International Journal of Future Generation Communication and Networking
Vol. 9, No. 2 (2016), pp. 73-82
http://dx.doi.org/10.14257/ijfgcn.2016.9.2.08

An Improved ECDSA Scheme for Wireless Sensor Network

Hong Zhong, Rongwen Zhao, Jie Cui*, Xinghe Jiang and Jing Gao

School of Computer Science and Technology, Anhui University, Hefei, 230039, China
**cuijie@mail.ustc.edu.cn*

Abstract

With the widely application of the Wireless Sensor Network, it is particularly significant to assure a secure communication mechanism between the nodes. In order to meet the needs of sensor nodes on low power consumption and less resource, lightweight cryptographic algorithm designed well is the key to constructing a riskless WSN scheme. In this paper, we propose an improved elliptic curve cryptography digital signature scheme by means of optimizing the multiplicative inverse module of ECDSA, based on ECC lightweight cryptographic algorithm. Moreover, we implement ECDSA scheme improved on Micaz, which is a kind of sensor network platform. And then, experimental results demonstrate that the performance of improved scheme is superior to the previous, not only on the speed but also on the efficiency, under the same experimental environment and encryption intensity. In a word, our scheme has stronger practicability.

Keywords: *WSN, Lightweight Cryptographic Algorithm, ECC, Digital Signature, ECDSA*

1. Introduction

In recent years, with the continuous development of the Internet of Things, Wireless Sensor Network used as the basis of supporting joint, which has been intensively applied with the characteristics of its miniaturization, low energy consumption and high performance [1]. Since Wireless Sensor Network is an open architecture system, which makes it vulnerable to outside attack. Correspondingly, the security can be compromised at ease. What is worse, invaders can falsify and fake message packets conveniently during the process of transmission. In this case, it is impossible to confirm the integrity and accuracy of the information. In order to ensure the security of information transmission, it is necessary to improve the security of Wireless Sensor Network [2].

In Wireless Sensor Network, in order to guarantee the integrity of information transmission, verify the identity authenticity of sender and prevent repudiation of transaction, we need to add a digital signature into the packet sent by the source sensor node. Wireless sensor nodes in the network are limited in computing power, battery capacity and storage capacity. Nevertheless, the large amount of data and more energy are required when digital signature generated. Consequently, a reasonable choice of digital signature algorithm, which can enhance the security of Wireless Sensor Network greatly and extend network lifetime effectively, has become an essential issue.

The implementation on Wireless Sensor Network based on asymmetric encryption system is eminently achievable, such as RSA and ECC. The comparison [3] of ECC and RSA is shown in Table 1. The main advantage of RSA is the long key and it is a highly reliable and effective cryptographic algorithm. In terms of the comparison between ECC and the traditional encryption algorithms, such as RSA, ELGmal, and so on, we can conclude that ECC has certain merits including short key length and low computing cost, on condition of the identical encryption intensity [4].

ISSN: 2233-7857 IJFGCN

Table 1. The Comparison of ECC and RSA

The length of RSA key/bit	The length of ECC key/bit	The ratio of ECC to RSA
1024	163	6.3 : 1
2048	233	8.8 : 1
3072	283	10.8: 1
7689	409	18.8 : 1

At present, with the purpose of solving these problems mentioned above, there are several kinds of digital signature schemes that have been widely applied to network communications. RSA algorithm is one of the most influential digital signature algorithms. However, it is not suitable for use on Wireless Sensor Network because of its too long key and signature [5]. By comparing the different digital signature schemes, ECDSA has become the first choice of the Wireless Sensor Network digital signature algorithm with advantages of the short length of signature and authentication time. Unfortunately, the traditional ECDSA has more reverse modules, which seriously affect the efficiency of the signature generation and verification [6].

2. Preliminary Background

2.1 Elliptic Curve Cryptography

Elliptic curve cryptography is defined on finite field. Its form is similar with Weierstrass equation:

$$y^2 + b_1xy + b_2y = x^3 + a_1x^2 + a_2x + a_3 \qquad (1)$$

In equation (1), a_1, a_2, a_3, b_1, $b_2 \in GF(q)$. $GF(q)$ can assume to real number field R or rational number field. When ECC is applied to encryption process, we put more attention on limited form of elliptic curve so the above-mentioned equation can be simplified to equation (2).

$$y^2 = (x^3 + ay^2 + b) \bmod p \qquad (2)$$

where a , b , $p \in GF(p)$, p is a prime number and satisfies the equation $4a^3 + 27b^2 \neq 0 \bmod p$. This assures that this curve is nonsingular [7].

