Journal of Computers Vol. 31 No. 1, 2020, pp. 166-175
doi:10.3966/199115992020023101013

Cross-regional Cross-level Encryption Collaborative Data Transmission

Tongfei Yao[1,2], Yun Liu[1,2*], Shih-Chen Wang[3], Sheng-Lung Peng[3], Kun Mi[4], Zhihong Ying[4]

[1] School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China

[2] Key Laboratory of Communication and Information Systems, Beijing Municipal Commission of Education, Beijing, China
{18120165, liuyun}@bjtu.edu.cn

[3] National Dong Hwa University, Hualien, Taiwan
{810621002, slpeng}@ndhu.edu.tw

[4] Beijing Thunisoft Information Technology Corporation Limited

Received 29 September 2019; Revised 19 October 2019; Accepted 29 October 2019

Abstract. In recent years, there has been a sharp increase in the number of collaborative execution cases in China. The enforcement of major difficult cases or some related cases need to be accepted and coordinated in multiple courts. Such cases have a significant affect, the implementation process is often highly concerned by the society. Therefore, it often relies on an integrated and coordinated remote command platform. How to ensure reliable data transmission of the collaborative command platform has become a key technical issue. In the existing public network environment, active attacks or passive attacks and other network attacks make data transmission unreliable. Therefore, in addition to the existing necessary security guarantees, we have presented a hybrid encrypted data transmission method based on elliptic curve cryptography (ECC) and a lightweight stream cipher encryption for the collaborative remote command platform. It provides safe and effective auxiliary support to realize secure communication, remote collaboration and cross-level management for the cross-regional cross-level's courts.

Keywords: encrypted transmission, elliptic curve, hybrid encryption, stream cipher

1 Introduction

With the development of information, more and more major cases have received social attention, and dealing with complex and major cases must rely on the simultaneous implementation of courts at different levels and in different regions. Therefore, establishing a secure integrated collaborative command platform and a multi-part cooperative execution mechanism is an inevitable choice to improve the efficiency and quality of execution cases. Due to the inherent openness of the network, how to ensure the security of data onto the data transmission system has also become an important issue.

At present, there is no safe and reliable cross-regional cross-level collaborative command platform in the court field. Courts still rely on manpower dispatch and written collaborative requests, which waste resources and is not efficient enough. The research on the encrypted collaborative data transmission platform can solve this problem well. Our main contributions are as follows:

(1) Design and develop a cross-regional and cross-level collaborative work structure, which can help court units to implement cases more effectively. This can realize information sharing among court units, improve the processing efficiency of complex cases, and improve the processing speed of emergencies.

(2) Develop a data encryption method to complete the encrypted data transmission of the above

* Corresponding Author

platform. The traditional single encryption algorithm becomes more and more insecure as the index of computer computing power increases. For the security threats faced by network data transmission, according to the characteristics of various cryptographic technologies and the data of enforcement records of the court's mission process is large, this paper presents a hybrid encryption method based on a lightweight hash chain encryption and the ECC encryption for network data transmission. Meanwhile, we present authentication function.

2 Related Works

This paper [1] considers security implications of k-normal Boolean functions when they are employed in certain stream ciphers. A generic algorithm is proposed for cryptanalysis of the considered class of stream ciphers based on a security weakness of k -normal Boolean functions. A new stream cipher, Grain-128, is proposed in [2]. In this paper [3] they present a very practical ciphertext-only cryptanalysis of GSM (Global System for Mobile communications) encrypted communication, and various active attacks on the GSM protocols. In this paper [4], an efficient signcryption scheme based on elliptic curve cryptosystem is going to be proposed which can effectively combine the functionalities of digital signature and encryption and also takes a comparable amount of computational cost and communication overhead. In this paper [5] evaluate the performance of these algorithms such as AES, DES, and RSA to encrypt text files under three parameters like computation time, memory usage, and output bytes [6].

Based on the advantages and disadvantages mentioned above, this paper presents a hybrid encryption method based on lightweight hash chain and ECC.

