An Improvement of ECDSA Weak Randomness in Blockchain

Nisreen T. Hussein¹, Ali H. Kashmar².
¹,² University of Baghdad, College of Science, Computer Science Dept.
nesreen.tareq1201a@sc.uobaghdad.edu.iq, Ali.Kashmar@sc.uobaghdad.edu.iq

Corresponding Author: nesreen.tareq1201a@sc.uobaghdad.edu.iq
Mobile: 07714330694

Abstract
Blockchain technology has already changed industry and commercial enterprises remarkably. It is the underlying mechanism of a very well-known cryptocurrencies such as Bitcoin and Ethereum, and many other business applications. Therefore, its security draws the researchers' attention more and more recently. One of Blockchain vulnerabilities is caused by weak randomness in ECDSA. A random number is not secure, cryptographically, which leads to a leakage in private key and even the user's fund theft. As well the spam transaction attack may exploit the ECDSA weak randomness. This problem in security has been well known in cryptocurrencies community such as Bitcoin and fixed by applying RFC 6979 update in 2013. However, the problem is not entirely solved.

The elliptic curve digital signature algorithm (ECDSA) was the first successful algorithm based on elliptic curve. This algorithm security depends on complexity of elliptic curve discrete logarithm problem (ECDLP). This algorithm applied in blockchain mechanism as a result of its low computational cost and short key.

In this paper, we analyze the ECDSA weakness in blockchain and enhance its scheme by generating the signature with two secret keys. Using two secret keys will reduce the risk probability of revealing the secret key by knowing two messages. Therefore, the improved scheme can improve the security of the ECDSA.

Key words: ECDSA, Blockchain, digital signature, Bitcoin, random number.

1. Introduction
Blockchain presented for the first time by Satoshi Nakamoto in 2008 after his paper publication named Bitcoin: A Peer To Peer Electronic Cash System, as a peer-to-peer payment for digital transactions which let financial actors do payments to each other without the intermediate agent (such as central bank), preventing the double-spending problem (1).

Since the Bitcoin is launched it has gained increasing attention by organizations, business companies, and governments. Recently, Japan and Germany admit Bitcoin as a method for international payment.

The Blockchain underlying technology is relies on many classical cryptography and distributed system technologies (2), such as ECDSA, consensus mechanism such as Proof-of-Work, and Merkle tree. However, the Blockchain security affected if one of its basic components are exposed to security vulnerabilities. Due to the wide use of Blockchain in cryptocurrencies and business applications, there would be great economic losses if there is no enough security protection for users' funds.
Digital Signature refers to sign by using cryptographic mechanisms, a digitized or computerized document, instead of seal or handwritten signature. The signature receiver can verify that the document is not altered after being signed, this called signature verification, thus ensuring the document integrity and information authenticity. Digital signature (3), as a technique, has many properties as follows:

- **Credibility**: the receiver of signature can ensure that the received signature is indeed signed by a legitimate signer.
- **Non-reusable**: the information in the signature file not to be used as other documents signatures.
- **Unforgeability**: the owned signature can be generated only by his signer.
- **Non-repudiation**: the signer unable to deny his signature.

As we know that the digital signature algorithm RSA is the one of the most well-known cryptosystems (4). It is applied to many electronic commerce applications. The RSA key to be broken, you need to factor a large number. Now, the RSA secure key size is 4,096 bits (5). Processing the large key has slower algorithms for signatures generating. This means slower loading time on websites and slower connection. ECDSA signature solves the above problems, and it is the best choice as a result of its short key size. Many cryptosystems apply the ECDSA digital signature instead of RSA, such as Bitcoin (1), Apple ecosystem, and SSL/TLS. Liao and Shen 2006 (6) shows that there is a weakness in ECDSA. When the random number is repeated to the ECDSA, an adversary will be able to derive the private key, which is vital to protecting users’ fund and even the system. To remedy this weakness, Liao and Shen proposed an enhanced ECDSA (6). The computational cost is reduced by the improved schemes, but the security is not improved. However, in this paper we will propose an ECDSA improvement using two secret keys.

This paper is organized as follows: in section two we explain preliminaries of ECDSA and blockchain cryptography briefly. Section three there is our improved ECDSA. In section four we analyze the security of our proposal. The last section contains the conclusion.

2. **Preliminaries**

In this section, we briefly explain elliptic curve (EC) with elliptic curve discrete logarithm problem (ECDLP) (7), ECDSA (8), and an overview of the entire cryptography of blockchain.

**Elliptic curve**

Consider $GF(p)$, be a prime field and $a, b \in GF(p)$ are constants. An elliptic curve $E_p(a, b)$ over $GF(p)$ is defined as the set of points $(x, y) \in GF(p) \times GF(p)$ where the following equation is satisfied:

$$y^2 = x^3 + ax + b \quad (2.1)$$

The EC security relies on intractability of discrete logarithm problem (ECDLP). The points on the elliptic curve form a group $G$, the ECDLP is to find the integer $d$, for group elements $P$ and $Q$, such that $Q = dP$.

**Elliptic curve digital signature algorithm**

Basically, in Blockchain this is used in transactions to prove that the coins are spending by the legal owners of the secrets. The ECDSA consists of three phases. The first phase is parameter generation, the second is signature generation, and signature verification.

