Journal of Software Engineering

ISSN 1819-4311

Academic Journals Inc.

www.academicjournals.com

Journal of Software Engineering 9 (1): 87-95, 2015
ISSN 1819-4311 / DOI: 10.3923/jse.2015.87.95

New Version of AES-ECC Encryption System Based on FPGA in WSNs

Bing Ji, Liejun Wang and Qinghua Yang
Information Science and Engineering, Xinjiang University, Urumqi, China

Corresponding Author: Liejun Wang, Information Science and Engineering, Xinjiang University, Urumqi, China

ABSTRACT

According to the threat of the data transmission on wireless sensor networks, a technique for speeding up point multiplication, an improved AES-ECC hybrid encryption system with cross encrypted keys for secure key exchange is presented. This scheme use AES algorithm to encrypt data, use ECC algorithm to encrypt private key K_{AES} and use SHA-1 algorithm and ECC algorithm to generate digital signature. With rapid advances in VLSI technology, a highly parallel FPGA design is used for their scheme, the computing efficiency of the algorithm is greatly improved. The AES encryption module and multl-scalar multiplication algorithm is also optimized.

Key words: Wireless sensor networks, hybrid encryption, hardware implementation, point multiplication optimization

INTRODUCTION

Wireless Sensor Networks (WSN) are consisting of a large number of low-cost sensor devices which, actually, are used in a wide range of applications such as national defense, environmental monitoring and urban transport, etc. Due to their low-cost and small-size design, the sensor nodes are limited in terms of storage space, energy resources, communication bandwidth and every possible solution that aims to reducing the usage of these resources is then widely sought (Chen *et al*., 2009). Therefore, various types of security threats and challenges are to be faced. Key management is one of the most challenging security issues in large wireless sensor networks which are severely constrained in the resources such as processor, battery and memory, etc. Therefore, the energy-efficient security of WSN will be a crucial issue (Mpitziopoulos *et al*., 2009; Anastasi *et al*., 2009).

The existing symmetric encryption schemes, such as AES, provide a strong security solution (Hoang, 2012) but maintenance of keys is difficult. When asymmetric schemes could be used, maintenance of keys become easier but they provide a lesser degree of security when compared to symmetric encryption schemes. To cope with these shortcomings, the use of a new version of the hybrid encryption system is proposed which is a combination of Advanced Encryption Standard and Elliptical Curve Cryptography (Sutter *et al*., 2013) with cross encrypted keys for secure key exchange.

Further, in order to consider the suitability of a cipher for application to a wireless sensor node which is an energy constrained device, it is most critical to consider the cost of encryption in terms of energy consumption. An FPGA embedded processor system offers many exceptional advantages compared to typical microprocessors such as customization, obsolescence mitigation, component and

cost reduction and hardware acceleration (Garcia *et al*., 2009). Hence, we proposed the FPGA implementation of AES-ECC hybrid encryption system targeted to Wireless Sensor Networks (WSNs).

AES ENCRYPTION MODULE

AES encryption algorithm: AES Rijndael is a block cipher developed by NIST as the Advanced Encryption Standard (AES) replacing DES and published as FIPS 197 in November 2001 to address the threatened key size of Data Encryption Standard (DES). The AES algorithm has become a most popular algorithm of symmetric encryption algorithm, it has a very high security, so it can effectively defense various attacks or linear cryptanalysis (Tran *et al*., 2008). Before applying the algorithm to the data, the block and key sizes must be determined. AES allows for block sizes of 128, 168, 192, 224 and 256 bits. AES allows key sizes of 128, 192 and 256 bits. The standard encryption uses AES-128 where both the block and key size are 128 bits. AES has been designed following the wide trail strategy. In the strategy, the round transformation is divided into different components, each with its own functionality. The encryption algorithm is organized as a set of iterations called round transformations. All round transformations are identical, apart from the final one as shown in Fig. 1.

