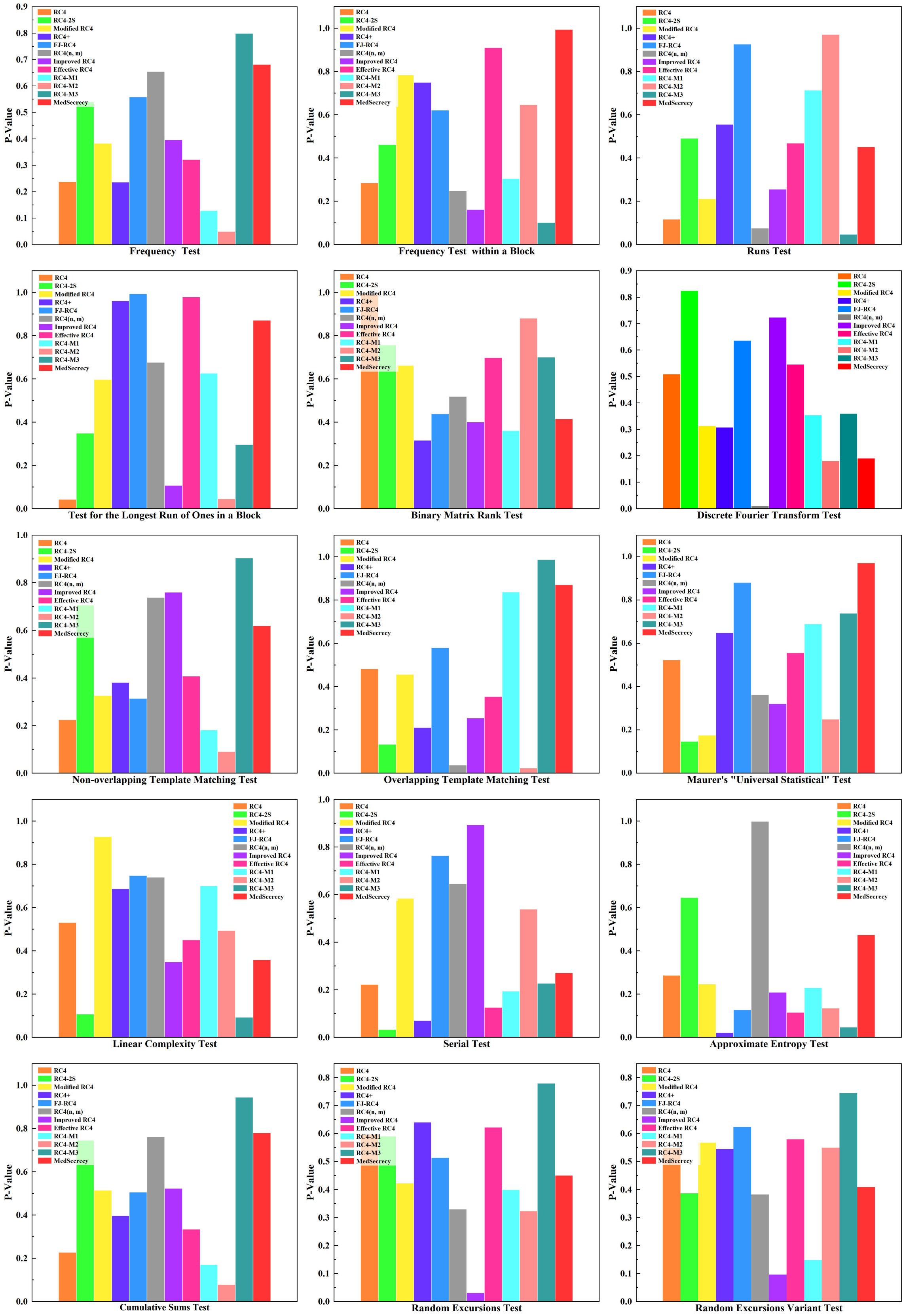


FIGURE 10. Random value comparison of algorithms.

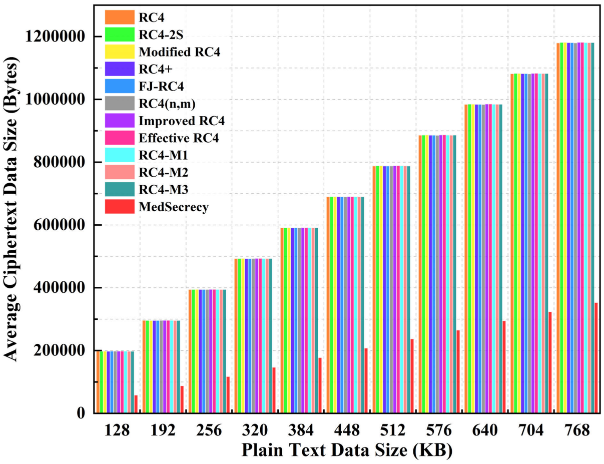


FIGURE 11. Comparison of cipher text data size of RC4 algorithms.