3 System Analysis

3.1 Cross-regional Cross-level Collaborative Command System

The cross-regional cross-level collaborative command system is mainly composed of the court web server and the individual law enforcement APP terminal. The individual terminal accesses the intranet of the court through the public mobile network. And we use the unified identity authentication technology to control the accesses. The intranets of the courts are connected by VPN, and the mobile devices that perform tasks are connected through a web server. Firewalls are required to isolate the network of each stage to ensure the security of the internal network. The specific architecture of the collaborative command system is shown in Fig. 1.

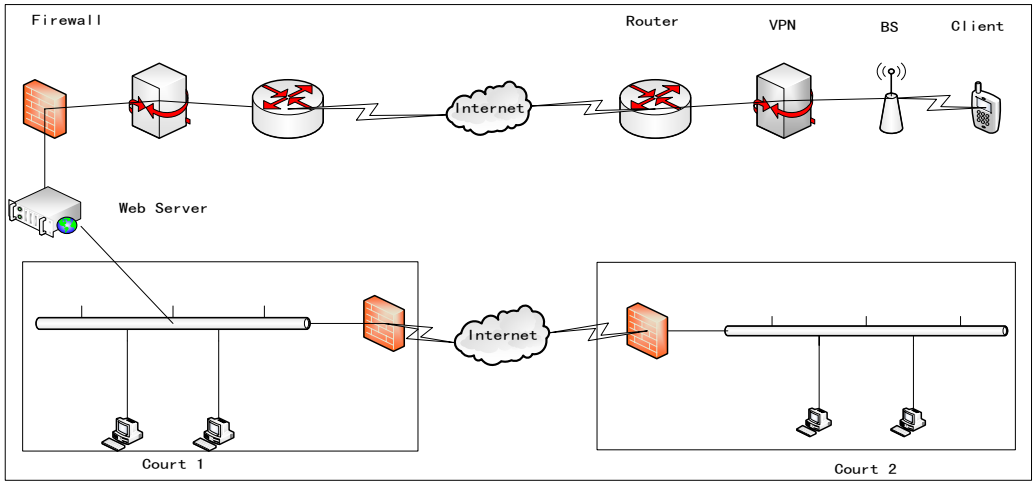

Fig. 1. Cooperative command system architecture

3.2 System Confidentiality Analysis

As the network's inherent openness and scalability, there are many security problems. Such as use the vulnerability of network to spread computer viruses, unauthorized intrusion user system, intrusion into

the privacy of the network information and so on. These can result in economic losses and even threaten National Security. When critical data is transmitted in the network, various types of attacks may be encountered. Therefore, we must use various services to ensure the confidentiality, integrity and availability of our data. The system's security design goals are as follows:

(1) Securing data confidentiality: The system must ensure that only the intended recipient of the file can read the file information.

(2) Guarantee the integrity of the data: The recipient of the document can verify whether the file he received is complete and has been illegally altered.

(3) Authentication of the data source: The file recipient can verify that the file was indeed sent by the specified sender.

(4) The real-time efficiency of data transmission: Since the law enforcement records are often viewed remotely by the superior during the execution of the court's tasks, the efficiency of the data transmission must be ensured to respond to various situations in a timely manner.

The courts at all level need to be isolated from the external networked, that is, using the IP address of the intranet network segment. The interconnection of courts and mobile terminals at all levels requires VPN technical support. In order to prevent ip spoof to maintain intranet security, firewall filtering is required. The system also need access control technology, that is, unified identity authentication technology. It is used to ensure the access control of the intranet. The data generated during the execution of the case, including authentication data, video data, etc, need to ensure secure by symmetric encryption technology and public key encryption technology.

According to the above system characteristics, the encrypted collaborative data transmission platform is divided into an encryption module and a transmission module.

4 Encryption and Authentication Module

In view of the characteristics of collaborative interactive tasks, a lightweight encryption transmission mechanism in collaborative task environment is proposed by combining one-time password scheme. The encryption mechanism adopts stream cipher and adds an encryption authentication algorithm which can authenticate the client and server [7].

