New digital signature algorithm based on ECC and its application in bitcoin and IoT

Shuai Xiao and Han Wang

Key Laboratory for Network and Information Security of PAP,
Engineering University of PAP,
Xi'an, Shaanxi, 710086, China
Email: xs18829581835@163.com
Email: aca_wang@163.com

Jindan Zhang*

The School of Electronics and Information,
Xianyang Vocational Technical College,
Xianyang, Shaanxi, 712000, China
Email: zhangjindan83@163.com
*Corresponding author

Abstract: Elliptic curve digital signature algorithm (ECDSA) is the simulation of digital signature algorithm (DSA) algorithm on elliptic curve. Compared with DSA, ECDSA has higher security and is the only widely accepted ECDSA, which has been adopted by many standardisation organisations. Based on the study of the original ECDSA scheme, this paper attempts to propose a new improved scheme. The proposed scheme has one main improvement. That is, considering that the original scheme has a finite field inversion process in the signature equation, the time-consuming inversion operation is completely avoided in the design. The proposed scheme has faster computation speed and reduces the ratio of verifying signature to signature generation time. The algorithm has certain significance for improving the efficiency of elliptic curve cryptography. Our simulation results show that the scheme runs faster and has higher signature and verification efficiency than that of the original scheme without compromising security. What's more, we also explore its application in bitcoin and Internet of Things (IoT).

Keywords: ECC; elliptic curve cryptography; bitcoin; IoT; Internet of Things; ECDSA; elliptic curve digital signature; modular inverse operation; signature; verification; efficiency.

Reference to this paper should be made as follows: Xiao, S., Wang, H. and Zhang, J. (2021) 'New digital signature algorithm based on ECC and its application in bitcoin and IoT', *Int. J. High Performance Systems Architecture*, Vol. 10, No. 1, pp.20–31.

Biographical notes: Shuai Xiao received his BS in Information Research and Security from the Engineering University of Chinese People's Armed Police Force, Xi'an, China, in 2018, where he is currently pursuing the MS in Cryptography. His main research interests include information security and cloud storage.

Han Wang received his BS in Computer Science and Technology from Northeastern University, Shenyang, China, in 2018, where he is currently pursuing the MS in Cryptography from the Engineering University of Chinese People's Armed Police Force, Xi'an, China. His main research interests include information security and cloud storage.

Jindan Zhang received her MS and PhD in cryptography in Xidian University, Xi'an, China, in 2008 and 2020. Her research interests include information security and cryptography. She is now a Lecturer in Xianyang Vocational Technical College.

1 Introduction

Cryptography aims to ensure the confidentiality and authenticity of information. Digital signature algorithm (DSA) is a standard and technology proposed by the National Institute of Research (NIST) in August 1991 (Hankerson et al., 2005; Johnson et al., 2001; Menezes et al., 1996). The Federal Information Processing Standard (FIPS 186) of the

US Government is called digital signature standard (DSS). Its security is based on the problem of computational traceability (DLP) of discrete logarithms in prime subgroups of Z_p^*. The purpose of digital signature design is to provide handwritten signature. Ideally, the digital signature scheme should be unforgeable under the attack of chosen plaintext. Elliptic curve digital signature algorithm (ECDSA) is the simulation of DSA using elliptic curve cryptography (ECC). ECDSA became ANSI standard in 1999 and IEEE and NIST standard in 2000. It was accepted by ISO in 1998, and some other standards including it are also considered by ISO. Unlike ordinary discrete logarithm problem (DLP) and integer factorisation problem (IFP), elliptic curve discrete logarithm problem (ECDLP) has no sub-exponential time solution. Therefore, the unit bit strength of elliptic curve cryptography is higher than that of other public key cryptosystems. Digital signature algorithm (DSA) is discussed in detail in the Federal Information Processing Standard (FIPS), which is called digital signature standard. Its security is based on the discrete logarithm problem in prime field. Elliptic curve cryptography (ECC) was invented by Neal Koblitz and Victor Miller in 1985. It can be seen as an elliptic curve simulation of the previous cryptosystem based on the discrete logarithm problem (DLP), except that the group elements are changed from the number of elements in the prime field to the points on the elliptic curve in the finite field. The security of elliptic curve cryptosystem is based on the difficulty of solving elliptic curve discrete logarithm problem (ECDLP). Elliptic curve discrete logarithm problem is far more difficult than discrete logarithm problem. The unit bit strength of elliptic curve cryptosystem is much higher than that of traditional discrete logarithm system. Therefore, ECC can achieve the same level of security as DL system with a shorter key. The advantage of this is that the calculation parameters are smaller, the key is shorter, the operation speed is faster, and the size of the signature is shorter. Therefore, elliptic curve cryptography is especially suitable for situations with limited processing power, storage space, bandwidth and power consumption.

In order to meet the high security and real-time requirements of mobile e-commerce, ECDSA is generally used to realise digital signature technology in mobile e-commerce. Elliptic curve cryptosystem has the advantages of high security intensity, short key length and low bandwidth requirement, so it can better solve the security problems in mobile e-commerce.

At present, ECDSA signature verification algorithm is widely used in bitcoin and other blockchains. This is clearly the technical decision made by Nakamoto (2008) in 2008 based on the then widely used and unauthorised digital signature system. However, there are some serious technical limitations in ECDSA. In particular, multiple signatures and threshold signatures (signed by a quorum of independent parties rather than one person) are difficult to generate together with ECDSA. ECDSA has complex algebraic structure, which makes it inflexible and difficult to use, forcing developers to use bitcoin scripts in applications such as cross-chain atomic exchange or lightning network, which can be implemented more compactly and privately through more flexible signature schemes.

In practical application, digital signature is the core technology of electronic commerce and network security authentication. There are three types of digital signature schemes based on public key cryptography: large integer factorisation, such as RSA; discrete logarithm problem (DLP), such as famous EIGamal, DSA; elliptic curve digital signature based on ECDLP, such as ECDSA in IEEEP 1363 standard. Among the three digital signature schemes, ECDLP is the most difficult to solve. There is still no effective algorithm for solving ECDLP except for some special elliptic curves.

The remainder of this paper is structured as follows. We first review the related work in Section 1.1. Next, Section 2 lists the basic concepts used in the proposed scheme. Section 3 analyses original ECDSA scheme. In Section 4, we analyse our scheme and give experimental results in Section 5. In Section 6 we explore the application of the proposed scheme in bitcoin and Internet of Things (IoT). Finally, Section 7 presents the conclusions and future work of the study.

1.1 Related work

With the in-depth study of ECDSA, the industry has proposed two main calculation factors that affect the signature time-consuming in ECDSA: one is the inversion operation, which is 10 times of the multiplication time consumption in literature (Johnson et al., 2001), and the other is the scalar multiplication operation (Guajardji et al., 1997)). The scalar multiplication operation is the operation process of finding kG with one point G and one random number k of elliptic curve.