Optimal design of FPGA-based AES encryption module: The design of AES encryption module is implemented on a chip of FPGA. Round-key generation and round operation adopt the mode of parallel computation and it can support three kinds of key length such as 128, 192 and 256 bit. The proposed scheme has the following properties: A temporary storage is used for the round operation. The processor performs each round operation while the round-key of the next round is generated. So, round-Key requires no extra storage. In this way, it not only saves the on-chip resources but also solves the delay problem caused by reading the key and it improves the

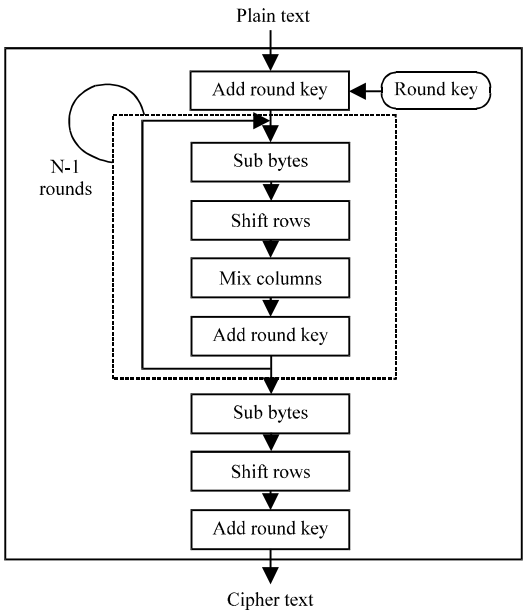

Fig. 1: AES encryption

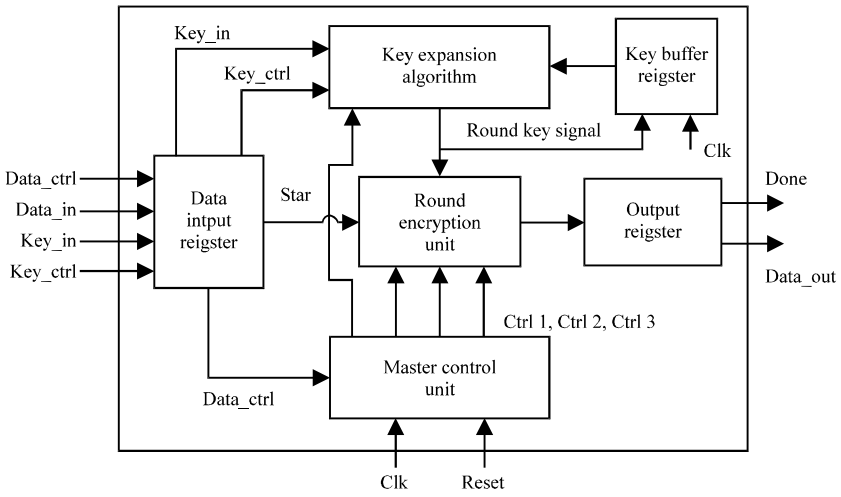

Fig. 2: AES data encryption module

clock frequency and the throughput of the system and reduced the memory requirements of the round key. Therefore, it enhances the security of wireless sensor network and service life of wireless sensor network nodes. The overall structure design as shown in Fig. 2.

AES encryption module consists of Data Input Register, Key expansion algorithm, Key buffer Register, Round encryption Unit, Master control Unit and Output Registers. Date_in is the data input signal, Date_ctrl is the data load signal, Key_in is the key input signal, Key_ctrl key-load signal, Ctrl1, Ctrl2, Ctrl3 is the control signal, Star is the key enable signal, Data_out is the data output signal and Done is completion signal.

ECC ENCRYPTION SYSTEM

ECC encryption algorithm theoretical model: According to the IEEE P1363 draft standard and the design of ECC basic encryption communication model (Sanjeev and Jose, 2010; Lin and Hsu, 2011), the user A select a specific elliptic curve is expressed as $E_p(a, b)$ and then select the point G on the curve; randomly generating private key k then, the public key is Q = kG, Q and G is a point on the elliptic curve as $E_p(a, b)$. Then, broadcast the characteristic information of elliptic curve parameters, Q and G. The plaintext which will be sent to A was encoded to a bit M of the curve by the user B then, generates a random number r. And generates C1, C2 at the same time then B sends them to A. According to its unique private key k, user A used it to decrypt the C1 and C2, then A get the plaintext M. In the process of communication, if the hacker H has monitored the five packets of information which were transmitted across the network, H was unable to get the private key k, thereby, was unable to get the plaintext M from C1 and C2 as shown in Fig. 3.