4.1 Bidirectional Authentication Mechanism

The shared information between user U and server S is: symmetric key k of U, encryption algorithm $E_k(m)$, decryption algorithm $D_k(m)$, secure one-way Hash function H, The user's identifier Uid. And server S holds user's $H(Uid)$ list. The protocol process is as follows:

Client:

Calculate h, send h to S, and start the timeout counter.
If the timeout does not receive an answer from S, exit the session. If the reply of S is received, use k to decrypt to get x, hs, and determine whether hs is valid. If not, it proves that S is fake and terminates the session with S. If it is true, the identity of S has been verified, calculate a, encrypt eu, and then send eu to S.

$$\begin{cases} h = H(uid) \\ hs = H(x,k) \\ (x,hs) = D_k(es) \\ a = H(x,H,k) \\ eu = E_k(a) \end{cases} \quad \textbf{(1)}$$

Server:

Listen for customer request links. If not, switch to another task.
If any request arrives, determine whether h belongs to list: If so, *Uid* is valid. The random number generator generates random number x, and uses symmetric secret k to calculate hs, es then send the es to U and start the timeout counter. Otherwise, it indicates that U is an illegal user and terminates the session with U. If you time out, you end the session. If there is no timeout and the customer's reply is received, decrypt the received eu, a, calculate b, and verify whether $a = b$ is true: If true, the identity of U is verified. Otherwise, it indicates that U is an illegal user and terminates the session with U.

$$\begin{cases} x = random() \\ hs = H(x,k) \\ es = E_k(x,hs) \\ a = D_k(eu) \\ b = H(x,H,k) \end{cases} \tag{2}$$

4.2 Public Key Cryptosystem

Since stream cipher encryption requires an initial password during the initial establishment of each session, if an initial password is retained locally, local password storage protection is a problem. Therefore, we adopt a public key cryptography to pass the initial secret key after the establishment of each session. So we firstly introduce the public key cryptosystem and ECC [10-13] allocation method.

In the encryption method of block cipher (such as AES, etc.), [8] because the same encryption key is used every time, it is not conducive to the encrypted transmission of a large number of similar structural information, such as documents transmitted by the court and other institutions. As computers become more and more computationally powerful, this kind of grouping and encryption of information becomes riskier and riskier. However, courts and other institutions have high confidentiality requirements on information, so an efficient and secure encryption method is extremely important.

The scheme proposed in this paper combines the ECC encryption algorithm to transmit the key. Because ECC is more secure [9]. The speed of encryption and decryption is faster. The small key also determines the small storage space.

The receiver gets the initial key and uses the key to form a secret key stream to decrypt the data sented by the sender and get the plaintext. The Individual equipment's initial key of lightweight hash chain stream cipher encryption can generate by random function and UID.

Elliptic Curve Cryptography (ECC) [14] is an asymmetric encryption method based on elliptic curves mathematics, it was proposed in 1985 by NealKoblitz and VictorMiller. Mathematically, elliptic curve (EC) is a class of algebraic curves, wherein, the elliptic curve equation based on the finite field Zp is defined as:

$$y^2 = x^3 + ax + b \tag{3}$$

The elliptic curve equation based on the finite field $GF(2^m)$ is defined as:

$$y^2 + xy = x^3 + ax + b \tag{4}$$

The domain parameters of the elliptic curve cryptosystem are (a, b, G, n, h). q is a prime number, a and b are the coefficients of the elliptic curve equation in (3) and (4). The commonly used elliptic curves are non-singular, so it satisfies the equation $4a^3 + 27b^2 \neq 0$. G is the base point on the elliptic curve. n is the large prime order of point G, and n is the smallest positive integer which satisfies $n \times G = 0$. h is a cofactor of the order of the elliptic curve divided by n, and $h \leq 4$.

The choice of these parameters directly affects the security of encryption. The larger *p* is, the safer it is. However, the computing speed will slow down, and 200 bits can meet the general safety requirements.