In order to solve the time-consuming problem of ECDSA calculation, the industry has proposed various improved schemes for inverse operation and scalar multiplication operation (Hou et al., 2009; Zhao et al., 2004). Chen et al. (2011)) proposed a new digital signature algorithm without inversion. In order to improve the efficiency of signature (Song et al., 2012) proposed that literature (Chen et al., 2011) could improve the efficiency of signature by reducing the inverse operation, but could cause the security problem of forged signature at the same time. Cao et al. (2013) proposed an improved ECDSA algorithm, which effectively avoided the inversion operation to improve the efficiency. Bai et al. (2003) put forward a method to calculate $KP + IQ$ at one time, which reduces the calculation amount of KP and IQ to 25%, and is suitable for wireless networks. Zhang et al. (2008)) proposed an improved ECDSA scheme, which can improve the speed of signature by two times of modular multiplication and one time of modular inversion. Wu (2010) analysed the limitations of Zhang et al. (2008)), and proposed that although the calculation efficiency of Zhang et al. (2008) was improved, it was vulnerable to forgery attack. In Sun (2005), the relation between Hamming distance and hash value is proposed. In this paper, an improved scheme is proposed. The efficiency of signature and verification is significantly improved by eliminating the modular inverse operation.

2 Preliminary

Since our new scheme is based on elliptic curve cryptosystem, it is necessary to review the preparatory knowledge of elliptic curve cryptosystem and elliptic curve discrete logarithm problem before describing the proposed scheme. Table 1 shows the symbols and definitions used in the paper.

Table 1 Symbolic description of schemes

Symbols and abbreviation	*Meaning*
U	Users
M	Message plaintext
H()	Hash function
G	A point on G elliptic curve
mod	Modular operation
d_u	User's private key
e	Hash value
s	Signature
k	Random integer
k^{-1}	Inverse operations for k
R	A quantity related to k
n	Modulus
V	Verifier
T	Elliptic curve domain parameters

2.1 Elliptic curve cryptography

Elliptic curve cryptosystem (ECC) was invented by Koblitz (1987) and Miller (1985). They can be regarded as elliptic curve analogues of the old discrete logarithm (DL cryptosystem), where the subgroups of Z_P^* are replaced by the point groups on elliptic curves over finite fields. The mathematical basis of the security of ECC is the computational difficulty of ECDLP. ECC is a relativity theory of discrete logarithmic cryptography.

The following is the general structure of ECC:

Let $P \in E(Fp)$, point Q be a multiple of P, that is, there exists a positive integer x so that $Q = xP$, then ECDLP is defined by given P and Q. From the current research results, it seems that the discrete logarithm problem on elliptic curve is more difficult to deal with than the discrete logarithm problem on finite field, which provides a new way to construct public key cryptosystem. Based on the elliptic curve discrete logarithm problem, elliptic curve cryptosystem was constructed.

Definition 2.1: Let E be an elliptic curve and P be a point on E. If there exists a positive integer n so that $nP = 0$, then n is the order of point P, where O is an infinite point.

Construction of Elliptic Curve Public Cryptography (ECC) system:

Select the base field F, elliptic curve E, select the point $P(x_p, y_p,)$ whose order is prime n on E. Open information: field F_p, curve equation E, point P and its order n.

User Alice randomly selects integer $d, 1 < d \leq n-1$, calculates $Q = dP$, takes point Q as public key and integer d as secret key.

To send secret information m to Alice, the following steps are required:

1 express plaintext m as an element m in field F_p

2 random selection of integer $k \in [1, n-1]$

3 computation point $(x_1, y_1) = kP$

4 computation point $(x_2, y_2) = kQ$, if $x_2 = 0$, then re-select k

5 calculate $c = mx_2$

6 send (x_1, y_1, c) to Alice.

When Alice receives the ciphertext, she calculates it using the secret key d.

$$D(x_1, y_1) = dkP = k(dP) = kQ = (x_2, y_2)$$

Then calculates $cx_2^{-1} = m$ to get plaintext m.

Here $Q = dP$ is public. If the decoder can solve the discrete logarithm problem on the elliptic curve, he can restore d from dP, and decryption is finished (Mihir, 1998).

2.2 Discrete logarithm problem of elliptic curves (ECDLP)

According to the definition of quantity multiplication of elliptic curves, we can find a point $P \in E$, which satisfies $kP = P + \cdots + P$. Obviously, given k and P, it is easy to calculate Q, but it is difficult to derive k from Q and P. This is the mathematical principle of elliptic curve. If there is another point p on an elliptic curve, the equation $kP = Q$ is satisfied. The elliptic curve discrete logarithm problem is the process of finding k when P and Q are known. That is $k = log_P Q$. In the F_p of the implemented cryptosystem, modular P can reach hundreds of bits. It is very difficult to calculate k.

2.3 Blockchain

The concept of blockchain is essentially the underlying technology in the Internet era. In a narrow sense, it is a kind of chain data structure, which is composed of blocks of data and connected by time sequence, and guarantees non-tampering and non-forgery distributed accounts by cryptography. Broadly speaking, it is a new distributed infrastructure and computing paradigm that uses blockchain data structure to verify and store data, distributed node consensus algorithm to generate and update data, cryptography to ensure data transmission and access security, and smart contract composed of automated script code to program and operate data.

Blockchain technology simply includes the following three concepts: transaction, block and chain.

1 *Transaction*: In bitcoin system, it refers to the exchange of the value of bitcoin, which is extended to the exchange of data based on some key data in the system.

2 *Block*: Records all transactions and states of the system over a period of time.

3 *Chain*: Represents the entire account book, which is connected from blocks generated in chronological order.

Blockchains are stored as data ledgers in point-to-point networks (Yuan and Wang, 2016). Transactions are generated over a period of time in each block storage network. Transactions are broadcast to the whole network by network nodes through broadcasting mechanism. Accounting nodes record transactions in blocks according to consensus mechanism and become new blocks. As shown in Figure 1, the latter block records the hash value of the former block and connects each block into a blockchain (Cai et al., 2017).

3 ECDSA principle and scheme analysis

3.1 Original ECDSA scheme

ECDSA is the combination of ECC and DSA. The whole signature process is similar to DSA. The difference is that the algorithm adopted in the signature is ECC, and the final value of the signature is divided into r, s.

1. Scheme establishment

U is the signer. V is the verifier.

1 U establishes elliptic curve City parameter $T(p, a, b, G, n, h)$ and chooses appropriate safety intensity;

2 U establishes its own key pair (d_u, Q_u), $Q = d_u G$;

3 U chooses a Hash number;

4 U transfers the selected Hash function and the parameter T of elliptic curve domain in a reliable way and passes it to V.