Classical algorithm of point multiplication operation: For elliptic curve cryptography, point multiplication KP of the group operation layer is the most time-consuming operation, is also the most frequently invoked operation. It plays a decisive role in the efficiency of the system (Elbirt *et al*., 2001). The binary algorithm is the basic algorithm of KP. K is generaly used for binary representation as follows:

$$K = \sum_{i=1}^{n} 2^i a_i = a_1 + 2a_2 + \cdots 2^{n-1} a_{n-1} + 2^n a_n \quad (1)$$

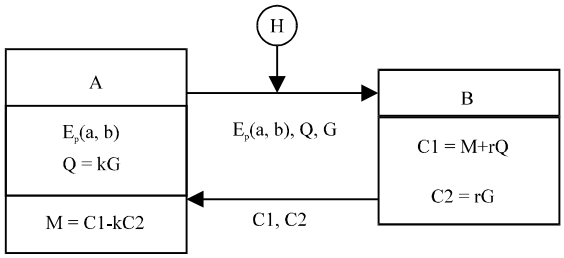

Fig. 3: ECC communication model

With this representation, $a_i \epsilon \{0, 1\}$, the calculation of public key Q equation can be expressed as:

$$Q = kP = \left(\sum_{i=1}^{n} 2^i a_i\right) P = 2[...[2(2a_{n-1}P + a_{n-2}P) + a_{n-3}P] + ... + a_1P] + a_0P \quad (2)$$

According to Eq. 2, with the iterative algorithm, it actually needs N times point doubling operation and M-1 times point addition operation (Jarvinen and Skytta, 2009). M is the number of 1's in sequences a. Using the 100 bit (large integer multiplication) as an example, the classical point multiplication algorithm needs 100 times point doubling operation and 49.5 (average) times point addition operation.

Improved algorithm of point multiplication operation: In this study, the fast algorithm performs the following steps:

Step 1: K can be conveniently expressed in the following binary form:

$$K = (a_i a_{i-1} \ldots a_j \ldots a_1)$$

For example, 100 can be expressed as 1100100. With this representation, $a_i \epsilon \{0, 1\}$, i>1, $i = [\log_2^K]+1$

Step 2: Remove the top bit a_i of $(a_i a_{i-1} \ldots a_j \ldots a_1)$, then develop into $(a_{i-1} \ldots a_j \ldots a_1)$

Step 3: Then perform this operation:

Algorithm 1: KP algorithm

Input: k (in its binary form), P

Output: Q = kp

1. n = i-1
2. While (n>1) do
   - 2.1 If $a_n = 1$, then Q <= 2p+p; else Q <= 2p
   - 2.2 n <= n-1.
   - 2.3 p <= Q.
3. Return Q

Verify this algorithm as follows:

For example, K = 30 = 11110. According to the classical algorithm, the number of point additions required is 30. According to this algorithm, the number of point additions required is only 11, if i = n, the number of point additions required is on average $5/3[\log_2^n]<n$ and the number of point additions required is on maximum $2[\log_2^n]$. As compared with classical scheme, the proposed algorithm has efficiency advantages.

AES-ECC HYBRID ENCRYPTION MODEL IMPLEMENTATION

ECC signature and verification process: Using the Hash function (De Canniere and Rechberger, 2006) which was selected to process messages first, Signature Scheme Based on Elliptic Curve follows: The signer A has a private key d and a public key Q, making known to the public the public key Q, the selected Hash and other necessary information; A will send B signature-messages, B can verify the legitimacy of signatures based on public news, n is the order of point G. The signature generation process is as shown in Fig. 4.

To sign a message m, the sender performs the following steps:

Step 1: $k\epsilon[1, n-1]$, $u = [k]G = (x_1, y_1)$; K is a random selection
Step 2: Compute $r = x_1 \bmod n$, if r = 0, return to step 1
Step 3: According sk = sha-1(m)+dr mod n, we would calculate s. If s = 0, return to step 1
Step 4: (u, s) is the signature information, then, we sent m and (u, s) to B as (m(u, s))

To verify the signature, the receiver performs the following steps:

Step 1: If $s\notin[1.n-1]$, the signature is forged, reject the signature
Step 2: Verify the equation:

$$[s]u = [\text{sha-1}(m)]G+[r]Q$$

Validation passes if and only if equality holds, otherwise, the signature is forged, reject the signature proving the equation:

- Because $s = k^{-1}(\text{sha-1}(m)+dr)$ and $u = [k]G$
- Compute $[s]u = [k^{-1}(\text{sha-1}(m)+dr)k]G$

$$= [\text{sha-1}(m)]G+[dr]G$$

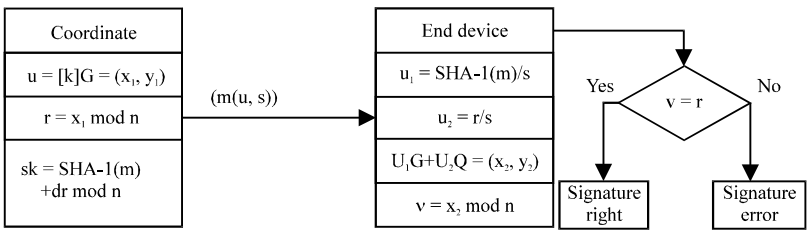

Fig. 4: Digital signature software process

= [sha-1(m)]G+[r]Q

Overall program module: The existing framework of the hybrid encryption scheme allows for only one way key encapsulation. That is, the AES key is protected by encrypting it with the ECC key. This necessitates periodic updation of AES key and ECC public key without increase in complexity and also cross encryption of AES and ECC keys with one another. The improved AES-ECC Hybrid encryption scheme is shown in Fig. 5.

The steps involved in the improved AES-ECC Hybrid encryption scheme is as follows:

Step 1: Data is collected by the sensor data acquisition system
Step 2: Using SHA-1 function to generate the data summary
Step 3: Using the sender's private key K and ECC digital signature module to generate Digital Signatures
Step 4: According to AES encryption module (the private key is KAES), encrypting digital signature and encrypting data which need to be sent. Then, getting data-ciphertext and signature-ciphertext
Step 5: Encrypting the private key KAES by ECC encryption module, then, generating key-ciphertext
Step 6: Packing all ciphertext and sending it by means of wireless sensor networks
Step 7: The sender upload that ciphertext to the internet by the sink node, the users can use the mobile terminal to receive data
Step 8: When the receiver receives the ciphertext, receiver uses his private key to decrypt the key KAES, then, decrypting the data-ciphertext and signature-ciphertext by KAES. Using the

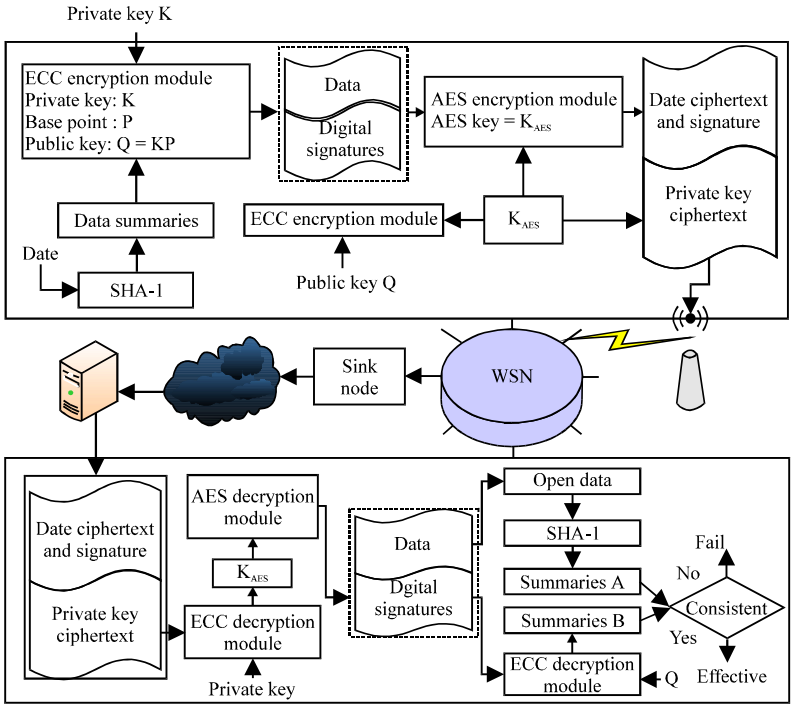

Fig. 5: AES-ECC hybrid encryption system

sender's public key to verify the signature and get the summary B; then we can get the summary A by using SHA-1 algorithm. Comparing summary A with summary B, if they are the same, then the data is valid and available; otherwise, it represents invalid data

RESULTS AND DISCUSSION

Verilog hardware description language is used for hierarchical top-down programming. Altera's development tools QUARTUS II 9.0 are used for compiler and simulation. Finally, this scheme is implemented on the EP4CE115 chip of Cyclone IV series. Cyclone IV series (the development of power is less than 1.5 watts) provides nearly 150000 logic units and 8 integrated 3.125 Gbps transceivers. It is ideally suited for wireless sensor networks in the field of low-cost, small form factor packaging applications (Fig. 6).