Public key cryptography algorithms are always based on a mathematical problem. For example, RSA is based on that given two numbers, *p* and *q*, it's easy to multiply to get *N*, but it's much harder to factor *N* to get *p* and *q*. The mathematical problem used by the elliptic curve cryptosystem (ECC) is to solve the discrete logarithm problem of the elliptic curve addition group, which is described as follows:

$$Q = k \times G \tag{5}$$

Wherein Q is a point on an elliptic curve, G is a base point on the elliptic curve, k is an integer smaller than n, n is an order of G, also $n \times G = 0$. According to the rules defined by ECC, it is not difficult to obtain Q by k and G. But given Q and G, k is relatively difficult to find in the curve. This is called the discrete logarithm problem on an elliptic curve. Now we describe a process of encrypting by using elliptic curves:

The protocol process is as follows:

(1) User A selects an elliptic curve $E_p(a,b)$ and takes a point on the elliptic curve as the base point G.

(2) User A selects a private key k and generates a public key $K=kG$.

(3) User A transmits $E_p(a,b)$ and points K, G to User B.

(4) After receiving the information, user B encodes the plaintext to a point M on $E_p(a,b)$, and generate a random integer r (r < n).

(5) User B calculates point $C_1=M+rK; C_2=rG$.

(6) User B passes C_1 and C_2 to User A.

(7) After user A receives the information, it calculates C_1-kC_2, and the result is point M. because

$$C_1 - kC_2 = M + rK - K(rG) = M + rK - r(kG) = M$$

Then the point M is decoded to obtain the plaintext.

4.3 Lightweight Hybrid Encryption Algorithm Description

The algorithm has two parts. The initial secret key's transfer of stream cipher encryption and lightweight hash chain stream cipher encryption will be detailed in the upcoming content.

The initial secret key's transfer of stream cipher encryption is as follows:

The mobile client and the remote server negotiate a set of shared parameters: elliptic curve $E_q(a,b)$, the base point *G* of the elliptic curve and a large prime order n of point *G*. The system's encryption and decryption processes are shown in Fig. 2.

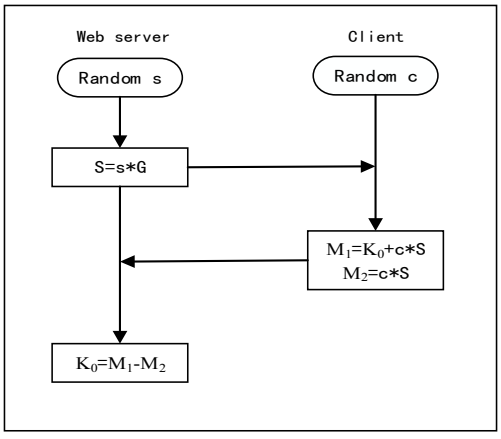

Fig. 2. The Initial Secret Key's Transfer

Encryption process shown in Fig. 2, the mobile client and the remote server select an elliptic curve $E_q(a,b)$ and the same parameters: a base point G and a prime order n of the point G. The client and the server respectively generate different random number c and s. And s multiplied point G to get S. The client calculates M_1 then sents to the server. The client uses its own random number c times S to get M_2. Then sents to the server, too. After getting M_1 and M_2, the server calculates the initial key K_0 of stream cipher encryption by $M_1 - M_2$.

It can be seen from the above process that the initial key K_0 of stream cipher encryption is encrypted by ECC. That is, the terminal cannot obtain the previous key without accessing the server through the authentication. So even if the eavesdropper keeps the encrypted data, he cannot crack the terminal to obtain the encryption key.

Lightweight hash chain stream cipher encryption is as follows:

The process of this scheme is mainly composed of initial secret key generation and transmission, stream encryption and stream decryption [8].

