Journal of Computers Vol. 31 No. 4, 2020, pp. 65-76
doi:10.3966/199115992020083104006

Research on Security of the Authorization Chain Smart Terminal Based on Polymorphic Key Exchange Protocol

Yi-Feng Yin[1], Qing Zhang[1]*, Yong Gan[2], Ting-Jun Zhang[1]

[1] School of Computer and Communication Engineering, Zhengzhou University of Light Industry,
Zhengzhou 450000, China
yinyifeng@zzuli.edu.cn, {zhangqing1608, zhangtingjun1994}@163.com

[2] Zhengzhou Institute of Technology, Zhengzhou 450044, China
ganyong@zzuli.edu.cn

Received 26 December 2018; Revised 10 May 2019; Accepted 30 May 2019

Abstract. Internet of Things (IoT) connects different types of sensors and intelligent devices to collect a range of data under the globalization economic integration background. Hence, it is very crucial to prevent data leakage and enhance information security for smart terminals, especially in product authorization chains. To solve data communication security problems between smart devices and servers, this paper proposes the polymorphic key exchange protocol for the authorization chain smart terminal, which ensures data transmission security between the two parties. This paper provides a crucial model definition of the product authorizing process in product authorization chains, and designs an improved PRNG (Pseudo Random Number Generator) that is constructed by random group to accomplish the polymorphic key exchange protocol. The security analysis shows that the polymorphic key exchange protocol is secure to some attacks. Finally, we test and verify that the improved PRNG for the polymorphic key exchange protocol satisfies the strict avalanche criterion by extensive experiment, and compare with the avalanche effect of Data Encryption Standard (DES) and Advanced Encryption Standard (AES). The experimental data shows the improved PRNG has same input bit with DES, but its anti-differential attack performance is in common with 128-bit AES, which demonstrates it is secure to brute-force attack.

Keywords: authorization chain, polymorphic key exchange protocol, pseudo random number generator, smart terminal

1 Introduction

Due to the advances of mobile intelligent devices, wireless communication, and sensor network technologies, Internet of Things (IoT) has emerged as a pervasive global network infrastructure in the information era. It enables more and more networked things or smart objects to be involved, such as physical sensors, smart terminals and so on [1]. That leads to an explosion growth of data in intelligent terminals. The U.S. International Data Corporation (IDC) published that the digital universe will grow 40% a year during the next decade [2-3]. Data security becomes particularly important. Owing to the large improvement in smart devices' portability, interaction and computing performance, they are widely used in our daily life, such as social relationships, financial services, electronic messages, product sales and even business work [4-5]. In the process of the whole product authorization chain, users hope futher to use intelligent devices to purchase products and obtain product information at any time and place. However, hand-held terminals are often used in railway stations, taxis, malls, airplanes—where they are most vulnerable to be stolen, lost or downloaded data by unauthorized persons frequently [6]. Therefore, data security is particularly essential in the process of data transmission in the authorization chain smart terminal [7].

* Corresponding Author

The authorization chain smart device system enables users to complete the man-machine interface steps throughout building virtual environment by using Internet technology. In this smart system, the data communication security refers to the data storage and transfer between control platform with smart devices [8]. In order to the data communication security in the authorization chain smart terminal, the establishment of the shared session keys between two parties is quite necessary in the public channel. The key exchange algorithm based on public key digital certificates takes up more computational resources of both sides to build the shared session key. Traditional key exchange protocols are vulnerable to obstructive and man-in-the-middle attacks, for example, such as Diffie-Hellman key exchange algorithm. They cannot meet the growing data communication security needs for users in smart system. How to provide data security protection and strengthen information security, the safer key exchange protocol needs to be put forward to address security issues.

In this paper, the main contributions are summarized as follows:

(1) An important model of the product authorization chain is given, and the smart terminal product authorizing process is described.

(2) Using the polymorphic cipher and random group, PRNG is improved.

(3) The polymorphic key exchange protocol based on the improved PRNG is proposed. The security analysis indicates that the polymorphic key exchange protocol is secure. We experiment the avalanche effect [9] of the improved PRNG, and then contrast with DES [10] and AES [11] in terms of the avalanche effect. The results testify that the improved PRNG is random and hard to be cracked.

Compared with the traditional key exchange protocol, the smart system based on the polymorphic cipher have following characteristics:

(1) Randomness [12]: Making enough and random session keys makes sure that smart devices can communicate with each other secretly.

(2) Unreadability: In the polymorphic encryption [13] system, its security depends on the unreadability of self-compiling system. When the key generator is attacked, the self-compiling system is affected by aggression variables. Then encryption algorithm reunites aggression variables to generate new random sequences. So attackers can’t get correct keys.

(3) Anti-differential cryptanalysis attack [14]: In order to strengthen the iterative process, we set the strong one-way function group to be augmented with the random group. After that, the random group combines with polymorphic cipher to improve PRNG, which is the basis of the polymorphic key exchange protocol. The polymorphic cipher has the self-compiling characteristics. If attackers want to break keys, they need use more variables. And that will make the breaking process complicated. So the polymorphic key exchange protocol can be against differential cryptanalysis attacks and brute force attacks [15]. The user’s data security can be guaranteed by using the polymorphic encryption technology.

The rest of this paper is organized as follows: In Section 2, we introduced the model of the smart terminal system, the process of the traditional key exchange protocol and the advantage of the polymorphic cipher. The model establishment of the authorization chain, the design of the random group operation, the improvement of PRNG, and the design of the polymorphic key exchange protocol is introduced in Section 3. The security analysis of the polymorphic key exchange protocol is shown in Section 4. The analysis and comparison of the strict avalanche criterion and the experimental results are in Section 5. Finally, the conclusion is given in Section 6.