2. Signature algorithm

1 Select the temporary key pair (k, R), where $R = kG = (X_R, Y_R)$ is related to the domain parameter T.

2 Let $r = X_R(mod\ n)$, if $r = 0$, return 1.

3 Calculate the hash value $H = Hash(M)$ of the message to be signed, and convert H into integer E.

4 Calculate $s = k^{-1}(e + rdu)(mod\ n)$, return 1 if $s = 0$.

5 Output $S = (r, s)$ for digital signature.

3. Verification algorithm

Verifier V verifies whether the digital signature sent from signer U is correct, so as to judge whether the received message is true or whether the other party is a real entity.

1 If $r, s \notin [1, n-1]$, the verification fails.

2 Calculate the hash value $H = Hash(M)$ of the message to be signed, and convert H into integer e.

3 Calculate $u_1 = es^{-1}(mod\ n)$, $u_2 = rs^{-1}(mod\ n)$

4 Calculate $R = (x_R, y_R) = u_1 G + u_2 Q_u$, if $R = O$, the validation fails.

5 Let $v = x_R(mod\ n)$, if $v = r$, the validation succeeds, otherwise the validation fails.

3.2 Analysis of ECDSA Scheme

Existing elliptic curve digital signature schemes, such as ECDSA, the ANSIX 9.62 standard issued by NIST, can be described in detail in reference (Hankerson et al., 2005; Johnson et al., 2001; Menezes et al., 1996); EC-KCDSA is an elliptic curve version based on certificate digital signature algorithm (KCDSA) in Korea, and the specific scheme can be described in reference (Hankerson et al., 2005).

According to Trappe and Washington (2004), the following transformations are needed to transform a traditional discrete logarithmic system into an elliptic curve system:

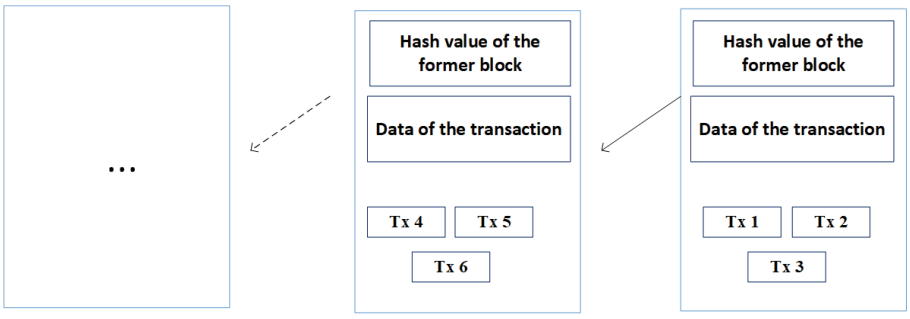
1 the modular multiplication operation is transformed into point addition operation on an elliptic curve

2 the modular exponentiation operation is transformed into the point multiplication operation on the elliptic curve.

The discrete logarithmic signature scheme over finite field Fp is generally based on the signature equation.

Details can be found in (Yang et al., 2002). It consists of five elements (d, k, u, v, w). d is the key of the signer, k is the random integer generated at each signature (i.e. the private key of message). (u, v, w) can be replaced by (e, r, s).$e = h(m)$ is the Hash value of the signed message, r is a quantity related to k, s is the signature.

Figure 1 Blockchain structure (see online version for colours)



Different combinations of (e, r, s) can be used instead of (u, v, w), and (e, r, s) can be replaced by their additive or multiplicative inverse elements respectively, thus different signature equations can be derived, and different digital signature schemes can be obtained. But not all mathematical combinations can produce secure digital signature schemes. Ham and Xu give the design rules of secure discrete logarithmic signature schemes, and list all digital signature schemes that meet the design rules (Washington, 2003).

ElGamal scheme and ECDSA scheme based on ECC are the fourth one. Take $u = e, v = r, w = s$ as variants of the general equation $u = vd + kw(mod\ p) :\ e = dr + ks$ is equivalent to $s = k^{-1}(e + dr)$, which is the signature equation of ECDSA; the corresponding verification equation $s^{-1}(eG + rQ) = kG$. The public key generation algorithm in ECDSA scheme is $Q = dG$.

When generating and verifying signatures, $k^{-1}modn$ and $s^{-1}mod\ n$ are computed respectively, requiring modular inversion. The most important feature of EC-KCDSA scheme is that $Q = d^{-1}p$ adopts inverse pre-operation, which makes modular inversion operation unnecessary in the process of signature generation and verification. In the existing elliptic curve encryption or signature process, the inversion is the most time-consuming operation. The time of one inversion is about 80 times that of point multiplication, so the inversion operation is the main computational burden.

In ECDSA scheme, the public key generation algorithm is $Q_u = d_uG$. In the process of signature generation and verification, $k^{-1}mod\ n$ and $s^{-1}mod\ n$ need to be calculated respectively, that is, modular inversion operation is needed.

If the data scale of modular multiplication is n, the complexity of a modular multiplication operation is $O(n^2 log\ n)$. In the signature algorithm, a point product, a modular inversion, two modular multiplications and one modular inversion are performed, which is equivalent to about nine modular multiplications, and the total amount of computation is $(log\ n + 11)n^2$. In the validation algorithm, two point products, one modular inversion and two modular multiplication are performed, and the total amount of computation is $(2log\, n + 11)n^2$. As can be seen, the operation of signature and verification algorithms is very complex. In the existing elliptic curve encryption or signature process, the main computational burden comes from the inversion operation, which is the most complex and time-consuming operation. One inversion takes about 80 times of point multiplication.

4 Proposal of new signature scheme

4.1 Demonstration of the scheme

Through the analysis of the original ECDSA signature scheme, the complex and time-consuming inversion operation restricts the improvement of the efficiency of digital signature. If the inversion operation can be completely avoided in the whole signature and verification process (including the parameter generation process), the operation efficiency can be improved.

According to the general signature equation $u = dv + kw(modp)$, when $u = e, v = r, w = s$, the equation is $e = dr + ks(mod\, p)$. The inversion operation can be avoided if the signature equation is modified appropriately (as long as s and k are two separate terms, d, e, r are three elements to form one). Because s is a single term, it can ensure that the calculation of signature s does not need to be inverted, k is a single term to ensure that the calculation of signature kG does not need to be inverted, d, e, r combination can be multiplied or added. Therefore, the signature equation can be designed as $s = k + erd$, which fully conforms to the secure digital signature rule (Cai et al., 2017) given by Harn and Xu, and the corresponding verification equation is $sG - erQ = kG$. Obviously, there is no modular inverse operation in both equations.