Initial secret key generation and transmission's proccessing is as follows:

The initial secret key is generated by the private device UID and random string X. The UID is the unique serial number of the remote device. Hash UID and X to obtain the initial secret key K_0.

$$\begin{cases} X = Random() \\ K_0 = Hash(UID, X) \end{cases} \tag{6}$$

Stream encryption's proccessing is as follows:

After the initial secret key is created and transferred, the data encryption client uses encryption algorithm to encrypt the hash value of key stream K_i and plaintext stream M_i by xor operation to obtain ciphertext stream C_i, and sends ciphertext stream C_i to the data decryption server for analysis. The specific encryption algorithm formula is as follows:

$$\begin{cases} C_i = Hash(K_i) \oplus M_i \\ K_i = C_{i-1} \oplus Hash(M_{i-1} \oplus K_0) \end{cases} \tag{7}$$

Where K_0 is the initial generation of the session establishment and is passed by the ECC, and the subsequent secret key stream is generated by the above formula.

Stream decryption's proccessing is as follows:

After the data decryption server receives the ciphertext stream C_i sented by the data encryption client, it uses the hash value of the key stream K_i generated by the data decryption party and the received ciphertext stream C_i to perform xor operation to decrypt and obtain the plaintext stream M_i. The specific decryption algorithm formula is as follows:

$$\begin{cases} K_i = C_{i-1} \oplus Hash(M_{i-1} \oplus K_0) \\ M_i = Hash(K_i) \oplus C_i \end{cases} \tag{8}$$

After receiving ciphertext stream C_i, the data decryption server first looks for the latest ciphertext stream C_{i-1} and plaintext stream M_{i-1} records in the device list:

(1) If no ciphertext stream C_{i-1} or plaintext stream M_{i-1} records are found, it means that the first time the data decryption server receives the data from the client. At this time i=0, data decryption server extracts K_0 transfered by ECC and decrypts data according to the formula.

(2) If the ciphertext stream C_{i-1} or plaintext stream M_{i-1} records are found, it means that the data decryption server is not the first time to receive the message sented by the data encryption client, and the data decryption server takes the ciphertext stream C_{i-1} and plaintext stream M_{i-1} received last time to obtain the key stream K_i after operation.

5 Transmission Module Analysis

The data transmission module is mainly the transmission of network data, which is a VPN wireless communication network based on 4G. At present, there are many ways of data transfer, such as socket and FTP. Since the system has real-time requirements for data transmission and errors are inevitable, the system establishes the connection between the two sides through the Windows Socket interface programming technology, and uses a simple sliding window mechanism to realize error retransmission and so on.

5.1 Sliding Window Mechanism

The sliding window protocol can solve the problem of data synchronization in the case of packet damage, packet loss and premature timeout. The sliding window protocol is divided into three types, namely, 1-bit sliding window protocol, back-n frame protocol and selective retransmission protocol. In the sliding window protocol, each packet is assigned a sequence number ranging from 0 to a maximum value representing a packet that has been sent but not yet acknowledged. Upon receipt of a data request from the application, the sender packages it and assigns a maximum sequence number, then increments the upper limit of the window by 1, and the lower limit of the window by 1 upon confirmation of arrival. In this way, the window can persist a series of unacknowledged packets. Because the current packet in the sender's window may be lost or damaged in transit, the sender must save all unacknowledged packets for retransmission.

The receiver's window corresponds to packets that are allowed to receive, and any packets that fall outside the window are dropped without explanation. When a packet with a serial number equal to the lower limit of the window is received, an acknowledgment is generated and the window is moved forward one position. The sliding window protocol generally adopts the so-called piggyback confirmation technology to improve the efficiency of data transmission. The process is that, when a packet arrives, the receiver does not immediately send an independent acknowledgement, but keeps waiting until the upper layer transmits the next packet to it, and the acknowledgement is attached to the packet to be sent, that is, the acknowledgement is attached to the next packet to be sent for transmission. This technology can effectively use the bandwidth and reduce the number of data communications.

5.2 Data Transmission Process

After the connection is established, the TCP protocol provides full-duplex communication services. The general client/server program flow is initiated by the client, and the server passively processes the request. Therefore, the server immediately calls *read*() after returning from *accept*(). Reading the socket is like reading the pipe. If no data arrives, it will block waiting. At this time, the client calls *write*() to send the request to the server. The server receives the *write*() of the client, then returns to process the client's request. During this processes, the client calls *read*() to block the response from the server, the server calls *write*() to send the result back to the client, and again calls *read*() to block the next request. After the client receives it, it returns from *read*(), sends the next request, and so on. If the client has no more requests, call *close*() to close the connection. If a part calls *shutdown*(), the connection is in a semi-closed state and can still receive data from the other part.