2 Related Work

2.1 Model of the Smart Terminal System

This smart terminal system has three parts: client programs, intelligent control platform and smart terminals. It’s shown in Fig. 1.

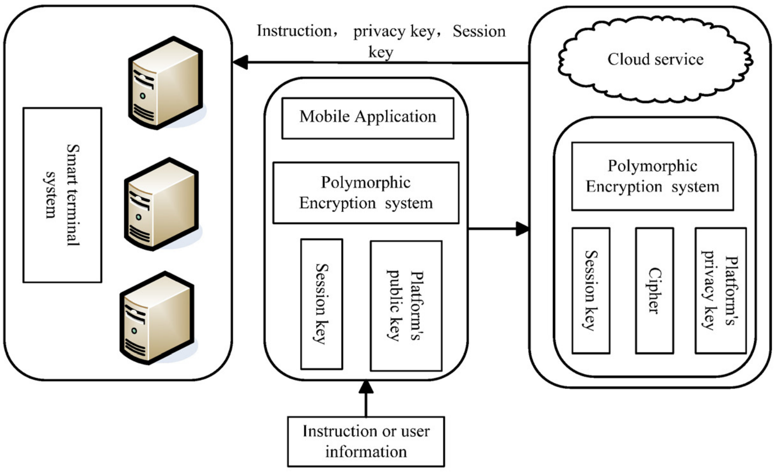

Instruction， privacy key，Session key
Smart terminal system
Mobile Application
Polymorphic Encryption system
Session key
Platform's public key
Cloud service
Polymorphic Encryption system
Session key
Cipher
Platform's privacy key
Instruction or user information

Fig. 1. The system of communication between smart terminals and cloud

The core of the system is the session key exchange protocol between smart terminals and the intelligent platform. In the course of generating session keys, the modular exponentiation has high complexity [16]. The intelligent platform is provided to solve these problems that mobile's insufficient internal memory and poor computational capability. The operation instructions of the equipment or the users' information are sent to smart terminal devices and service. When users want to operate the device by the internet, these orders will be sent to the smart terminals after being encrypted by shared session keys.

2.2 Process of the Traditional Key Exchange Protocol

Step 1. The client programs receive the digital certification from the intelligent control platform. Then they verify the digital certification at the same time.

Step 2. The platform makes a request for building the session key after the digital certification is verified by the client programs, and this request should include the information parameters, in which the key generation date is hidden. The intelligent platform of control generates the platform parameters, public-key PK_p and private-key SK_p. The platform sends the platform parameters and PK_p to the client.

Step 3. The client generates the shared session key according to platform parameters. The application encrypts user information and register code by using the shared session key, and then uses PK_p to encrypt the shared session key.

Step 4. The mobile devices send the text encrypted by shared session key and shared session key encrypted by PK_p to intelligent control platform, in which the text includes user information and register code and was saved in the platform service.

Step 5. Users need to sign in the client program before they want to control the smart device. Users submit the user login message encrypted by shared session key to intelligent control platform. The platform uses the platform SK_p to decrypt the shared session key, which saved in platform service. Then the platform uses shared session key to decrypt the login message. The platform verifies the correctness of the user login message.

Diffie-Hellman key exchange algorithm is a relatively common key exchange protocol, whose validity depends on the difficulty on computing discrete logarithm. However, there are a lot of shortcomings in it, such as computational intensive that is vulnerable to obstructive attacks.

2.3 Advantages of Polymorphic Cipher

The polymorphic cipher (PMC) provides a general pattern of realizable self-compiled ciphers, which was proposed by Roellgen [17]. One of its advantages is to use the self-compiler to arrange the encryption algorithms randomly. The other one is that the encryption and decryption speed of PMC are also significantly higher than AES.

Performance evaluation of PMC is structured by comparing with AES (Rijndael) in the software environment using C++ compiler on 64-bit Windows7 system. Fig. 2 shows the encryption speed comparison of PMC and AES. In terms of the encryption speed described in Fig. 2, PMC shows better performance and greater advantage compared with AES, especially for long sequences.

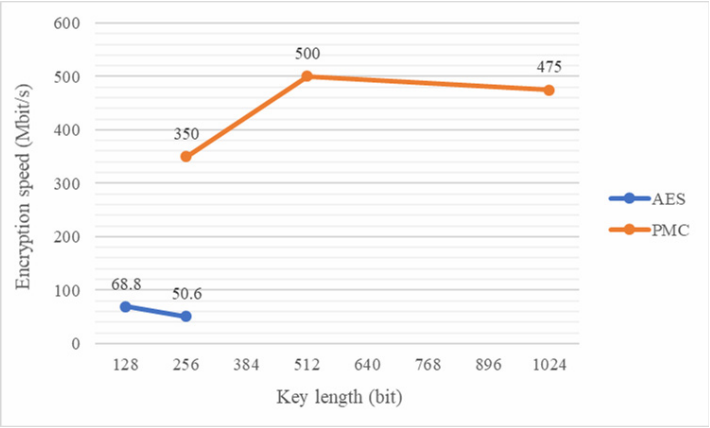


Fig. 1. The encryption speed comparison of PMC and AES

3 Polymorphic Key Exchange Protocol for the Authorization Chain Smart Terminal

3.1 Model Establishment of the Authorization Chain

The product authorization chain generally contains three roles: producer, consumer and agent, which includes primary agent, secondary agent, and nth agent. Different roles get the corresponding product information by authorizing. We build a model of the product authorization chain. Fig. 3 shows the authorization process of the product authorization chain. The current role registers and certificates, then authorizes corresponding amount of information to the next role, which completes that through the cloud service. Producer knows all product information content *C*, such as product ingredients, processing technology, cost of production and so on. Primary agent, secondary agent, nth agent and consumer receive the same or different corresponding product information content, in which *C1*, *C2*, ⋯, *Cn+m+1*, *C1i*, *C3j* are equal or not. Generally, the current role does not authorize all product sensitive information to the next level role. Therefore, the relationship of product information content is shown as: producer>primary agent>secondary agent>nth agent>consumer, which is given by the previous level in the product authorization chain based on smart terminals. With the substantial improvement of the deployment of IoT [18] in the intelligent terminal transaction [19], the demand for secure communication between intelligent terminals and server is increasing sharply in the authorization chain.