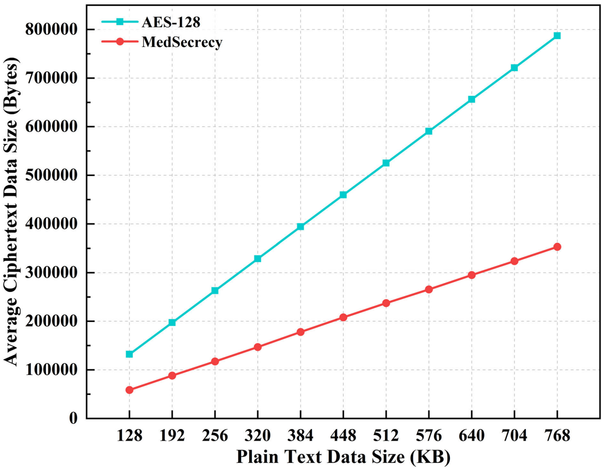


FIGURE 12. Comparison of cipher text data size between MedSecrecy and AES-128.

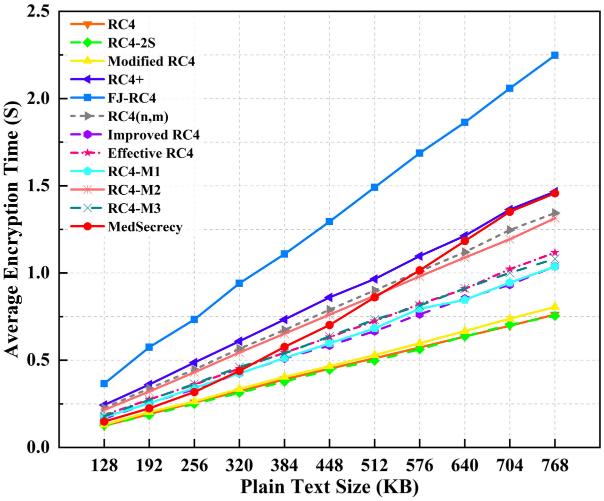


FIGURE 13. Comparison of encryption time of RC4 algorithms.

plain text data increases to 512KB, their gap is still in the millisecond range. Morover, under the premise that the amount of cipher text data is greatly reduced and the encryption security is improved, MedSecrecy algorithm is still generally energy-efficient in terms of data transmission and storage. Therefore, we believe that MedSecrecy algorithm has better energy performance.

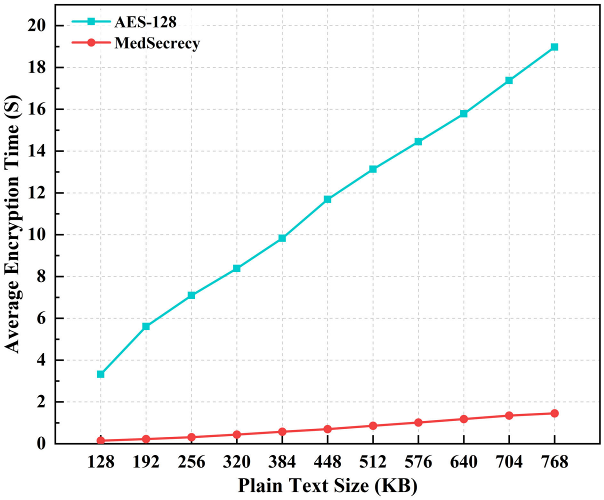


FIGURE 14. Comparison of encryption time between MedSecrecy and AES-128.