6 Algorithm Analysis

6.1 Security Analysis

The security of stream cipher depends entirely on the unpredictability and randomness of the key stream generated by the key stream generator. In the scheme, Ki is generated through the last stream cipher C_{i-1}, the last plaintext M_{i-1} and the initial key K_0. The hash value of the key stream is XORed with the current plaintext stream to obtain the current ciphertext stream C_i. The ciphertext stream after each XOR encryption is different, which ensures the encryption of different plaintext streams. The key flow is also

different, so the scheme has One-Time-Pad security. Since the key stream used for encryption in this scheme is generated by the first three parts XOR and hash operation, even if the key stream K_i corresponding to a ciphertext stream C_i and the plaintext stream M_{i-1} and the initial key are broken, the hash value, and the initial key cannot be obtained, guarantees the confidentiality of the scheme.

At the same time, the program has different random initial keys for each session and different devices, so even statistical analysis of large computing power is difficult to crack. Because the elliptical curve and G point that the eavesdropper cannot obtain, it is impossible to spoof the public key generated by the client and the server, so that the session initial key cannot be obtained, and the man-in-the-middle attack can be resisted.

This solution also provides a hash-based bidirectional authentication mechanism that can resist identity spoofing and spoofing.

6.2 Performance Analysis

ECC key pair's generation rate is faster than RSA, and ECC is a better choice in business scenarios where key pair's production is required in large quantities. Compared with single ECC, if each packet needs to produce an ECC secret key pair, would cause transmission efficiency too low and CPU consumption is serious. So it's not suitable for large amounts of data transmission. Combining our lightweight hash chain stream cipher encryption in this paper, it is more suitable for a large number of data transmission, and will improve the transmission efficiency and ensure the safety. Compared with other grouping encryption methods such as DES/AES/RC4, it is more efficient and secure.

7 Simulation

This experiment simulates the interaction of the encrypted data between the client and the remote server of the court, and tests the encryption and decryption process with python.

In the process of experimental simulation, 30 video data were collected through video stream simulation, and get the plaintext stream M_i. Every data is 30MB. At the same time, AES and DES encryption and decryption schemes were selected for experimental simulation comparison with this scheme, and the encryption time of the 30 plaintext data was obtained, as well as the decryption time of the 30 plaintext data of the communication server under different schemes. As shown in Fig. 3 and Fig. 4, the shortest average encryption time of this scheme is 2.3745 s. The shortest average decryption time of this scheme is 1.6621 s. The average encryption and decryption time of each scheme is shown in Table 1. The simulation results show that the lightweight stream secret key encryption scheme can encrypt and transmit the data generated by the remote soldier equipment efficiently.