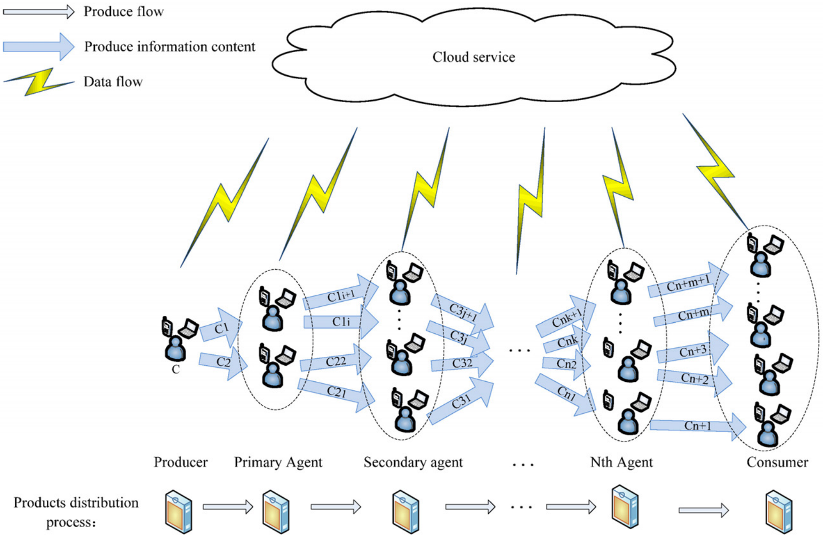


Fig. 3. The authorization process of the product authorization chain

3.2 Design of Random Group Operation

In order to improving iterated block cipher, we design a random group operation. The round function is embedded on random bit permutation [20]. It can against the differential cryptanalysis attacks and brute force attacks better. The structure formula of the random group operation is showed as

$$\begin{cases} f_1(a,b) = Q(P(a \otimes b)) \\ f_2(a,b) = Q(a) \otimes b \end{cases} . \tag{1}$$

In the above formula 1, $\otimes$ is Able random group operation, it picks value equiprobability from the random group $\{\otimes_1, \otimes_2, \otimes_3, \cdots, \otimes_i\}$, $P(.)$ is a convolutional cipher function, $Q(.)$ is an involution permutation and the automorphism of $\otimes$, $f_1(.)$ is a round function, and $f_2(.)$ is an output transformation function. And there is a great similarity between the encryption and decryption of the random group operation.

3.3 Improvement of PRNG

Table 1 lists the notations used in Section 3.

Table 1. Notations

Notations	Meanings
a_i ,b_i	32 bit input and 32 bit input
m	The modulus
s_3, s_2, s_1, s_0	Four continue bits of a pseudo random number sequence
$g(.)$	One-way functions as the factors in random group
$h(.)$	Strong one-way hash function
r_0	Initial 128 bit random seed
r_n	128 bit output of 128 bit output of n^{th} time
K_p , K_c	The session key of the platform and client
$S(.)$	A virtual iterated function contained a self-compiling system
K_{sp} , K_{sc}	Private key parameter combinations of the platform and client
$F_{platform}$, F_{client}	Respective secret subkeys of the platform and client
V_{fhash}	Hash functions for virtual iterative function
F_1 , F_2	Respective established secure session keys of the client and platform

PRNG (Pseudo Random Number Generator) [21] is the core of polymorphic cipher. It is used to generate random sequences. We improved the traditional PRNG by using polymorphic cipher. The improved PRNG used the random group to generate random sequences in accordance with the particular standard of the polymorphic key exchange protocol protocol. Fig. 4 shows the mechanism of the improved PRNG in the following.

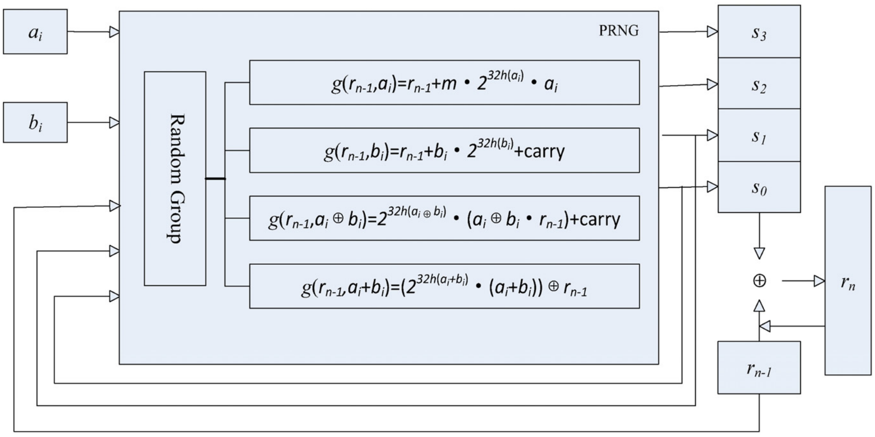

a_i
b_i
PRNG
Random Group
$g(r_{n-1},a_i)=r_{n-1}+m \bullet 2^{32h(a_i)} \bullet a_i$
$g(r_{n-1},b_i)=r_{n-1}+b_i \bullet 2^{32h(b_i)}$+carry
$g(r_{n-1},a_i \oplus b_i)=2^{32h(a_i \oplus b_i)} \bullet (a_i \oplus b_i \bullet r_{n-1})$+carry
$g(r_{n-1},a_i+b_i)=(2^{32h(a_i+b_i)} \bullet (a_i+b_i)) \oplus r_{n-1}$
s_3
s_2
s_1
s_0
$\oplus$
r_n
r_{n-1}