Based on the above demonstration, the improvement scheme is put forward.

4.2 Description of the new scheme

(1) Parameter selection

(a) Take MD5 as hash function, input messages of any length, and output compressed value as 128 bit.

(b) Select a secure elliptic curve $E(F_p) : y^2 = x^3 + ax + b\ (\mathrm{mod}\ p)$ over a large prime field F_p. It is described by the six tuple parameter T $=$ $(P,\ a,\ b,\ G,\ n,\ h)$ specified by SEC 1, where p is a prime number greater than 3, which determines the finite field F_p; the element $a, b \in F_p$ determines the elliptic curve; G is a base point on the elliptic curve, and the order of the base point is prime number n; if the order of elliptic curve group $E(F_p)$ is n, then $h = N/n$ is called cofactor. Among them, $E(F_p), F_p, G, n$ are open.

(c) The private key of user is d_U, $d_U \in [1, n-1]$, and the public key of user is $Q_u = d_UG$. d_U is confidential and Q_u is open.

(2) Signature generation

(a) U randomly generates an integer k as the secret key of the message, $k \in [1, n-1]$, and calculates $R = (x_1, y_1) = kG$, let $r = x_1(\mathrm{mod}\ n)$, if $r = 0$, select k again;

(b) Calculate the hash value of message m;

(c) Calculate signature $s = k + erd_U$;

(d) U sends message m and its signature (s, r) to V.

(3) Signature Verification

(a) For signature (s, r), if r and s are not integers in $[1, n-1]$ interval, then reject the signature, otherwise, the verification continues;

(b) For message m, calculate $e = H(m)$;

(c) Calculate $u = er(\mathrm{mod}\ n)$, and $(x_R, y_R) = sG - uQ_U$;

(d) let $r' = x_R(\mathrm{mod}\ n)$. If $r' = r$, then (s, r) is the correct signature of U.

Figure 2 shows the algorithm flowchart of the proposed scheme.The proposed scheme is improved on the basis of the original ECDSA scheme. The following is illustrated by comparison with the original scheme. The other steps remain unchanged. Therefore, the description is omitted here. As can be seen from Table 2, the main improvement in the proposed scheme is to avoid inversion operation when calculating signature, which reduces the amount of computation and improves the efficiency of signature and verification.

4.3 Algorithmic analysis of the scheme

The security of the improved scheme proposed in this paper is based on the difficulty of solving elliptic curve discrete logarithm problem. The signature equation is a variant of the general signature equation based on elliptic curve. The improvement of the algorithm aims at improving the operation efficiency. By choosing the signature equation, the whole calculation process from key generation, signature process to verification process does not need inverse operation, which is more efficient than the ECDSA scheme. Except for the multiplication and modular operations on elliptic curves, the others are algebraic operations in finite fields, which reduces the computational complexity, greatly improves the computational speed, reduces the computational burden and improves the efficiency.

Of course, the actual operation speed depends not only on the algorithm design of the scheme, but also on the selection of elliptic curve parameters.

4.4 Analysis of the correctness of the scheme

The correctness of the improved algorithm is proved as follows: if (r, s) is the signature information of m, then

$$\begin{aligned} R &= (x_R, y_R) \\ &= sG - uQ_u \\ &= [k + (d_u er)]\, G - (er)Qu \\ &= kG + erd_u G - erQ_u \\ &= kG + erQ - erQ \\ &= kG \\ &= (x', y') \end{aligned}$$

So there is $\nu = x' = x = r \bmod n$. That is to say, the improved algorithm is correct.

4.5 Security analysis of the scheme

(1) *Unforgeability*

The most important security of ECDSA is unforgeability. The forger can be anyone except the signer himself.

To forge signatures, an attacker must determine a pair of (r, s) so that the verification equation $R = (x_R, y_R) =$

Figure 2 Algorithm flowchart of the proposed scheme

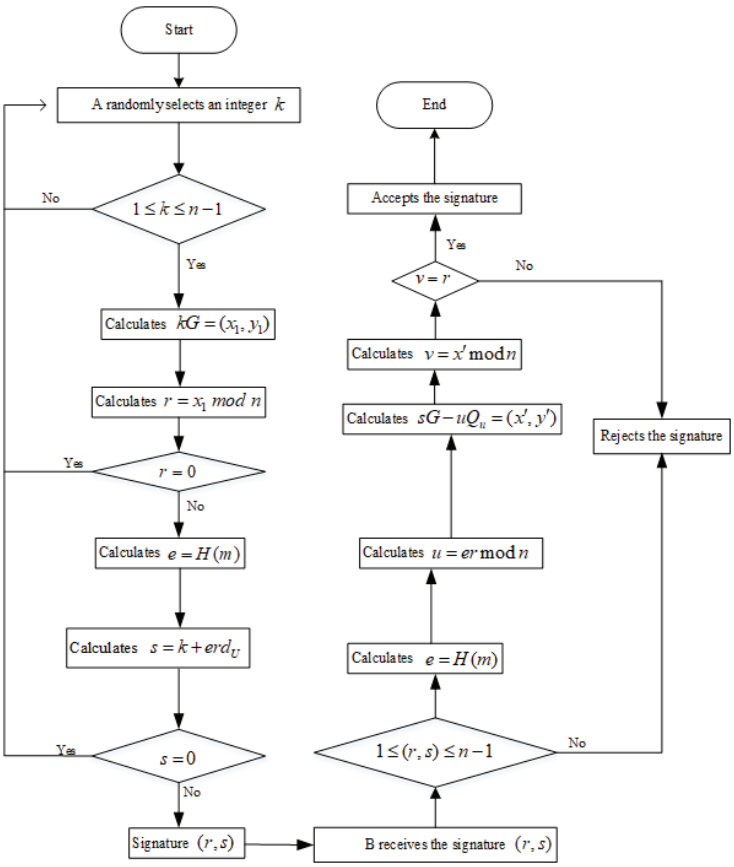


Table 2 Comparisons of ECDSA and the proposed scheme

	The original ECDSA scheme	*The proposed scheme*
Signature algorithm	The hash value $e = hash(M)$ of the message to be signed is calculated.	The hash value $e = hash(M)$ of the message to be signed is calculated.
Verification algorithm	Calculate signature $s = k^{-1}(e + rd_u)$ mod n; Calculate $u_1 = es^{-1} mod n$, $u_2 = rs^{-1} mod n$; Calculate $R = (x_R, y_R) = u_1G + u_2Qu$	Compute the signature $s = (k + erd_u)(mod\, n)$; Calculate $u = er(mod\, n)$, and $(x_R, y_R) = sG - uQ_u$

$sG - uQ = [k + (d_u er)]G - (er)Qu$ holds. If an attacker first determines an r, then R is determined accordingly. The verification equation requires that the solutions belong to the typical ECDLP.