VII. CONCLUSION

In this paper, we focus on the security of communication and storage model for EHR. Firstly, a secure energy-saving communication and encrypted storage model based on MedGreen communication authentication scheme and MedSecrecy algorithm is proposed. The model balances communication overhead and improves data security. Secondly, in terms of model communication, we propose a MedGreen communication authentication scheme based on the elliptic curve and bilinear pairing theory. The scheme combines the two stages of key calculation and identity authentication, improves the key authentication method. Meanwhile, the scheme solves the computational load of PKI during busy hours by allocating key calculations to users, and alleviates data congestion and long waiting times. Finally, in the encryption phase of the model, a MedSecrecy algorithm based on Huffman compression and RC4 algorithms is proposed. On the basis of maintaining the original RC4 basic structure, we improved the key scrambling scheme in the KSA phase and the dual S-box key stream generation scheme in the PRGA phase. These enhance the confidentiality of the algorithm and the stability of the random key stream generation. Moreover, in view of the characteristics of large amount of medical data and high repetition rate, we combine Huffman compression algorithm in MedSecrecy to reduce the cipher text size while maintaining the encryption efficiency. Extensive simulations validate the effectiveness of our system on performance and security.

In the future, we will add an anonymous solution for data sharing in the model to ensure that the process of information sharing will not allow the attacker to infer the identity of the user or obtain more possible user information.

REFERENCES

[1] H. Ullah, N. Gopalakrishnan Nair, A. Moore, C. Nugent, P. Muschamp, and M. Cuevas, "5G communication: An overview of Vehicle-to-everything, drones, and healthcare use-cases," *IEEE Access*, vol. 7, pp. 37251–37268, 2019.

[2] J.-S. Pan, C.-Y. Lee, A. Sghaier, M. Zeghid, and J. Xie, "Novel systolization of subquadratic space complexity multipliers based on Toeplitz matrix–vector product approach," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 7, pp. 1614–1622, Jul. 2019.

[3] X. Bai, Z. Wang, L. Sheng, and Z. Wang, "Reliable data fusion of hierarchical wireless sensor networks with asynchronous measurement for greenhouse monitoring," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 3, pp. 1036–1046, May 2019.

[4] R. Li, C. Sturtivant, J. Yu, and X. Cheng, "A novel secure and efficient data aggregation scheme for IoT," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1551–1560, Apr. 2019.

[5] X. Bai, Z. Wang, L. Zou, and C. Cheng, "Target tracking for wireless localization systems with degraded measurements and quantization effects," *IEEE Trans. Ind. Electron.*, vol. 65, no. 12, pp. 9687–9697, Dec. 2018.

[6] L. Ni, J. Zhang, C. Jiang, C. Yan, and K. Yu, "Resource allocation strategy in fog computing based on priced timed Petri nets," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1216–1228, Oct. 2017.

[7] Z. Liu, Y. Cao, L. Cui, J. Song, and G. Zhao, "A benchmark database and baseline evaluation for fall detection based on wearable sensors for the Internet of medical things platform," *IEEE Access*, vol. 6, pp. 51286–51296, 2018.

[8] J. Zhang, X. Wang, Y. Yuan, and L. Ni, "RcDT: Privacy preservation based on R-constrained dummy trajectory in mobile social networks," *IEEE Access*, vol. 7, pp. 90476–90486, 2019.

[9] L. Ni, F. Tian, Q. Ni, Y. Yan, and J. Zhang, "An anonymous entropy-based location privacy protection scheme in mobile social networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, pp. 93–111, Apr. 2019.

[10] S. Sharaf and N. F. Shilbayeh, "A secure G-cloud-based framework for government healthcare services," *IEEE Access*, vol. 7, pp. 37876–37882, 2019.

[11] K. Riad, R. Hamza, and H. Yan, "Sensitive and energetic IoT access control for managing cloud electronic health records," *IEEE Access*, vol. 7, pp. 86384–86393, 2019.

[12] Y. Wang, A. Zhang, P. Zhang, and H. Wang, "Cloud-assisted EHR sharing with security and privacy preservation via consortium blockchain," *IEEE Access*, vol. 7, pp. 136704–136719, 2019.

[13] X. Liu, Z. Wang, C. Jin, F. Li, and G. Li, "A blockchain-based medical data sharing and protection scheme," *IEEE Access*, vol. 7, pp. 118943–118953, 2019.

[14] L. Ni, C. Li, X. Wang, H. Jiang, and J. Yu, "DP-MCDBSCAN: Differential privacy preserving multi-core DBSCAN clustering for network user data," *IEEE Access*, vol. 6, pp. 21053–21063, 2018.

[15] K. Seol, Y.-G. Kim, E. Lee, Y.-D. Seo, and D.-K. Baik, "Privacy-preserving attribute-based access control model for XML-based electronic health record system," *IEEE Access*, vol. 6, pp. 9114–9128, 2018.

[16] H. Zhang, J. Yu, C. Tian, P. Zhao, G. Xu, and J. Lin, "Cloud storage for electronic health records based on secret sharing with verifiable reconstruction outsourcing," *IEEE Access*, vol. 6, pp. 40713–40722, 2018.