The definition of Elliptic Curve Discrete logarithm Problem (ECDLP): Consider two points， P and Q lying on an elliptic curve (EC), such that $Q = k \times P$, where k is a scalar. Given these two points, it is infeasible to obtain the value of k , assuming it is very large and k is the discrete logarithm of Q to the base P . No sub-exponential algorithm to solve the ECDLP has been reported so far [8].

2.2 ECC Encryption and Decryption

Suppose that Alice and Boy want to communicate with each other. They should follow the next steps.

Step 1. Alice selects a , b , p to create a elliptic curve $E_p(a,b)$ and choose a point M as based point.

Step 2. Alice selects a private key y and gets public key $Y = y \times M$. Then Alice sends $E_p(a,b), Y$ and M to Bob.

Step 3. Bob receives the message, encode message L to a point N in elliptic curve $E_p(a,b)$. Next Bob chooses a private key $r < n$.

Step 4. Bob computes $C1 = N + rY$, $C2 = rM$.

Step 5. Bob sends $C1, C2$ to Alice.

After receiving Bob's message, Alice decrypts the message by computing $N = C1 - yC2$ to obtain the plaintext.

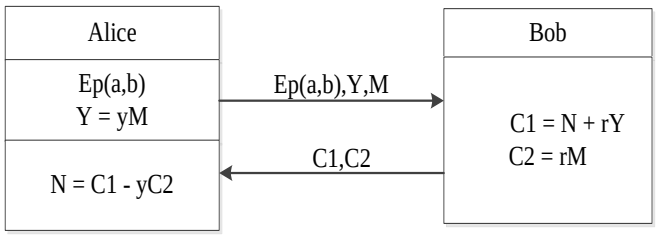


Figure 1. ECC Encryption/Decryption Schematic

One eavesdropper Helen may listen above communications. He can see $E_p(a,b)$, Y, M, $C1$ and $C2$. It is impossible to get y or r through the above information. So we can believe this communication is safe.

2.3 TinyECC

TinyECC is a software package providing ECC-based PKC operations that can be flexibly configured and integrated into sensor network applications. TinyECC has many optimization techniques, such as Barrett Reduction, Hybrid, Multiplication and Hybrid Squaring, Sliding Window for Scalar Mulitiplications [9].It provides many interfaces:

1. Interface NN, which provides the basic operations of large integer number.
2. Interface ECC, which provides some initialization operations, such as initialize of an elliptic curve.
3. Interface ECDH, which implements a key exchange protocol.
4. Interface ECDSA, which implements a digital signature scheme.
5. Interface ECIES, which implement s a public key encryption scheme.

3. Elliptic Curve Digital Signature Algorithm (ECDSA)

In real life, three cases occur most commonly within the process of communication:

(1) Message has been tampered.
(2) The sender deny sending the message.
(3) The receiver fake the message.

With the aim to solve problems mentioned, we need to sign a digital signature, as real world, to protect the benefits of each other.

Digital signatures are some big integers, such as a long string of 1024 bits. It can be sent by sender, others cannot fake it. Once receiver obtain the signature, sender cannot deny transmitting it. If others fake the signature, receiver can compare it by gaining the

message integrity.

3.1 Signature Generation

Alice negotiates with Bob and selects a point G. Alice, the sender, chooses parameters a, b, p, n to constitute a elliptic curve $E_p(a,b)$ and then Alice obtains a private key d and a public key $Q = d \times G$.

Signer Alice generates the signature for the message m, as follows[10].

Step 1: Select a random key k in[1, n - 1].

Step 2: Calculate a curve point $k \times G = (x_1, y_1)$.

Step 3: Calculate $r = x_1 mod\ n$. If $r = 0$, go back to step 1.

Step 4: Calculate $e = SHA-1(m)$.

Step 5: Calculate $s = k^{-1}(e + dr)\ mod\ n$. If $s = 0$ then go back to step 1.