Comparing this study with the similar schemes, as shown in Table 1, the results show that maximum frequency can reach 60 MHz of encryption and throughput is 985 Mbps. It uses only 846 (LE) and resource consumption rate is only 0.0066% (Fig. 7).

For ease of presentation, the improved ECC modules is compared with the similar functions of the software, as shown in Table 2.

The improved ECC and TinyECC (Liu and Ning, 2008) are written into two nodes (FPGA) with the same performance. And the code is based on 160 bit standard elliptic curve parameters secp160r1 (SECG established). Computing the same operation 100 times, then taking the average as shown in Table 2.

Table 1: AES scheme comparison

Scheme	Operation mode	Frequency (MHZ)	Throughput (Mbps)	LE	Key length (bit)
This study	ECB	60.0	985	846	128
Wang (2008)	ECB	29.4	376	3235	128
Shen (2006)	ECB	67.0	779	881	128
Elbirt *et al.* (2001)	ECB	40.0	487	857	128

Table 2: Experimental results compare

Scheme	Execution time (sec) (sign)	Execution time (sec) (verify)	Fixed point scalar multiplication (sec)	RAM/B (k)
This study	2.28	3.21	1.74	2.1
Tinyecc (Liu and Ning, 2008)	7.71	14.82	7.32	1.7

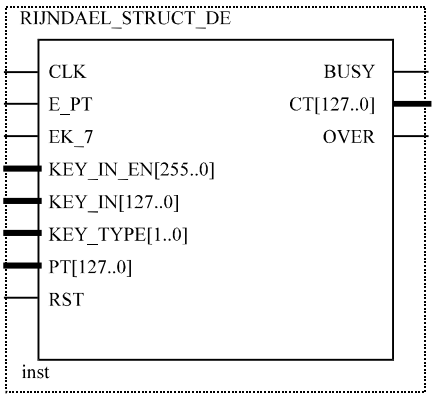

Fig. 6: AES STRUCT DE

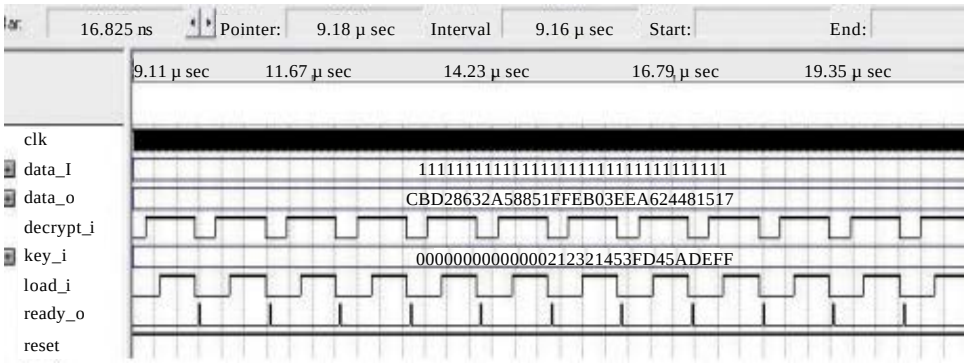

Fig. 7: Data encryption simulation

Fixed point scalar multiplication: The improved ECC algorithm improves nearly 3.5 times faster, results show that it is different with that mentioned above. Because the optimization mechanism of sliding window and optimization of pre computed are added to the classical algorithm, it improved the operation effect of certain.

Digital signature verification: It is a collectivity of a fixed bit multiplication (U_1P) and a random point multiplication (U_1Q). For the random point multiplication, it is unable to calculate the budget table by using pre. Thus, effect cannot be improved significantly.

RAM of code: The improved code occupies RAM space slightly larger than Tinyecc, this is because the code is needed to open up a static memory space in memory and the static memory is used to place the budget table. However, 0.4 k RAM space has exchanged a significant speed boost, it shows that the improved algorithm has higher application value.

CONCLUSION

According to the characteristics of the wireless sensor network, an improved AES-ECC hybrid encryption system is proposed. This design has good flexibility and versatility. And we optimized the design of ECC multiplying unit. It improved the speed of the digital signature generation and authentication. Fully meeting the requirements of wireless sensor networks for the stability, power consumption and processing speed. Currently, working is done on high performance (increase throughput, reduce logic unit occupancy) AES encryption module and optimization problems of random point multiplication of ECC cryptographic module.