Fig. 4. The improved PRNG

We select four one-way functions as random seeds in random group, and use them as the self-compiling functions of polymorphic cipher. The one-way functions in the random group is defined as follows:

$$\begin{cases} g(r_{n-1},a_i)=r_{n-1}+m\bullet 2^{32h(a_i)}\bullet a_i \\ g(r_{n-1},b_i)=r_{n-1}+b_i\bullet 2^{32h(b_i)}+\text{carry} \\ g(r_{n-1},a_i\oplus b_i)=2^{32h(a_i\oplus b_i)}\bullet(a_i\oplus b_i)\bullet r_{n-1})+\text{carry} \\ g(r_{n-1},a_i+b_i)=(2^{32h(a_i+b_i)}\bullet(a_i+b_i))\oplus r_{n-1} \end{cases}. \quad \mathbf{(2)}$$

The probability of each function being chosen is equal in this random group. We combine random group with self-compiling polymorphic cipher. Variables a_i and b_i are 32-bit input sequences. From the random group, we select two functions randomly to generate two new 32-bit sequences. The new 32-bit sequences are regarded as new inputs to be sent to the random group again. Finally, we combine the four 32-bit output sequences into a 128-bit sequence.

The improved PRNG with the random group is described as follows:

Step 1. Choosing a 32-bit sequence is as a random seed of PRNG, that is r_0. In order to satisfy the requirement of the algorithm, we set the bit of r_0 to be 128, in which the low 32 bits are filled by ordinary binary number, and other bits are 0. Two sets of 32-bit data randomly obtained from the network node are input as the initial input of the improved PRNG, which are a_i and b_i.

Step 2. The two 32-bit data a_i and b_i are fed into the improved PRNG, which are gained from the network are input sequences. The sequence generator chooses two one-way functions randomly in the random group to generate two 32-bit sequences.

Step 3. The resulting 32-bit sequences is re-entered as new inputs into the random group to produce two sequences again.

Step 4. PRNG combines the four 32 bit sequences into a 128-bit output that is $A=s_3\,||\,s_2\,||\,s_1\,||\,s_0$, then makes $r_0\oplus A\rightarrow r_n, r_n\rightarrow r_{n-1}$. r_n is the final 128-bit output sequence.

Step 5. If the recipient thinks the final 128-bit output sequence are dissatisfactory, we carry out Step 2 to Step 4 again, until PRNG gets the pseudo-random sequences that satisfy cryptographic properties, and then output it.

3.4 Design of Polymorphic Key Exchange Protocol

To make the authorization chain smart terminal system fast and secure communication, the polymorphic key exchange protocol used by smart terminal system is presented. Details of each step of the polymorphic key exchange protocol is described as Fig. 5 showing.

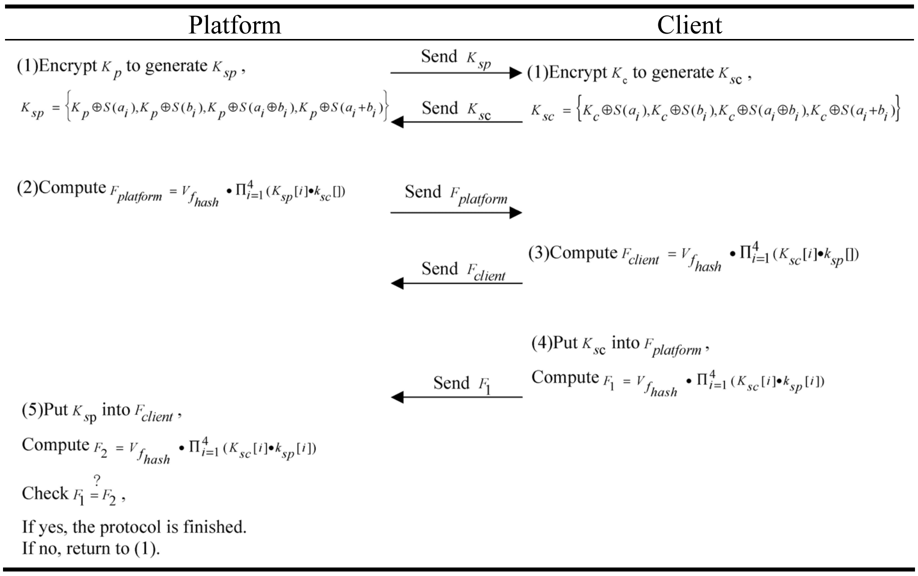

Platform
Client
(1)Encrypt K_p to generate K_{sp} ,
$K_{sp}=\{K_p\oplus S(a_i),K_p\oplus S(b_i),K_p\oplus S(a_i\oplus b_i),K_p\oplus S(a_i+b_i)\}$
Send K_{sp}
(1)Encrypt K_c to generate K_{sc} ,
Send K_{sc}
$K_{sc}=\{K_c\oplus S(a_i),K_c\oplus S(b_i),K_c\oplus S(a_i\oplus b_i),K_c\oplus S(a_i+b_i)\}$
(2)Compute $F_{platform}=V_{f_{hash}}\bullet\Pi_{i=1}^{4}(K_{sp}[i]\bullet k_{sc}[])$
Send $F_{platform}$
(3)Compute $F_{client}=V_{f_{hash}}\bullet\Pi_{i=1}^{4}(K_{sc}[i]\bullet k_{sp}[])$
Send F_{client}
(4)Put K_{sc} into $F_{platform}$,
Compute $F_1=V_{f_{hash}}\bullet\Pi_{i=1}^{4}(K_{sc}[i]\bullet k_{sp}[i])$
Send F_1
(5)Put K_{sp} into F_{client} ,
Compute $F_2=V_{f_{hash}}\bullet\Pi_{i=1}^{4}(K_{sc}[i]\bullet k_{sp}[i])$
Check $F_1\overset{?}{=}F_2$,
If yes, the protocol is finished.
If no, return to (1).