Step 6: Send the digital signature (r, s) and m.

3.2 Signature Verification

To verify Alice's message and signature (r, s), Bob should do the followings.

Step 1: Verify that r and s are integers in[1, n - 1]. If not, the signature in invalid.

Step 2: Calculate $e = SHA-1(m)$.

Step 3: Calculate $w = s^{-1} mod\ n$.

Step 4: Calculate $u_1 = (e \times w)\ mod\ n$, $u_2 = (r \times w)\ mod\ n$.

Step 5: Calculate the curve point $X = u_1 G + u_2 Q = (x_1, y_1)$.

Step 6: If $X = O$, the signature is invalid, and refuse it. Else calculate $v = x_1 mod\ n$.

Step 7: Bob will accept the signature if and only if $v = r$.

3.3 Mathematical Proof

Suppose that the message transmitted correctly, then

$$\begin{aligned} X &= u_1 G + u_2 Q \\ &= (e \times w + r \times w \times d) G\ mod\ n \\ &= (e \times w) G\ mod\ n + (r \times w) Q\ mod\ n \end{aligned} \tag{3}$$

Due to $s = k^{-1}(e + r \times d)\ mod\ n$, So substituting $k = s^{-1}(e + r \times d)$ into (3) can get $X = s-1(e + r \times d)G = k \times G = (x_1, y_1)$. In this case $v = x_1\ mod\ n = r$, message can be verified successfully.

4. Our Improved Scheme

Through the analysis of the signature generation phase, it is obvious that *ECDSA* takes advantage of *ECC*, which has small key size and high security. Unfortunately, this scheme need twice modular multiplicative inverse and the time cost of one modular multiplicative inverse is ten times the same as multiplication, which is very time-consuming for elliptic curve discrete logarithm system [11]. In order to adapt the

wireless sensor nodes characterized by resource limited, we intend to improve the original algorithm.

4.1 Signature Generation

When Alice sends the message to Bob, and then obtains a digital signature (r, s) generated by following steps.

Step 1: Select a random k in [1, n - 1].

Step 2: Calculate a curve point $k \times G = (x_1, y_1)$.

Step 3: Calculate $r = x_1 \ mod \ n$. If $r = 0$, then go back to step 1.

Step 4: Calculate $e = SHA-1(m)$.

Step 5: Calculate $s = (e + k + rd) \ mod \ n$. If $s = 0$, then go back to step 1.

Step 6: Send the message m and digital signature (r, s).

4.2 Signature Verification

From the steps of digital signature generation, we can design the key step of verification by Backward Recurrence Algorithm:

$$\begin{aligned} X &= k \times G \\ &= (s - e - r \times d)G \\ &= (s - e)G - r \times d \times G \\ &= (s - e)G - r \times Q \\ &= (x_1, \ y_1) \end{aligned}$$

Bob should do the followings to verify the digital signature:

Step 1: Verify that r and s are integers in [1, n - 1].If not, the signature in invalid.

Step 2: Calculate $e = SHA-1(m)$.

Step 3: Calculate $w = (s - e) \ mod \ n$.

Step 4: Calculate a curve $X = w \times G - r \times Q = (x_1, y_1)$.

Step 5: If $X = O$, the digital signature is invalid and refuse to it, else calculate $v = x_1 \ mod \ n$.

Step 6: Bob will accept the digital signature if and only if $v = r$.

From the verification steps above, we can see that digital signature verification is the reverse of digital signature generation.

5. The Algorithm Implementation and Performance Test

5.1 The Experimental Environment

The experimental environment is configured as follows: Ubuntu14.04, TinyOS 2.1.2, JDK 1.6, using Crossbow's Micaz node to establish a Wireless Sensor Network test environment. The improved algorithm can be implemented not only by calling the relevant components and interfaces in TinyOS, but also by utilizing TinyECC. Figure 2 describes the relationship of modules called in the experiment in detail.