(2) *Non-Repudiation*

Non-repudiation means that a signer cannot deny his signature.The object of non-repudiation is the signer himself.

Generally speaking, unforgeability implies non-repudiation signature. If it is possible for others to forge or extend a signature. It can be denied completely because the recipient does not have enough evidence to prove that the signature is from the signer. If the signature is not forgeable, when the receiver presents a legitimate signature, the signature may only be issued by the signer himself. So the signer can not deny it. In 2002, J. Stern, D. Pointcheval, J. Malone-Lee and others found that ECDSA signatures can produce duplicate signatures. Therefore, it is not sufficient to rely solely on non-forgery to illustrate non-repudiation.

Because the X-coordinates of two symmetric points of elliptic curve are the same: $R = (x_R, y_R), -R = (x_R, -y_R)$. Therefore, mapping in ECDSA $f : R \to r$ is not a one-to-one mapping. For triples (m_1, R, s) and $(m_2, -R, s)$, the same signature text (r, s) can be obtained. Since both (m_1, r, s) and (m_2, r, s) can be verified, signers can deny (m_1, r, s) with (m_2, r, s).

Let m_1 and m_2 be two different messages, $e_1 = hash(m_1)$, $e_2 = hash(m_2)$.

(r_1, s_1) is a signature for m_1, with $s_1 = (k + d_u e_1 r_1)$ $mod\ n$.

(r_2, s_2) is a signature for m_2, with $s_2 = (k + d_u e_2 r_2)$ $mod\ n$.

Let $r_1 = r_2 = r, e_2 = -e_1 - 2k/rd$, there is $s_2 = s_1$. The signature generated for message m_2 is called copy signature. $e_2 = hash(m_2)$ is a one-way function, and the signer cannot calculate the appropriate m_2 for message m_1. But if the signer can control the key generation process, he can generate a copy signature by generating a key: when the signer signs a given message m_1, he can randomly select a copy message m_2 and calculate the private key $d_u = -2k(e_2 + e_1)r\ mod\ n$ for the signature, and the corresponding public key is $Q_u = d_u G$.

4.6 Risk analysis

In ECDSA, random number k has the same security requirements as private key d. Because if the random number k of user A is known, the attacker can calculate the private key of user A. Therefore, it is necessary to ensure that random number k can be generated, stored and destroyed safely. What's more, the random number used to sign multiple messages must be independent of each other. Every time k is signed, it must be different, otherwise d can be recovered. In practice, the pseudo-random number generator is generally used to generate different k. The security of the improved scheme is guaranteed by the security of the traditional elliptic curve digital signature scheme, while the security of the traditional elliptic curve digital signature scheme is based on the difficulty of solving the elliptic curve discrete logarithm problem. In other words, the proposed scheme has the same security as the original one.

If the attacker forges a signature, there is $s' = k' + e'r'd'$. According to the verifying equation

$$\begin{aligned} R &= (x_R, y_R) \\ &= s'G - u'Qu \\ &= (k' + e'r'd')G - e'r'd'G \\ &= k'G \\ &\neq kG \end{aligned}$$

so the verification fails, and the signature is invalid.

4.7 Efficiency analysis

In the scheme of this paper, there is no modulo inverse operation in the process of signature and verification. We analyse the efficiency change of the improved scheme through specific numerical value. In the process of digital signature, the time-consuming is mainly concentrated in multiplication, inverse operation and scalar multiplication operation. We can simplify them respectively as follows: $[l]$, $[i]$, $[h]$. Considering that addition and other operations have little influence on the time-consuming, they can be ignored. One inversion calculation is equivalent to about 10 multiplication operations, that is $[i] = 10\,[l]$. According to (Bai and Huang, 2003), scalar multiplication meets the requirement that $[h] = 75\,[i] + 173\,[l] = 750\,[l] + 173\,[l] = 923\,[l]$ under 163b, suppose the data scale of modular multiplication operation is m, Table 3 (Sig is short for signing and Ver is short for verifying. The unit of numerical value in the table is the number of operations) shows the time-consuming comparison between the improved new scheme and the classical ECDSA scheme.

It can be seen that in the ECDSA scheme, the signature process uses two times of multiplication, one time of scalar multiplication and one time of modular inversion. The total amount of calculation is $s_1 = 2\,[l] + [h] + [i] = 2\,[l] + 923\,[l] + 10\,[l] = 935\,[l]$. The verification process uses two times of multiplication, two times of scalar multiplication

Table 3 Time consumption comparison between ECDSA and the proposed scheme

	Multiplication		*Scalar multiplication*		*Inverse operation*		*Total*	
Schemes	*Sig*	*Ver*	*Sig*	*Ver*	*Sig*	*Ver*	*Sig*	*Ver*
ECDSA	2	2	1	2	1	1	$935[l]$	$1858[l]$
The proposed scheme	3	2	1	1	0	0	$926[l]$	$925[l]$

and one time of modular inversion. The total amount of calculation is $s_2 = 2[l] + 2[h] + [i] = 2[l] + 2 \times 923[l] + 10[l] = 2[l] + 1846[l] + 10[l] = 1858[l]$. In the proposed scheme, three times of multiplication and one time of scalar multiplication are used in the signature process. The total amount of computation is $s_1{}' = 3[l] + [h] = 3[l] + 923[l] = 926[l]$. The verification process uses two multiplication operations, one scalar multiplication operation, and no modular inversion operation. The total amount of calculation is $s_2{}' = 2[l] + [h] = 2[l] + 923[l] = 925[l]$. Comparing the two schemes, the signature efficiency is improved by about 0.96% ($\eta_1 = \frac{935-926}{935} \times 100\% \approx 0.96\%$) and the verification efficiency is improved by about 50.22% ($\eta_2 = \frac{1858-925}{1858} \times 100\% \approx 50.22\%$).

5 Experiment on ECC-based signature scheme

5.1 Experimental method

Next, we use Java programming to simulate the signature scheme, and further verify the efficiency of the proposed scheme.

In the following experiments, we take elliptic curves over a smaller finite field, and take the same group of elliptic curve parameters. The security of the improved scheme is guaranteed by the security of the traditional elliptic curve digital signature scheme, while the security of the traditional elliptic curve digital signature scheme is based on the difficulty of solving the elliptic curve discrete logarithm problem, that is, under the same security conditions, we use bouncycastle cryptography library and select P-384 curve to carry out Java experiments to analyse and compare the execution time between the proposed scheme and ECDSA.

5.2 Experimental environment

Hardware environment: Surface Pro (I5-8250U, 8G memory).