Fig. 5. The polymorphic key exchange protocol

Step 1. The platform and client use the improved PRNG to encrypt their session key K_c and K_p. They construct two parameter combinations K_{sp} and K_{sc} by using random sequences and session keys. K_{sp} and K_{sc} are the encrypted session keys of the platform and client. Then the platform and client send K_{sp} and K_{sc} to each other. The calculation formula about K_{sp} and K_{sc} are as follows:

$$\begin{cases} K_{sp} = \left\{K_p \oplus S(a_i), K_p \oplus S(b_i), K_p \oplus S(a_i \oplus b_i), K_p \oplus S(a_i + b_i)\right\} \\ K_{sc} = \left\{K_c \oplus S(a_i), K_c \oplus S(b_i), K_c \oplus S(a_i \oplus b_i), K_c \oplus S(a_i + b_i)\right\} \end{cases}. \tag{3}$$

Step 2. The platform calculates the parameter set K_{sp} to get $F_{platform}$ and then sends $F_{platform}$ to the client. $F_{platform}$ are derived as

$$F_{platform} = V_{f_{hash}} \bullet \prod\nolimits_{i=1}^{4} (K_{sp}[i] \bullet k_{sc}[]) . \tag{4}$$

Step 3. The client computes the parameter set K_{sc} to obtain F_{client}, and then sends F_{client} to the platform. F_{client} is obtained as

$$F_{client} = V_{f_{hash}} \bullet \prod\nolimits_{i=1}^{4} (K_{sc}[i] \bullet k_{sp}[]) . \tag{5}$$

Step 4. The client puts K_{sc} into the above formula 4 to obtain F_1. Then the client sends F_1 to the platform. F_1 is acquired as follows:

$$F_1 = V_{f_{hash}} \bullet \prod\nolimits_{i=1}^{4} (K_{sp}[i] \bullet k_{sc}[i]) . \tag{6}$$

Step 5. The platform puts the K_{sp} into F_{client} and computes it to obtain F_2. Then the platform sends F_2 to the client. If F_1=F_2, the polymorphic key exchange protocol is finished. Otherwise, return to Step 1. The calculation formula about F_2 is as follows:

$$F_2 = V_{f_{hash}} \bullet \prod\nolimits_{i=1}^{4} (K_{sc}[i] \bullet k_{sp}[i]) . \tag{7}$$

4 Security Analysis of Polymorphic Key Exchange Protocol

In the communication process, the main attacks method adopted by attackers is to acquire private keys and the initial random seeds, in order to obtaining the shared session key, thereby acquiring the subsequent session. Since the improved PRNG has the self-compilation characteristic of the polymorphic cipher, each attack from attackers will be as a part of the input data to change the internal algorithm of PRNG. Therefore, even if adversaries get the random seed, the session key of communication parties can't be guessed. This attack does nothing.

Secondly, attackers want to obtain the initial input data from both sides of communication. Because of the immediacy of network data, attackers can't get them. When an attacker conducts a network attack, it will cause the improved PRNG to self-compile and fail to obtain the final session key of both parties. This attack fails.

In addition, attackers spoof one party of communication and attempt to implement a man-in-the-middle attack. Since attackers cannot obtain the internal structure of PRNG, the final session key cannot be generated by attackers, even if the transmission parameters of both parties are intercepted by attackers. This attack is failed.

The security analysis shows that the polymorphic key exchange protocol can protect against the common types of attacks. Simultaneously, it uses the polymorphic cipher to enhance the speed of encryption and decryption.

5 Strict Avalanche Criterion analysis

5.1 Strict Avalanche Criterion (SAC)

SAC was proposed by Webster and Tavares in 1985. It points that the reversal probability of each output bit is 50 percent [22]. Setting function $f(.):GF(2)^n \to GF(2)^m$, which satisfies the strict avalanche criterion, when $g \in GF(2)^n$ and $W_H(g)=1$, $f(x+g)+f(x)$ is the balance function, where x is a plaintext. We choose a and b as the set elements. Each output bit has 50 percent possibility to be changed with when the bit of input sequence changes one bit.

$$\frac{1}{2^n}W(g_b^{e_a})=\frac{1}{2},\ a,b\in(1,2,\ldots,n)\,. \tag{8}$$

The above formula 8 displays the ideal result when the test of SAC finishes. Supposing $f(.)$ has two n bit sequences to be as inputs and two n bits sequences to be as outputs. The two input sequences have different values in one bit, then the hamming weight [23] of two sequences is 1, whose function is $w(.)$. Considering x_1 and x_2 are two input sequences, and y_1, y_2 are two output sequences. After one bit of x_1 and x_2 is changed, the bits in two output sequences have 50 percent possibility to be inverted. We get the probability:

$$p(y_1,y_2)=\frac{w(y_1\oplus y_2)}{n}\,. \tag{9}$$

Fig. 6 shows the changing rates of input and output bits.