As compared with several schemes (Tang and Wang, 2013; Rafik and Mohammed, 2013; Zhang, 2007), the proposed scheme has the following properties (1) It provide better security requirements with relatively low-resource request, (2) It is simple to execute the key management, (3) It is secure against some potential attacks and (4) It is fast and easy to generate digital signatures and verify the digital signatures.

ACKNOWLEDGMENT

This study is supported by the Natural Science Foundation of XJ autonomous region under grant No. 2013211A012. The authors would like to thank the anonymous reviewers for their constructive comments that helped to improve the quality of this study.

REFERENCES

Anastasi, G., M. Conti, M. di Francesco and A. Passarella, 2009. Energy conservation in wireless sensor networks: A survey. Ad Hoc Networks, 7: 537-568.

Chen, X., K. Makki, K. Yen and N. Pissinou, 2009. Sensor network security: A survey. IEEE Commun. Surveys Tutorials, 11: 52-73.

De Canniere, C. and C. Rechberger, 2006. Finding SHA-1 characteristics: General results and applications. Proceedings of the 12th International Conference on the Theory and Application of Cryptology and Information Security, December 3-7, 2006, Shanghai, China, pp: 1-20.

Elbirt, A.J., W. Yip, B. Chetwynd and C. Paar, 2001. An FPGA-based performance evaluation of the AES block cipher candidate algorithm finalists. IEEE Trans. Very Large Scale Integr. Syst., 9: 545-557.

Garcia, R., A. Gordon-Ross and A.D. George, 2009. Exploiting partially reconfigurable fpgas for situation-based reconfiguration in wireless sensor networks. Proceedings of the 17th IEEE Symposium on Field Programmable Custom Computing Machines, April 5-7, 2009, Napa, CA., pp: 243-246.

Hoang, T., 2012. An efficient fpga implementation of the advanced encryption standard algorithm. Proceedings of the IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future, February 27-March 1, 2012, Ho Chi Minh City, pp: 1-4.

Jarvinen, K. and J. Skytta, 2009. Fast point multiplication on Koblitz curves: Parallelization method and implementations. Microprocess. Microsyst., 33: 106-116.

Lin, Y.L. and C.L. Hsu, 2011. Secure key management scheme for dynamic hierarchical access control based on ECC. J. Syst. Software, 84: 679-685.

Liu, A. and P. Ning, 2008. TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks. Proceedings of the 7th International Conference on Information Processing in Sensor Networks, April 22-24, 2008, IEEE Xplore, London, pp: 245-256.

Mpitziopoulos, A., D. Gavalas, C. Konstantopoulos and G. Pantziou, 2009. A survey on jamming attacks and countermeasures in WSNs. IEEE Commun. Surveys Tutorials, 11: 42-56.

Rafik, M.B.O. and F. Mohammed, 2013. The impact of ECC?s scalar multiplication on wireless sensor networks. Proceedings of the 11th International Symposium on Programming and Systems, April 22-24, 2013, Algiers, pp: 17-23.

Sanjeev, C. and G.J.A. Jose, 2010. Elliptic curve cryptography enabled security wireless communication. Int. J. Comput. Sci. Eng., 2: 2187-2189.

Shen, Q.F., 2006. FPGA optimization implementation for AES algorithm. J. Xian Univ. Technol., 22: 203-206.

Sutter, G.D., J. Deschamps and J.L. Imana, 2013. Efficient elliptic curve point multiplication using digit-serial binary field operations. IEEE Trans. Ind. Electron., 60: 217-225.

Tang, J. and L. Wang, 2013. An improved rijndael encryption algorithm based on niosii. Inf. Technol. J., 12: 1434-1438.

Tran, M.T., D.K. Bui and A.D. Duong, 2008. Gray s-box for advanced encryption standard. Proceedings of the International Conference on Computational Intelligence and Security, December 13-17, 2008, Suzhou, pp: 253-258.

Wang, J.Y., 2008. Reconfigurable design for encryption/decryption of AES based on FPGA. Comput. Eng., 34: 163-164, 167.

Zhang, D.X., 2007. Design and implementation of AES algorithm based on FPGA. J. Univ. Sci. Technol. China, 37: 1461-1465.