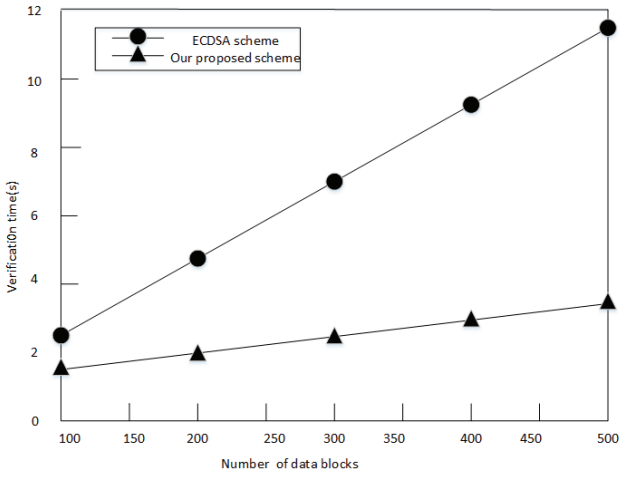
Software environment: Under Java 11, using the same elliptic curve parameter, two schemes are used to verify the same documents.

The experiment results are shown in Figure 3.

5.3 Analysis of experimental data

The file is divided into several data blocks. It can be seen from Table 3 that the proposed scheme has little improvement in signature efficiency, therefore, we only implement the efficiency experiment of signature verification. Experiments are carried out on the efficiency of verification under fixed block sizes (The size of the data block is 1KB). When the user's data block size is fixed (1 KB). As the number of data blocks increases, the verification time required for our proposed scheme takes less than ECDSA scheme, as shown in Figure 3. when there are a lot of signature and verification calculations, the proposed scheme has a greater efficiency advantage than the classical ECDSA scheme. The signature and verification time of the proposed scheme is shorter than that of the ECDSA scheme, which is consistent with the theoretical analysis.

Figure 3 Verification time comparison for the two schemes



Because the proposed scheme avoids the inversion operation, only one finite field multiplication is used in the signature equations, and the rest are all finite field addition and subtraction operations. In the signature equations in ECDSA, there are one finite field inversion, one multiplication and the rest are addition and subtraction operations. Inversion is the most time-consuming operation, so the proposed scheme is more time-saving than the ECDSA scheme. In summary, the improvement made in the proposed scheme is helpful for the improvement of the efficiency of signature and verification.

6 Application of the proposed scheme

6.1 Application in bitcoin

6.1.1 Bitcoin

The concept of bitcoin was first proposed by Nakamoto on 1 November, 2008, and was formally born on 3 January, 2009. According to the idea of Nakamoto, open source software is designed and released, and the P2P network on it is constructed. Bitcoin is a virtual encrypted digital currency in the form of P2P. Point-to-point transmission means a decentralised payment system.

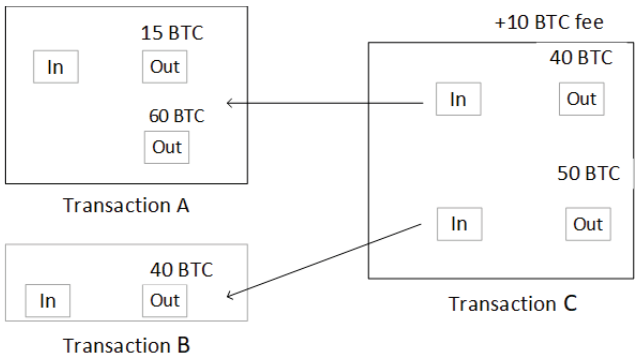
Unlike all currencies, in bitcoin economy, a distributed database is used , which is composed of many nodes in the entire P2P network to confirm and record all transactions, and cryptographic design is used to ensure the security of all aspects of currency circulation. The de-centralisation of P2P and the algorithm itself can ensure that the value of the currency cannot be manipulated artificially by making a large number of bitcoins. Cryptographic design allows bitcoins to be transferred or paid only by real owners. This also ensures the anonymity of currency ownership and circulation transactions.

6.1.2 *The process of bitcoin transaction*

Bitcoin system has no concept of balance. It uses UTXO (Unspent Transaction Outputs) model. The wallet balance in the transaction process is actually a set of UTXO wallet addresses. Therefore, in the bitcoin network, the balance of the bitcoin is stored in the output of the transaction, in other words, the output of the unused transaction, and the input of each transaction actually refers to the output of the previous transaction.

Figure 4 shows an example of a bitcoin transaction. Transaction C costs 100 bitcoins from transaction A and transaction B. In this transaction, 60BTC in transaction A and 40 BTC in transaction B are spent, and then 40 BTC , 50 BTC and 10 BTC (The number of creation is specified by the user and can be any number. As long as the total amount does not exceed 100 BTC, it is a legal transaction) in transaction C are created. The difference of 10 BTC will be paid to the miner as a fee.

Figure 4 An example of a bitcoin transaction



6.1.3 *Application model of the proposed scheme in bitcoin transaction*

In bitcoin transaction, ECDSA is used to unlock the unspent balance in UTXO. The unlock script can verify whether UTXO belongs to a user. The unlock script includes the user's digital signature and public key. The specific work process of bitcoin and the application model of the proposed scheme in bitcoin transaction are shown in Figure 5. The working mechanism of transactions can be divided into four steps.

Step 1: Bob and Alice create the transaction. First, a transaction is created between the two parties. Anyone can create a transaction using three required components (input, amount, and output). For example, when Bob sends bitcoin to Alice, Alice needs to send her bitcoin address (public), Bob creates the transaction and signs it with the private key.

Step 2: Live bitcoin Transaction. Then bitcoin transaction is broadcast. Once a transaction is created, it is sent to the nearest node on the bitcoin network. It can be sent for a long time after creation (just make sure there is enough bitcoin in the wallet when deciding to send).

Step 3: Propagation and Validation. Next is the propagation and validation phase. Once the transaction reaches the nearest node, it is propagated to the network and verified. After it passes the validation successfully, it will enter the Memory Pool and wait patiently for the miners to extract it and include it in the next block. Finally, the block is verified.

Step 4: Block validation. Once the transaction is on Memory Pool, the mining staff will extract the transaction (first of all, those who pay more transaction fees) and group it. By using proof of work consensus algorithm, the network will reach an agreement on the effective block and transaction every 10 minutes on average.

With the rapid development of internet, e-commerce has become a new model of business activities. As a new type of e-commerce, mobile e-commerce takes advantage of many advantages of mobile wireless network. Compared with traditional wired e-commerce, it has obvious advantages and is a useful supplement to traditional e-commerce. With the continuous improvement of the processing speed of mobile communication terminals, mobile e-commerce will also develop rapidly. Although the development momentum of mobile e-commerce is amazing, it only accounts for a very small part of the global trade volume. Security is still the biggest obstacle to the development of mobile e-commerce. Only by further improving various security technologies of mobile e-commerce, including digital signatures, can the healthy and smooth development of mobile e-commerce be ensured. One of the main obstacles is how to ensure the security of data transmission and identity confirmation of both parties. In this sense, the proposed scheme will also make bitcoin transaction better and more efficient.