Input sequences		Probability of the output bit being changed
x_1 and x_2 are two input sequences, $g \in GF(2)^n$		y_1 and y_2 are two output sequences
x	$\xrightarrow{a\to b}$	$\frac{1}{2^n}W(g_b^{e_a})=\frac{1}{2}\quad a,b\in(1,2,\ldots,n)$
$x_1,x_2,x_1=x_2\oplus g$	$\longrightarrow$	$p(y_1,y_2)=\frac{w(y_1\oplus y_2)}{n}$

Fig. 6. The changing rates of input and output bits

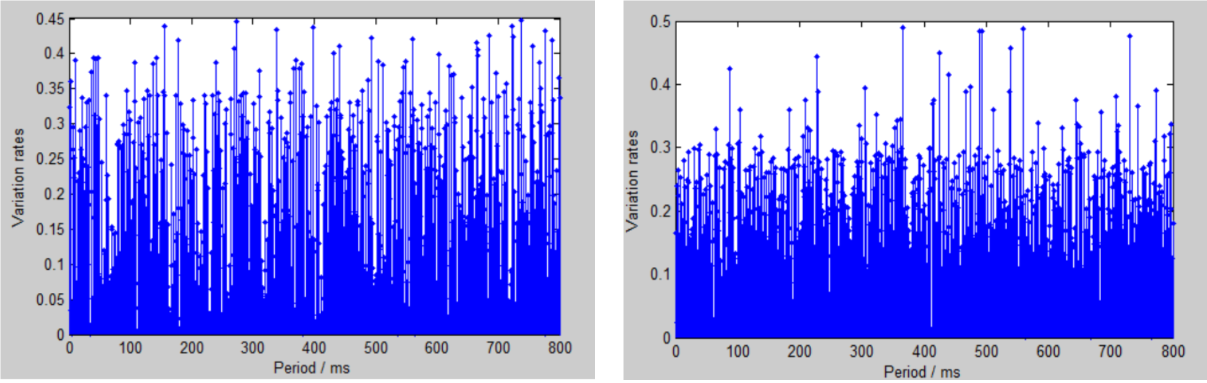
5.2 Experiment Results Comparison

5.2.1 Improved PRNG test SAC

We must test if the polymorphic key exchange protocol satisfies the SAC by checking the output sequences of the improved PRNG. In order to verify the strict avalanche criterion of the virtual iterative function, we have conducted extensive experiments in the network environment with 1 Gbps bandwidth. We choose 1000 set {a_i, b_i} as the improved PRNG's inputs, and a_i and b_i are 32-bit data, which are obtained randomly in network nodes. In terms of the quantitative analysis of output sequences, we can use the Hamming distance to observe the changing extent of one output with two inputs. When 1-bit of a_i and b_i is changed, it causes the changing rates of the output's bit to be close to 0.5.

Supposing $\partial a=\frac{a_h(a_i,a_{i-1})}{32}$ and $\partial b=\frac{b_h(b_i,b_{i-1})}{32}$ is the rate of the changed bit in a_i and b_i sequences. And $\partial r=\frac{r_h(r_i,r_{i-1})}{128}$ is the rate of the changed bit in r_n.

Fig. 7 shows the bit changing rates of 32-bit format PRNG inputs intercepted by a network node in a period of 800ms. As we can see from Fig. 7, ∂a and ∂b frequently are 0, videlicet, the parameters obtained by the network node are monotonous at some two instants. Fig. 8 shows bit changing rates of the improved PRNG output.



(a) The bit changing rates of ∂a (b) The bit changing rates of ∂b

Fig. 7. The bit changing rates of ∂a and ∂b

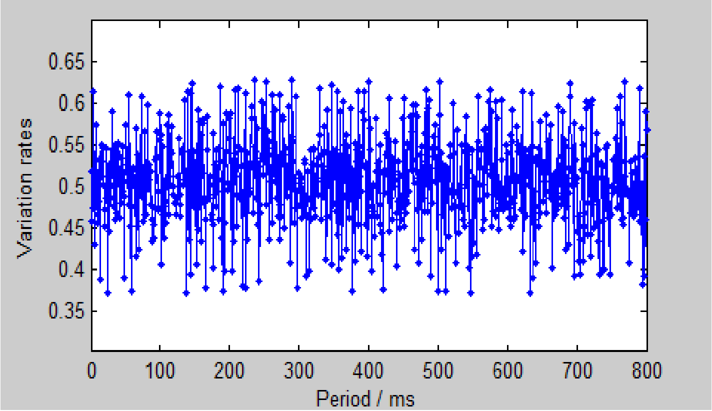


Fig. 8. The bit changing rates of ∂r

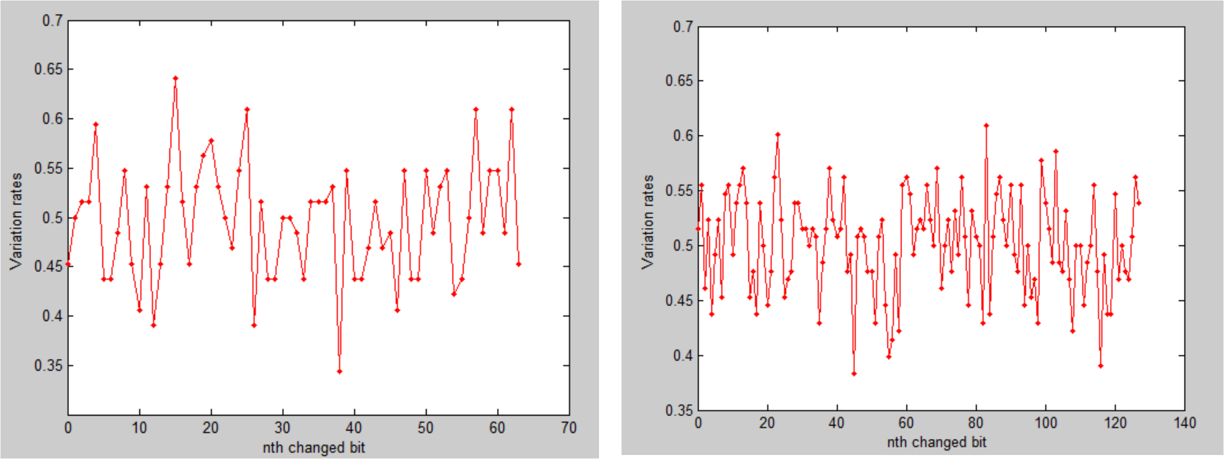
From the experimental result, we can know that with the bit of two 32-bit input sequences changing, 128-bit output's bit rates of change are mostly about 0.5, in Fig. 8, which fluctuates around 0.5. The bit changing rates of outputs are the range from 0.37 with 0.65 shown in Fig. 8. It explains that the system can satisfy SAC in the general case.