[17] X. Zhang, H. Chen, K. Lin, Z. Wang, J. Yu, and L. Shi, "RMTS: A robust clock synchronization scheme for wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 135, pp. 1–10, Jun. 2019.

[18] R. Rivest, *The RC4 Encryption Algorithm*, (Proprietary). Bedford, MA, USA: RSA Data Security, 1992.

[19] A. Chiesa, L. Chua, and M. Weidner, "On cycles of pairing-friendly elliptic curves," *SIAM J. Appl. Algebra Geometry*, vol. 3, no. 2, pp. 175–192, Apr. 2019.

[20] Y. Zheng and H. Imai, "How to construct efficient signcryption schemes on elliptic curves," *Inf. Process. Lett.*, vol. 68, no. 5, pp. 227–233, Dec. 1998.

[21] R.-J. Hwang, C.-H. Lai, and F.-F. Su, "An efficient signcryption scheme with forward secrecy based on elliptic curve," *Appl. Math. Comput.*, vol. 167, no. 2, pp. 870–881, Aug. 2005.

[22] B. Nayak, "Signcryption schemes based on elliptic curve cryptography," Ph.D. dissertation, Dept. Comput. Sci. Eng., Nat. Inst. Technol. Rourkela, Rourkela, India, 2014.

[23] S.-D. Bao, M. Chen, and G.-Z. Yang, "A method of signal scrambling to secure data storage for healthcare applications," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 6, pp. 1487–1494, Nov. 2017.

[24] A. Kale, "Combining data owner-side and cloud-side access control for encrypted cloud storage," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 7, no. 6, pp. 629–631, Jun. 2019.

[25] C. Hu, W. Li, X. Cheng, J. Yu, S. Wang, and R. Bie, "A secure and verifiable access control scheme for big data storage in clouds," *IEEE Trans. Big Data*, vol. 4, no. 3, pp. 341–355, Sep. 2018.

[26] J. Zhang, Y. Yuan, X. Wang, L. Ni, J. Yu, and M. Zhang, "RPAR: Location privacy preserving via repartitioning anonymous region in mobile social network," *Secur. Commun. Netw.*, vol. 2018, pp. 1–10, Nov. 2018.

[27] H. A. Al Hamid, S. M. M. Rahman, M. S. Hossain, A. Almogren, and A. Alamri, "A security model for preserving the privacy of medical big data in a healthcare cloud using a fog computing facility with pairing-based cryptography," *IEEE Access*, vol. 5, pp. 22313–22328, 2017.

[28] S. Sciancalepore, G. Piro, G. Boggia, and G. Bianchi, "Public key authentication and key agreement in IoT devices with minimal airtime consumption," *IEEE Embedded Syst. Lett.*, vol. 9, no. 1, pp. 1–4, Mar. 2017.

[29] J. Cai, X. Huang, J. Zhang, J. Zhao, Y. Lei, D. Liu, and X. Ma, "A handshake protocol with unbalanced cost for wireless updating," *IEEE Access*, vol. 6, pp. 18570–18581, 2018.

[30] S. M. M. Rahman, M. Masud, A. N. M. Noman, A. Alamri, and M. M. Hassan, "Towards secure data exchange in Peer-to-Peer data management systems," *Appl. Math. Inf. Sci.*, vol. 8, no. 6, pp. 2775–2787, Nov. 2014.

[31] S. Maitra and G. Paul, "Analysis of RC4 and proposal of additional layers for better security margin," in *Proc. Inter. Conf. Cryptol. India*. Berlin, Germany: Springer, 2008, pp. 27–39.

[32] P. Jindal and B. Singh, "Optimization of the security-performance tradeoff in RC4 encryption algorithm," *Wireless Pers. Commun.*, vol. 92, no. 3, pp. 1221–1250, Aug. 2016.

[33] M. A. Orumiehchiha, J. Pieprzyk, E. Shakour, and R. Steinfeld, "Cryptanalysis of RC4(n, m) stream cipher," in *Proc. 6th Inter. Conf. Secur. Inf. Netw.*, Nov. 2013, pp. 165–172.

[34] M. M. Hammood, K. Yoshigoe, and A. M. Sagheer, "RC4-2S: RC4 stream cipher with two state tables," in *Information Technology Convergence*. Dordrecht, The Netherlands: Springer, 2013, pp. 13–20.