6.2 *Application in IoT*

The IoT is a worldwide interconnection of devices (Yaqoob et al., 2019). In dynamic networking devices and systems, devices communicate, perceive and interact with their environment. The generated data are collected and transmitted through sensors to achieve real-time transmission of information (Zhao et al., 2017; Sarkar et al., 2018). This feature makes the related applications of IoT grow rapidly, replacing the traditional industries (such as manufacturing). Division of work (Zhao et al., 2018). IoT has been greatly developed in recent years. Its application fields include smart city (Shafiq et al., 2020), logistics management, medicine, industrial management (Sadoughi et al., 2020), the convergence of cloud computing (Banijamali et al., 2020), etc.

The traditional IoT can no longer satisfy users' privacy and information security, and access control and data are

Figure 5 The basic process of bitcoin transactions (see online version for colours)

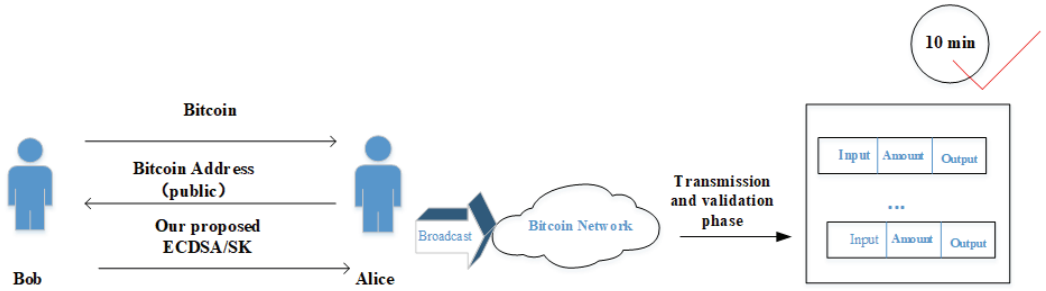
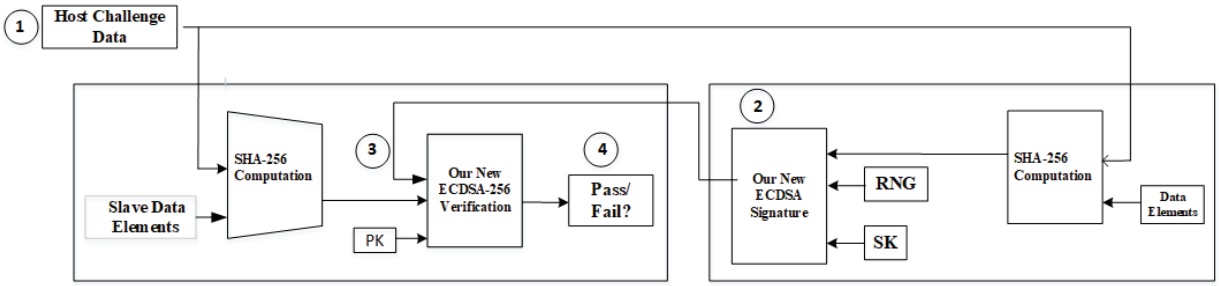


Figure 6 Secure authentication based on new ECDSA algorithm



also threatened by the outside world. In order to protect the privacy of IoT, we need to introduce a mechanism to ensure the privacy of data in IoT. Blockchain technology with the characteristics of de-centralisation, de-trust of third parties and data encryption provides a consensus mechanism for the IoT, which can solve the security and privacy problems of the traditional IoT (Zhang et al., 2017). Blockchain technology is also known as distributed bookkeeping technology. It records growing transaction lists, distributes bookkeeping and cryptography through consensus mechanism, forms shared, non-tampering information and transaction blocks, and then links them in sequence (Hoy, 2017).

Digital security, such as IoT devices and intelligent electronic devices, is now one of the hottest topics in the field of intelligent life. However, when it comes to security, people often think of encryption, and only a few people think of security authentication. In fact, real device and identity security is not only simple encryption, security authentication and protection play a very important role in device security, IoT identity authentication cannot be separated from security authentication. Security authentication is the basic function for electronic devices to participate in safe interaction and use. Especially in the field of IoT, security authentication is very important: untrustworthy terminals may put the entire infrastructure at risk.

According to a simple example, we are now using more and more electronic devices with Internet access functions, such as smart cameras, smart rice cookers, smart speakers and other smart home devices, as well as sharing bicycles and other devices that need authentication. Without security authentication function, the invasion of any node will cause huge losses to users or manufacturers. Another point is that for shared devices or devices with unique property rights, security chips must be used to ensure that devices are not used plagiarised. A better security chip generally has two functions:

(a) Protect electronic devices from plagiarism or intrusion, and ensure that the device itself is not plagiarised.

(b) Access to the IoT can implement the core functions of identity authentication to ensure that business profit models are not embezzled.

The simplest way to secure authentication is to use a password. But because attackers can easily monitor communications, record passwords, and then use them to authenticate non-real devices. Therefore, password-based security authentication methods are relatively weak. A better way to perform security authentication in the electronic field is the handshake response method. There are two ways to shake hands: One is based on symmetric encryption, the other is based on asymmetric encryption. Security authentication based on asymmetric encryption depends on two keys: private key and public key. Only the authenticated device knows the private key, and the public key can be disclosed to any party wishing to authenticate the device securely. It is important that functions used to calculate digital signatures have specific mathematical properties. The most commonly used functions in asymmetric methods are RSA and ECDSA. Similarly, without revealing the key, the device submits a proof that it knows the key, that is, the private key. Safety certification is a key and important issue that needs to be considered in the design of a preliminary product plan. In security authentication chips, ECDSA or RSA-based security chips use private/public key pairs.

6.2.1 Application model of the proposed scheme in IoT chip security authentication

Figure 6 shows the logic interface of the proposed scheme in the application of IoT chip security authentication. Circle 1 shows that the host sends a challenge to the device. Circle 2 shows that the device calculates the digital signature according

to the challenge and private key, Circle 3 shows that the device uses the public key pair and digital signature to verify, Circle 4 shows the verification result, and if the signature verification is passed, it indicates that the chip has passed the security authentication. The host challenge data and the slave data elements in the host system are computed under SHA-256, and the computation result is recorded as e. On the other hand, the host challenge data and the data elements in the accessories are calculated under SHA-256, and the calculation result, together with the random number generated (RNG) by the random number generator, and the private key is signed under the proposed scheme, and the signature result is recorded as s. Finally, e, s use the public key to authenticate under the proposed scheme, so as to judge whether the authentication passes or fails.