5.2.2 The Avalanche Effects of DES and AES

To compare with the improved PRNG, the avalanche effects of DES and AES are tested.

Data Encryption Standard (DES) and Advanced Encryption Standard (AES) are two symmetric-key algorithm. As a block cypher, DES is designed according to the Feistel structure with a 56-bit key size and 64-bit block size, which is put forward by International Business Machines (IBM) in 1972 [24]. There is a 64-bit output ciphertext for DES through cryptographic operations. When the encryption key is fixed, two input plaintexts only vary a bit among them. We compare the change bit rates of the corresponding output ciphertext to test the avalanche effect of DES, and the result is given in Fig. 9(a).

Adopted as the standard of encryption, Advanced Encryption Standard (AES) is designed in 1998, which has 128, 192 or 256-bit key size and 128-bit block size [25]. AES is the only one that it has 128-bit block size. It is a Rijndael variant based on a substitution-permutation network without using Feistel structure. AES has displaced the DES and Triple DES algorithms, and the last one is beginning to disappear little by little. In this experiment, we choose 128-bit input plaintexts and 128-bit encryption keys. We make the comparison of the changed bit rates between two output ciphertexts, where the encryption key is constant and two input plaintexts only change a bit. The test result is given in Fig. 9(b).



(a) The bit changing rates of DES output

(b) The bit changing rates of AES output

Fig. 9. The bit changing rates of DES output and AES output

In Fig. 9(a), the gradient ranges of the changed bit in two output ciphertext are between 0.34 and 0.65, which have large fluctuations around 0.5. With the bit of 64-bit input sequences changing, in which encryption key is always invariant, 128-bit output's bit changing rates are close to 0.5. It proves that DES meets the requirement of SAC.

In Fig. 9(b), the bit changing rates in two output ciphertext fall in the interval from 0.38 to 0.61, and they have a high undulation nearby 0.5. The 128-bit output's bit changing rates are over 0.5 in most situations. It demonstrates that AES cuts the mustard in terms of SAC.

5.2.3 Data Statistics Analysis

In order to prove the accuracy of the experiment, we count the experimental result, as shown in Table 2 and Table 3. In Table 2, we analyze the statistical data from Fig. 7 and Fig. 8 to verify the strict avalanche characteristic of the improved PRNG.

Table 2. The bit changing rates of ∂a , ∂b and ∂r

Object	Max	Min	Mean	Standard deviation
∂a	0.4494	0	0.1632	0.0939
∂b	0.4976	0	0.1083	0.0945
∂r	0.6485	0.3768	0.4997	0.0412

From Table 2, when both of input's bit rates are 0, the output's bit rate is 0.3768. The average rate of the output's changing bit in the whole process is 0.4997. And the standard deviation of the ∂r bit changing rates is 0.0412, which shows the stability of the improved PRNG.

Table 3. The comparison of DES, improved PRNG and AES

Object	Input bit	Output bit	Max	Min	Mean	Standard deviation
DES	64	64	0.6406	0.3438	0.4954	0.0598
Improved PRNG	64	128	0.6485	0.3768	0.4997	0.0412
AES	128	128	0.6094	0.3828	0.5012	0.0807

In Table 3, we select the data of Fig. 8 and Fig. 9 to be as the changed bit rates and count the experimental data about improved PRNG, DES and AES. As far as the data in Table 3 is concerned, AES is the best in terms of satisfying SAC. Nevertheless, the output bit is the fewest and the fluctuation of the bit changing rates is the highest. DES and improved PRNG are meet the demand of SAC, which are very close to 0.5. And the standard deviation of the bit changing rates is the lowest for improved PRNG, which illustrates that the anti-differential attack performance is more stable. What's more, the input bit of DES is 64, which is identical with its output bit. and AES has 128-bit inputs and outputs in experiment. By contrast, there is the improved PRNG that has the same input bit with DES and uniform output bit with AES.

The first vulnerability of DES algorithm is its small key length, which means it easier to be broken up by brute force. In 1998, the feasibility of cracking DES quickly was certified by the DES-cracker, which was built by Electronic Frontier Foundation (EFF). The machine violently cracked the key in the period of more than 22 h and 15 min [26]. With key size increasing, the resources used to brute-force attack grow exponentially. Modern symmetric algorithms usually use 128 and 256 bits, which are impossible to break this cypher in fact. The complexity to break 128-bit keys is 2128, and it means that it has 3.4×1038 possible combinations, which implies 1.02×1018 years, amounting to one billion years, to crack. The improved PRNG generates into a 128-bit sequence to proceed encryption operations in spite of only two 32-bit input sequences. It has a 64-bit input in common with DES, but its anti-differential attack performance is same with 128-bit AES. It demonstrates the improved PRNG is secure to brute-force attack and provides higher security for polymorphic key exchange protocol.