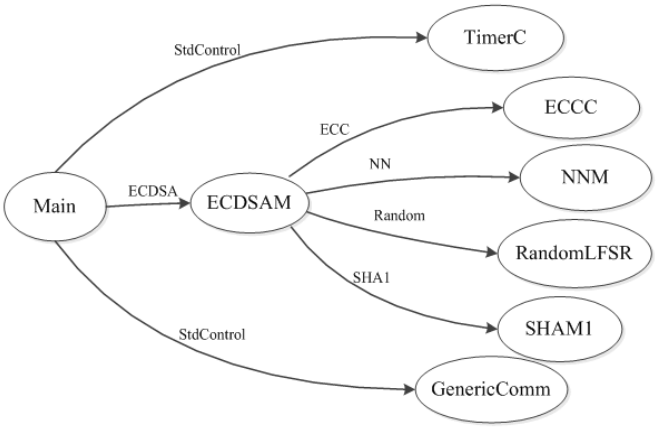


Figure 2.The Diagram of Modules Called

In TinyOS [13], component Main is the entry of all programs. The interface StdControl provided by Main implements some hardware initialization. Whereas, component Timer, Leds and GenericComm accomplish initialization by implementing interface StdControl. What is more, component ECDSA is the core of the program, whose main tasks include: (1) call component Timer to achieve the trigger time, (2) call component Leds to trigger indicator leds, (3) call component GenericComm provided by TinyOS to implement data transmition, (4) call component Random to generate random numbers, (5) call component ECC and component NN for basic computing and rounded operations on elliptic curves.

5.2 Cost Analysis

Sensor nodes are extremely limited in terms of energy supply, storage space and other resources. We suppose that the consuming time of plus operation, multiply operation, modular arithmetic and Hash operation were E1, E2, E3 and E4 respectively. Assuming the cost of ellipse curve initialization is E, therefore, the cost comparison of the improved ECDSA with the original ECDSA listed in Table 2.

Table 2.The Cost Comparison of Our Scheme with the ECDSA Scheme

Schemes	ECDSA Scheme	Our Scheme
Generation	E+n(E1+3E2+2E3+E4)	E+n(2E1+2E2+E3+E4)
Verification	E+n(E1+4E2+3E3+E4)	E+n(2E1+3E2+2E3+E4)

Elliptic curves system over finite fields, which is very time-consuming. Inverse operation is much slower than multiplication. Knuth [12] pointed that using Extended Euclidean algorithm to achieve modular inverse also need to complete the $0.843*\log_2(n)+1.47$ times division operation averagely. The improved signature scheme utilize plus operation instead of using time-consuming modular inverse, which can effectively reduce the computational complexity.

5.3 Security Analysis

The difficulties associated with the attacks are based on the solution of the $ECDLP$, and the security resulted from such problems is still sufficient under the reasonable computational complexity.

(1) If an attacker got the public key Q, who wants to obtain its private key d

through Q. This is one problem for solving the elliptic curve discrete logarithm, so this approach is not feasible.

(2) If an attacker obtained $m, (r, s)$, who wants to obtain the private key d by solving $s = (SHA-1(m) + k + rd) \bmod n$, $d = ((s - SHA-1(m)) - k) r-1 \bmod n$. Because k is selected by the signer randomly, so this approach is also not feasible.

(3) If the attacker got $m, (r, s)$, although the attacker cannot acquire the private key d of the signer, but want to fake signature by the equation $s = (e + k + rd) \bmod n$. Only have to generate k_1 randomly, to find s_1 by r_1.When authentication only by the equation $X = (s_1 - e - r_1 d) G \bmod n = k_1 G != (x_1, y_1)$.The attacker even though can avoid solving d, however, because of random k, so forgery is not feasible.

5.4 Efficiency Analysis

(1) Implement inverse-free operation. In the usual elliptic curve encryption or signature process, inverse operation is the main computational burden. The improved scheme eliminates the inverse operation, and the experimental results show that the above scheme can correctly judge the validity of the signature, and ease the required computing. Compared to other elliptic curve signature schemes, the system reduce time consumption and improve the efficiency of signature.

(2) By avoiding twice modular inversion, which can effectively reduce the generation time of a digital signature and digital signature verification time. By the following average time in the first nine rounds can draw a conclusion that digital signature generation time is reduced by 7.3%, the digital signature verification time reduced by 6.1%.

5.5 The Experimental Results

In this experiment, 160 bits ECC algorithm key adopted for encryption and decryption, which utilizes the actual sensor node to implement elliptic curve encryption algorithm. We accomplished encryption and decryption in the first nine rounds. Experimental results are shown in Figure 3, Figure 4 below.