[35] T. D. B. Weerasinghe, "Analysis of a modified RC4 algorithm," *Int. J. Comput. Appl.*, vol. 51, no. 22, pp. 12–16, Aug. 2012.

[36] J. Xie and X. Pan, "An improved RC4 stream cipher," in *Proc. Int. Conf. Comput. Appl. Syst. Modeling (ICCASM)*, Oct. 2010, pp. 156–159.

[37] T. D. B. Weerasinghe, "An effective RC4 stream cipher," in *Proc. IEEE 8th Int. Conf. Ind. Inf. Syst.*, Dec. 2013, pp. 69–74.

[38] M. K. Chande, C.-C. Lee, and C.-T. Li, "Cryptanalysis and improvement of a ECDLP based proxy blind signature scheme," *J. Discrete Math. Sci. Cryptogr.*, vol. 21, no. 1, pp. 23–34, Mar. 2018.

[39] M. Kumar and P. Gupta, "An efficient and authentication signcryption scheme based on elliptic curves," *Malaysian J. Ind. Appl. Math.*, vol. 35, no. 1, pp. 1–11, Apr. 2019.

[40] C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval, "Elliptic curve lightweight cryptography: A survey," *IEEE Access*, vol. 6, pp. 72514–72550, 2018.

[41] C. Dobraunig, M. Eichlseder, and F. Mendel, "Analysis of SHA-512/224 and SHA-512/256," in *Proc. Inter. Conf. Theory Appl. Cryptol. Inf. Secur.* Berlin, Germany: Springer, 2015, pp. 612–630.

[42] S. Bayat-Sarmadi, M. Mozaffari-Kermani, and A. Reyhani-Masoleh, "Efficient and concurrent reliable realization of the secure cryptographic SHA-3 algorithm," *IEEE Trans. Comput.-Aided Design Integr.*, vol. 33, no. 7, pp. 1105–1109, Jul. 2014.

[43] D. Eastlake and P. Jones, *US Secure Hash Algorithm 1 (SHA1)*, document Internet Society RFC 3174, Reston, VA, USA, 2001.

[44] H. Krawczyk, R. Canetti, and M. Bellare, *HMAC: Keyed-Hashing for Message Authentication*. document Internet Society RFC 2104, Reston, VA, USA, 1997.

[45] G. S. Sandeep, B. S. S. Kumar, and D. J. Deepak, "An efficient lossless compression using double huffman minimum variance encoding technique," in *Proc. Int. Conf. Appl. Theor. Comput. Commun. Technol. (iCATccT)*, Oct. 2015, pp. 534–537.

[46] A. Abrol and R. K. Jha, "Power optimization in 5G networks: A step towards GrEEn communication," *IEEE Access*, vol. 4, pp. 1355–1374, 2016.

[47] S. Rizvi, A. Aziz, M. T. Jilani, N. Armi, G. Muhammad, and S. H. Butt, "An investigation of energy efficiency in 5G wireless networks," in *Proc. Int. Conf. Circuits, Syst. Simul. (ICCSS)*, Jul. 2017, pp. 142–145.

[48] P. Jindal and B. Singh, "RC4 encryption—A literature survey," *Procedia Comput. Sci.*, vol. 46, pp. 697–705, Jan. 2015.

JINQUAN ZHANG received the Ph.D. degree in computer science and technology from Tongji University, Shanghai, China, in 2007. He is currently an Associate Professor with the College of Computer Science and Engineering, Shandong University of Science and Technology. He is also with the Key Laboratory of the Ministry of Education for Embedded System and Service Computing, Tongji University. His current areas of research are cloud computing, Petri net, privacy preservation, machine learning, and parallel and distributed processing. He is also a Senior Member of the China Computer Federation (CCF). He is also the Committee Member of the Professional Committee of Network Information Service of China Automation Federation.

HAORAN LIU is currently pursuing the M.S. degree with the College of Computer Science and Engineering, Shandong University of Science and Technology. His main research interests include cloud computing, big data analysis, privacy preservation, and machine learning.

LINA NI received the Ph.D. degree in computer software and theory from Tongji University, Shanghai, China, in 2009. She is currently an Associate Professor with the College of Computer Science and Engineering, Shandong University of Science and Technology. She is also with the Key Laboratory of the Ministry of Education for Embedded System and Service Computing, Tongji University. Her current areas of research are cloud computing, Petri net, distributed algorithms, privacy preservation, machine learning, and intelligent computing. She is a member of the ACM and a Senior Member of the China Computer Federation (CCF). She is also the Committee Member of the Professional Committee of Network Information Service of China Automation Federation.