6 Conclusions

This paper defines an important model about the process of the product authorization chain. And this paper constructs the PRNG based on random group and uses the polymorphic key exchange protocol based on modified PRNG to ensure the security of data transmission in smart terminal system of the authorization chain. From the SAC test and experiment result of improved PRNG, DES and AES, we can conclude that the improved PRNG has high scalability by using the component of the public virtual iterative function. It can generate the periodic and random sequences which satisfies cryptology character in the authorization chain smart terminal based on polymorphic key exchange protocol. What's more, it also can resist against differential cryptanalysis. In the further, we can optimize the polymorphic key exchange protocol to achieve the high efficiency and security between the client and intelligent platform in the authorization chain.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 61572445 and No. 61272038.

References

[1] L.D. Xu, W. He, S. Li, Internet of Things in industries: a survey, IEEE Transactions on Industrial Informatics 10(4)(2014) 2233-2243.

[2] IDC Analyze the Future, The digital universe of opportunities: rich data and the increasing value of the internet of things. <https://uk.emc.com/leadership/digital-universe/2014iview/internet-of-things.htm>, 2014.

[3] N. Kaaniche, M. Laurent, Data security and privacy preservation in cloud storage environments based on cryptographic mechanisms, Computer Communications 111(2017) 120-141.

[4] S. Qiao, Y. Zeng, L. Zhou, Z. Liu, J. Ma, A Secure Authentication Method of Intelligent Terminals Based on Jensen-Shannon Divergence, in: Proc. 2017 International Conference on Networking and Network Applications, 2017.

[5] A. Siddiqa, A. Karim, A. Gani, Big data storage technologies: a survey, Frontiers of Information Technology & Electronic Engineering 18(8)(2017) 1040-1070.

[6] S. Hong, S. Park, L.W. Park, M. Jeon, H. Chang, An analysis of security systems for electronic information for establishing secure internet of things environments: Focusing on research trends in the security field in South Korea, Future Generation Computer Systems 82(2018) 769-782.

[7] H. Dong, C. Wu, Z. Wei, Y. Guo, Dropping activation outputs with localized first-layer deep network for enhancing user privacy and data security, IEEE Transactions on Information Forensics & Security 13(3)(2018) 662-670.

[8] V. Stupka, M. Horák, M. Husák, Protection of personal data in security alert sharing platform, in: Proc. 12th International Conference on Availability, Reliability and Security, 2017.

[9] E.J.M. Capó, O.J. Cuellar, C.M.L. Pérez, G.S. Gómez, Evaluation of input — output statistical dependence PRNGs by SAC, in: Proc. 2016 International Conference on Software Process Improvement, 2016.

[10] G. Sánchez-Arias, C.G. García, B.C.P. G-Bustelo, Midgar: study of communications security among smart objects using a platform of heterogeneous devices for the Internet of Things, Future Generation Computer Systems 74(2017) 444-466.

[11] J. Cui, L. Huang, H. Zhong, C. Chang, W. Yang, An improved AES S-box and its performance analysis, International Journal of Innovative Computing Information & Control 7(5)(2011) 2291-2302.

[12] M. Feltz, C. Cremers, Strengthening the security of authenticated key exchange against bad randomness, Designs Codes & Cryptography 86(3)(2018) 481-516.

[13] M.M. Kangethe, C. Kamau, Polymorphic Encryption Algorithms, in: Proc. AfricaHackOn Cyber Security Annual Conference, 2015.

[14] H. Zhu, X. Hai, J. Lin, Integral and impossible differential cryptanalysis of RC6, in: Proc. International Conference on Cloud Computing and Security, 2018.

[15] Y. Yin, Y. Yang, Y. Gan, M. Zhang, Researching security mechanisms of the polymorphic authentication service protocol, Journal of Computational Information Systems 9(7)(2013) 2641-2647.

[16] Y.H. Hou, X.L. Zhou, Security problems and solutions of smart terminal mobile operating system, Telecommunications Science 31(3)(2015) 2015077.

[17] C.B. Roellgen, Polymorphic cipher theory. <https://http://www.pmc-ciphers.com/eng/content/Backround-Info/PMC-Explained.html>, 2007.

[18] R. Du, C. Liu, F. Liu, Trust authorization monitoring model in IoT, Journal of Perlormability Engineering 14(3)(2018) 453-462.

[19] Z. Gao, L. Xu, L. Chen, Y. Lu, W. Shi, CoC: a unified distributed ledger based supply chain management system, Journal of Computer Science and Technology 33(2)(2018) 237-248.

[20] V. Patidar, N.K. Pareek, G. Purohit, K.K. Sud, A robust and secure chaotic standard map based pseudorandom permutation-substitution scheme for image encryption, Optics Communications 284(19)(2011) 4331-4339.

[21] M.A. Ivanov, E.B. Roslyj, A.V. Starikovskiy, S.A. Krasnikova, N.A. Shevchenko, L.I. Shustova, Non-binary pseudorandom number generators for information security purposes, Procedia Computer Science 123(2018) 203-211.

[22] Q. Wang, A new method on constructing boolean functions satisfying the strict avalanche criterion and bounds on the number of SAC functions, in: Proc. 2015 3rd AASRI Conference on Computational Intelligence and Bioinformatics, 2015.

[23] J. Park, I. Kim, H.Y. Song, Construction of parity-check-concatenated polar codes based on minimum Hamming weight codewords, Electronics Letters 53(14)(2017) 924-926.

[24] M.F. Mushtaq, S. Jamel, A.H. Disina, Z.A. Pindar, N.S.A. Shakir, M.M. Deris, A survey on the cryptographic encryption algorithms, International Journal of Advanced Computer Science & Applications 8(11)(2017) 333-343.

[25] S. Heron, Advanced Encryption Standard (AES), Network Security 2009(12)(2009) 8-12.

[26] S. Kumar, C. Paar, J. Pelzl, G. Pfeiffer, A. Rupp, M. Schimmler, How to break des for euro 8,980, in: Proc. 2nd Workshop on Special-purpose Hardware for Attacking Cryptographic Systems, 2006.