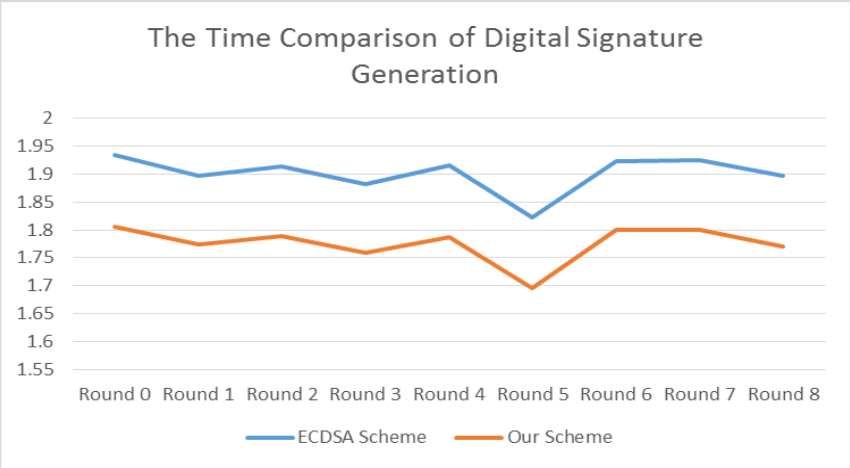


Figure 3. The Time Comparison of Digital Signature Generation

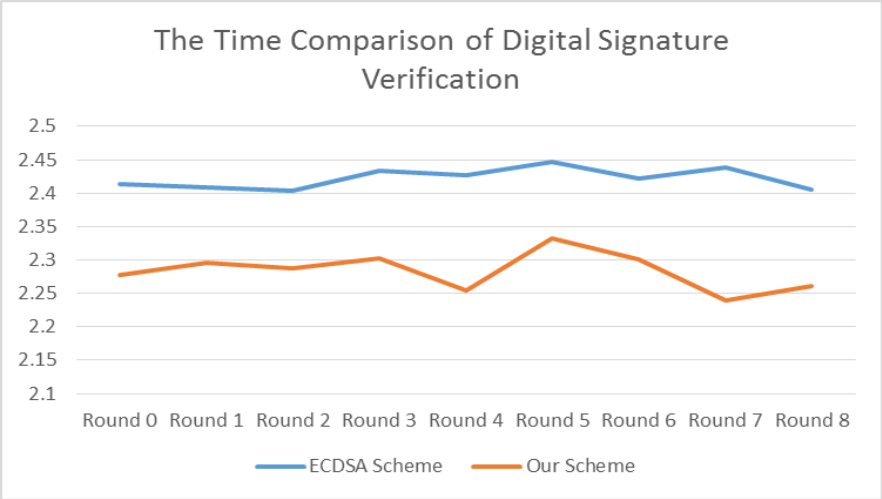


Figure 4. The Time Comparison of Digital Signature Verification

The average time of signature generation and verification are on the decrease, which can be seen through the first nine rounds. Generating an elliptic curve and choosing a base point, base points are required in the system initialization phase, in this case, as the rounds increasing, the average time consumption of signature generation and signature verification will decline. The improved algorithm has a certain improvement efficiently in the signature generation and verification, which can be seen through comparison in the figures.

6. Conclusion

In this paper, we introduce the general principle of elliptic curve encryption algorithm in detail, conduct the thorough research on ECDSA, and find out the existing problems with low efficiency of ECDSA algorithm. By reducing unnecessary modular inverse operation, we improve the efficiency of the digital signature greatly. We implement our scheme on Micaz and TinyOS, and the experimental results show that the new scheme can be used more efficiently in the wireless sensor nodes. Our next step is to study the blind signature algorithm utilized in Wireless Sensor Network applications, and propose a new blind signature scheme based on ECC.