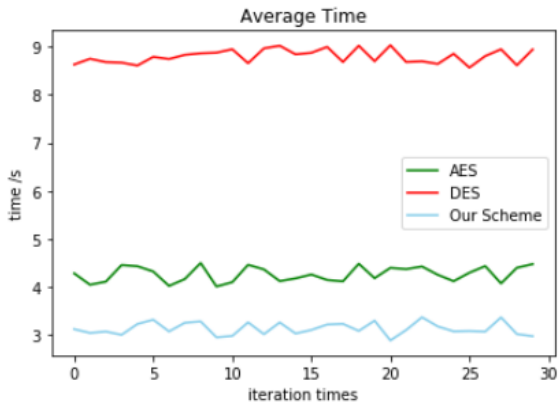

Fig. 3. Average time of encryption

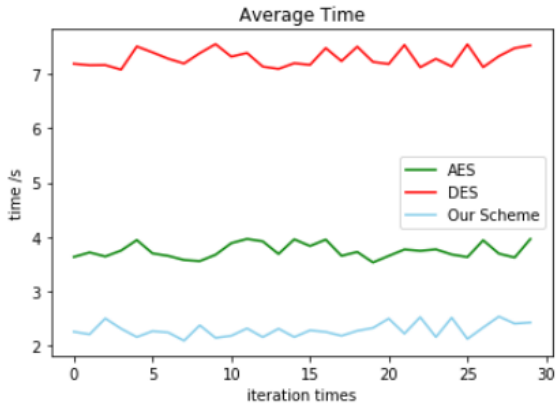

Fig. 4. Average time of decryption

Table 1. Average time taken by each scheme to encrypt and decrypt

Scheme	Average time of encrypt /s	Average time of decrypt /s
AES	4.305667	3.677017
DES	8.954062	7.501383
Our Scheme	2.374492	1.6621444

8 Conclusions

Encryption technology both now and in the future is the focus of all network data transmission. Symmetric encryption algorithm has the advantages of simplicity and fast encryption speed. The algorithm shows good performance in terms of algorithm implementation, efficiency and strength. On the other hand, the public key encryption algorithm is very convenient for key management, and it can implement functions such as data signature and identity verification. This paper combines ECC and a lightweight stream cipher encryption to ensure the transmission efficiency while ensuring the security of data transmission. It provides a safe and effective means of data transfer for the cross-regional and cross-level remote command platform. It provides a safe and effective auxiliary support for the cross-regional and cross-level courts to realize safe communications and cross-tier management.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (grant number 2018YFC0831300).

References

[1] M.J. Mihaljevi, S. Gangopadhyay, G. Paul, H. Imai, Generic cryptographic weakness of k -normal Boolean functions in certain stream ciphers and cryptanalysis of grain-128, Periodica Mathematica Hungarica 65(2)(2012) 205-227.

[2] M. Hell, T. Johansson, A. Maximov, A stream cipher proposal: grain-128, in: Proc. IEEE International Symposium on Information Theory, 2006.

[3] E. Barkan, E. Biham, N. Keller, Instant ciphertext-only cryptanalysis of GSM encrypted communication, Journal of Cryptology 21(3)(2008) 392-429.

[4] M. Dutta, A.K. Singh, A. Kumar, An efficient signcryption scheme based on ECC with forward secrecy and encrypted message authentication, in: Proc. 2013 3rd IEEE International Advance Computing Conference (IACC), 2013.

[5] A. Berent, Advanced encryption standard by example, http://www.networkdls.com/Articles/AESbyExample.pdf.

[6] C.J. Benvenuto, Galois field in cryptography, https://sites.math.washington.edu/~morrow/336_12/papers/juan.pdf.

[7] A.M. Abdullah, R.H.H. Aziz, New approaches to encrypt and decrypt data in image using cryptography and steganography algorithm., International Journal of Computer Applications 143(4)(2016) 11-17.

[8] G. Singh, A study of encryption algorithms (RSA, DES, 3DES and AES) for information security, International Journal of Computer Applications 67(19)(2013) 33-38.

[9] K. Gaj, P. Chodowiec, Fast implementation and fair comparison of the final candidates for advanced encryption standard using field programmable gate arrays, in: Proc. Cryptographers' Track at the RSA Conference, 2001.

[10] R. Padate, A. Patel, Encryption and decryption of text using AES algorithm, International Journal of Emerging Technology and Advanced Engineering 4(5)(2014) 54-59.

[11] M.S. Reddy, Y.A. Babu, Evaluation of microblaze and implementation of AES algorithm using spartan-3E, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering 2(7)(2013) 3341-3347.

[12] N. Selmane, S. Guilley, J.L. Danger, Practical setup time violation attacks on AES, in: Proc. Dependable Computing Conference, 2008.

[13] M. Bafandehkar, S.M. Yasin, R. Mahmod, H.M. Hanapi, Comparison of ECC and RSA algorithm in resource constrained devices, in: Proc. 2013 International Conference on IT Convergence and Security (ICITCS), 2013.

[14] F. Amounas, E.H. El Kinani, ECC encryption and decryption with a data sequence, Applied Mathematical Sciences 6(101)(2012) 5039-5047.