7 Conclusion

In this paper, an improved digital signature algorithm without modular inverse operation is proposed. Our contributions can be roughly summarised as the following: We analyse the classical ECDSA scheme and propose an improved digital signature algorithm without modular inverse operation. For the proposed improvement scheme, we carry out simulation experiments. Our simulation results show that the scheme runs faster and has higher signature and verification efficiency than that of the original scheme without compromising security. Theoretically, the analysis shows that it not only completely avoids the most time-consuming inversion operation, but also is a safe signature scheme, and in the experiment, the java 11 environment is tested to prove that the signature and verification time of the scheme is shorter than ECDSA. Finally, the significance of improving ECDSA scheme and the application of the proposed scheme in bitcoin transactions and the Internet of Things are described. Of course, there are some limitations in this paper, such as the lack of provable security. In addition, it is our hope that the new scheme can also be used to aggregate signatures, which could be our next research direction.

Acknowledgement

This work is supported by the National Cryptography Development Fund of China Under Grants No. MMJJ20170112, an Open Project from Guizhou Provincial Key Laboratory of Public Big Data under Grant No. 2019BDKFJJ008.

References

Bai, G. and Huang, Z. (2003) 'Fast verification algorithm in elliptic curve digital signature algorithm. *Journal of Tsinghua University: Natural Science Edition*, Vol. 43, No. 4, pp.564–568.

Banijamali, A., Pakanen, O-P., Kuvaja, P. and Oivo, M. (2020) 'Software architectures of the convergence of cloud computing and the internet of Things: a systematic literature review', *Information and Software Technology*, Vol. 122.

Cai, W., Yu, L., Wang, R., Liu, N. and Deng, E. (2017) 'Blockchain application development techniques', *Journal of Software*, Vol. 28, No. 6, pp.1474–1487.

Chen, L. and You, L. (2011) 'Optimization and design of elliptic curve digital signature algorithm', *Electronic devices*, Vol. 34, No. 1, pp.89–93.

Hankerson, D., Menezes, A. and Vanstone, S. (2004) *Guide to Elliptic Curve Cryptography*, Electronic Industry Press, Springer.

Hou, A., Gao, B., Zhang, W. and Qiang, Y. (2009) 'An efficient digital signature based on elliptic curve', *Computer Application and Software*, Vol. 26, No. 2, pp.58–60

Hoy, M. (2017) 'An introduction to the Blockchain and its implications for libraries and medicine', *Medical Reference Services Quarterly*, Vol. 36, No. 3, pp.273–279.

Johnson, D., Menezes, A. and Vanstone, S. (2001) 'The elliptic curve digital signature algorithm (ECDSA)', *International Journal of Information Security*, Vol. 1, pp.36–63.

Koblitz, N. (1987) 'Elliptic curve cryptosystems', *Mathematics of Computation*, Vol. 48, pp.203–209.

Li, X., Niu, Y., Wei, L., Zhang, C. and Yu, N. (2019) 'A survey of bitcoin privacy protection', *Journal of Cryptography*, Vol. 6, No. 2, pp.133–149.

Menezes, A., Orschot, P. and Vanstone, S. (1996) *Handbook of Applied Cryptography*, CRC Press, London, pp.454–459.

Mihir, B., Ran, C. and Hugo, K. (1998) 'A modular approach to the design and analysis of authentication and key exchange protocols', *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing*, TX, USA, pp.419–428.

Miller, V. (1985) 'Use of elliptic curves in cryptography', *CRYPTO*, Santa Barbara, California, USA, 18–22 August; Proceedings. Springer-Verlag, New York, Inc., 1986.

Nakamoto, S. (2008) *Bitcoin: A Peer-To-Peer Electronic Cash System*, https://bitcoin.org/bitcoin.pdf (Accessed 1 January, 2015).

Ray, P. (2018) 'A survey on Internet of things architectures', *Journal of King Saud University-Computer and Information Sciences*, Vol. 30, No. 3, pp.291–319.

Sadoughi, F., Behmanesh, A. and Sayfouri, N. (2020) 'Internet of things in medicine: a systematic mapping study', *Journal of Biomedical Informatics*, Vol. 103.

Sarkar, S., Chatterjee, S. and Misra, S. (2018) 'Assessment of the suitability of fog computing in the context of Internet of things', *IEEE Trans on Cloud Computing*, Vol. 6, No. 1, pp.46–59.

Shafiq, M., Tian, Z., Sun, Y., Du, X. and Guizani, M. (2020) 'Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city', *Future Generation Computer Systems*, Vol. 107, pp.433–442.

Trappe, W. and Washington, L. (2004) *Introduction to Cryptography with Coding Theory*, Zou, H., Xu, P. and Li, Y. Translated, People's Posts and Telecommunications Press, Pearson.

Washington, L. (2003) *Elliptic Curves Number Theory and Cryptography Discrete Mathematics and its Applications*, Chapman&Hall/CRC Press, New York.

Yang, Y, Sun, W. and Niu, X. (2002) *New Theory of Modern Cryptography*, Beijing Science Press.

Yaqoob, I., Hashem, I.A.T., Ahmed, A., Ahsan Kazmi, S.M. and Hong, C.S. (2019) 'Internet of things forensics: recent advances, taxonomy, requirements, and open challenges', *Future Generation Computer Systems*, Vol. 92, pp.265–275.

Yuan, Y. and Wang, F. (2016) 'Blockchain: the state of the art and future trends', *Acta Automatica Sinica*, Vol. 42, No. 4, pp.481–494.

Zhang, Y., Zhou, W. and Peng, A. (2017) 'Survey of internet of things security', *Journal of Computer Research and Development*, Vol. 54, No. 10, pp.2130–2143.

Zhang, Q., Guo, B. and Cheng, D. (2008) 'Fast elliptic curve verification algorithm', *Computer Engineering and Design*, Vol. 17, No. 29, pp.4425–4427.

Zhao, Z., Liu, F. and Xu, H. (2004) 'Construction method of signature equation based on elliptic curve cryptosystem', *Computer Engineering*, Vol. 30, No. 19, pp.96–97.

Zhao, H, Lian, X, Hao, B., Luo, X. and Gong, Y. (2017) 'Remote control system of air conditioning system based on internet of things cloud platform', *Computer Engineering and Design*, Vol. 38, No. 1, pp.265–270.

Zhao, H., Li, H. and Liu, J. (2018) 'Study on RFID complex event pattern clustering algorithm of Internet of things', *Application Research of Computers*, Vol. 35, No. 2, pp.339–341.