Acknowledgments

The work was supported by the National Natural Science Foundation of China (No. 61173188, No. 61173187), the Educational Commission of Anhui Province, China (No. KJ2013A017), The Natural Science Foundation of Anhui Province (No. 1508085QF132), the Research Fund for the Doctoral Program of Higher Education (No. 20133401110004), the Science and Technology Project of Anhui Province (No. 1401b042015), the Tender Project of the Center of Information Support &Assurance Technology of Anhui University(No. ADXXBZ2014-7), and the Doctoral Research Start-up Funds Project of Anhui University. The authors are very grateful to the anonymous referees for their detailed comments and suggestions regarding this paper.

References

[1] I. Butun, S.D. Morgera and R. Sankar , “A Survey of Intrusion Detection Systems in Wireless Sensor Networks”, Communications Surveys & Tutorials, vol. 16, no. 1, **(2014)**, pp. 266-282.
[2] D. He, C. Chen and S. Chan, “Security Analysis and Improvement of a Secure and Distributed Reprogramming Protocol for Wireless Sensor Networks”, IEEE Transactions on Industrial Electronics,

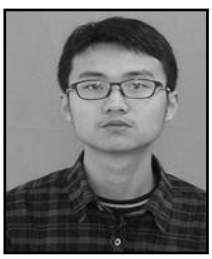
vol. 60, no. 11, **(2013)**, pp. 5348-5354.
[3] K.N. Bidkar, "Energy Analysis of Algorithms in Public Key Cryptography of WSN", International Journal of Advance Research in Computer Sciencer and Management Studies, vol. 3, no. 3, **(2015)**, pp. 190-197.
[4] X.Wei and P. Zhang, "Research on Improved ECC Algorithm in Network and Information Security", International Journal of Security and Its Applications, vol. 9, no. 2, **(2015)**, pp. 29-36.
[5] K.K. Gola, B. Gupta and Z. Iqbal, "Modified RSA Digital Signature Scheme for Data Confidentiality", International Journal of Computer Applications, vol. 106, no. 13, **(2014)**, pp. 13-16.
[6] A. Khalique, K. Singh and S. Sood, "Implementation of elliptic curve digital signature algorithm", International Journal of Computer Applications, vol. 2, no. 2, **(2010)**, pp. 21-27.
[7] Y.F. Chung, K.H. Huan and F. Lai, "ID-based digital signature scheme on the elliptic curve cryptosystem", Computer Standards & Interfaces, vol. 29, no. 6, **(2007)**, pp. 601-604.
[8] Y.J. Huang, C. Petit and N. Shinohara, "Improvement of FPPR method to solve ECDLP", Pacific Journal of Mathematics for Industry, vol. 7, no. 1, **(2015)**, pp. 1-9.
[9] A. Liu and P. Ning, "TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks", Information Processing in Sensor Networks, IPSN'08, **(2008)**, 245-256.
[10] S. Karati, A. Das and D. Roychowdhury, "New algorithms for batch verification of standard ECDSA signatures", Journal of Cryptographic Engineering, vol. 4, no. 4, **(2014)**, pp. 237-258.
[11] D. Hankerson, A.J. Menezes and S. Vanstone, "Guide to elliptic curve cryptography", Computing Reviews, vol. 46, no. 1, **(2005)**, p. 13.
[12] D.E. Knuth, "The Art of Computer Programming", Addison-Wesley, Boston, **(1998)**.
[13] M.O. Farooq and T. Kunz, "Operating systems for wireless sensor networks: A survey", Sensors, vol. 11, no. 6, **(2011)**, pp. 5900-5930.

Authors

Hong Zhong (1965), She is a professor (from 2009), PhD supervisor and dean of the School of Computer Science and Technology, Anhui University, China. She received her PhD degree from University of Science and Technology of China in 2005. Her research interests cover network and information security.

Rongwen Zhao, He was born in 1993. Currently he is studying for his B.S. degree in Network Engineering from Anhui University. His research interests include network and information security.

Jie Cui, He was born in 1980, is now an associate professor in the School of Computer Science and Technology, Anhui University. He received PhD degree in University of Science and Technology of China in 2012. He has published 20 papers. His research interests include network and information security.



Xinghe Jiang, He was born in 1993. Currently he is studying for his B.S. degree in Network Engineering from Anhui University. His research interests include network and information security.

Jing Gao, he was born in 1992. Currently he is studying for his B.S. degree in Software Engineering from Anhui University. His research interests include network and information security.