First of all, the sender A need to generate private (32 bytes) and public key as follows (9):

- Choose an elliptic curve E defined over $F_{2^m}$. The number of points should be divisible by a large prime $n$.
- Choose a point $P \in E(F_{2^m})$ of order $n$.
- Choose a statistically unpredictable and unique integer $d$ in the interval $[1, n -1]$.
- Calculate $Q = dP$. 
Then the sender A generates the signature. To sign a message \( m \), the sender should do as follows (9):

- Select a statistically unpredictable and unique integer \( k \) in the interval \([1, n-1]\).
- Calculate \( kP = (x, y) \) and \( r = x \mod n \). If \( r = 0 \), then go to step 1.
- Calculate \( k^{-1} \mod n \).
- Calculate \( s = k^{-1} \{h(m) + dr \mod n \} \). where \( h \) is a secured hash function such as SHA-1 or SHA-2.
- If \( s = 0 \) then go to step 1.
- The signature for the message \( m \) is the integers pair \((r, s)\).

The recipient B should do the following to verify A’s signature \((r, s)\) (9):

- Gain an authentic of A’s public key \((E, P, n, Q)\). Verify that \( r \) and \( s \) are integers in the interval \([1, n-1]\).
- Calculate \( w = s^{-1} \mod n \) and \( h(m) \).
- Calculate \( u_1 = h(m)w \mod n \) and \( u_2 = rw \mod n \).
- Calculate \( u_1P + u_2Q = (x_0, y_0) \) and \( v = x_0 \mod n \).
- The signature is accepted if and only if \( v = r \).

Blockchain cryptography overview

The blockchain is the technology behind cryptocurrencies like Bitcoin, Ethereum and Litecoin and others. Like many other internet technologies, blockchain relies on public key cryptography to protect accounts of users from unauthorized parties. The public and private keys enable users to encrypt information and send it to each other, where the receiver will be able to verify the message authority, and whether it had been altered by an adversary or not (10).

Many previous researchers have been focused on ECDSA weak randomness in blockchain especially in Bitcoin because it is the most popular cryptocurrency. Giechaskiel, Cremers and Rasmussen talked about Bitcoin ECDSA weak randomness in their paper (7) and conclude the consequences for two random number reuse cases. Weak randomness in ECDSA means generating a random number which is not secure enough. If it extended to a broad sense, reusing the same random number also belongs to weak randomness.

In ECDSA the repeated random number may assist the holder of a public key to compute the private key of other user. Both public key and random reuse lead to private key leakage to anyone who has access to the public blockchain. In ECDSA signature generation, as described in the previous section, a random number \( k \) is needed to be selected securely. Suppose that there are two transactions \( tx_i \) \((i = 1, 2)\). The signature is \((r_i, s_i)\) for each \( tx_i \), the public key and private key are \( Q_i \) and \( d_i \) in pair \((Q_i, d_i)\). \( k_i \) is an intermediate variable. In practice, hashes of transactions \( tx_1 \) and \( tx_2 \) are \( h_1 \) and \( h_2 \), respectively. The two cases the attacker could be able to reveal the private key as follows:

**First case : Random number reuse with different public keys**

Suppose that the public keys of two transactions are different but the values of two corresponding \( k \) numbers are identical. Then, we have \( r_1 = r_2 = r \) due to:

\[
(x, y) = k \times P, \quad r = x \mod n.
\]  \hspace{1cm} (2.2)

Both of the two transaction owners could compute the private key of the other one according to:

\[
d_i = r^{-1}(s_i k - z_i) \mod n.
\]  \hspace{1cm} (2.3)

**Second case : Reused Both random number and public key**
If two transactions have the identical public key (i.e., the identical private key $d_1 = d_2 = d$) and a uniformly random $k$, the private key $d$ could be revealed directly by anyone according to the following method:

- Calculate:
  \[ k = (s_1 - s_2)^{-1} \left( h(m_1) - h(m_2) \right) \mod n. \]  
  (2.4)

- Calculate the private key:
  \[ d = r^{-1} (s_1 k - h(m_1)) \mod n. \]  
  (2.5)

In 2013 Thomas Pornin tried to solve this problem by deterministic digital signature generation procedure RFC 6979 (11) stipulates that the number $k$ should be no longer a random number in an implementation of DSA or ECDSA. Instead, there would be a deterministic algorithm calculating $k$ depends on a private key $d$ and a message $m$. With a non-negligible probability, an honest user does not select the same pseudorandom number more than once, or there would not be more than one signature in transactions having the same random number.

In 2020 the researchers Ziyu Wang, Hui Yu, Zongyang Zhang, Jiaming Piao and Jianwei Liu in their paper (12) they systematically revisited the two cases where random numbers are repeated and they evaluated them depending on practical Bitcoin transactions. After analyzing the dataset of Bitcoin transactions in the period between January 2009 and July 2017, they found that there were still about 48% of transactions involving this vulnerability, and 1331 private keys have been compromised. Furthermore, the transactions related to many involved addresses have a similar pattern that gives a clue that a spam transaction attack may take advantage of weak randomness in ECDSA. They also examined mainstream Bitcoin software wallets to see if they are susceptible to weak randomness of ECDSA. Even the result was quite optimistic, an example that one of the influenced addresses leaked in April 2014 is remain in use in August 2017 reflects that the risk of ECDSA weak randomness may not be had enough attention even after its discovery and solution in 2013.

### 3. The ECDSA Improvement

In this section, we enhance the ECDSA scheme by generating signature using two secret keys. This will reduce the probability of revealing secret key in case when random numbers are reused, as we use two private keys $d_1$ and $d_2$ for generating different signatures on different messages. This improved algorithm needs more computation, however, the security has been improved.

#### Parameter generation

- Select an elliptic curve $E$ defined over $F_{2^m}$. The number of points should be divisible by a large prime $n$.
- Select a point $P \in E(F_{2^m})$ of order $n$.
- Select a statistically unpredictable and unique integers $d_1$ and $d_2$ in the interval $[1, n-1]$.
- Calculate $Q_1 = d_1 P$ and $Q_2 = d_2 P$.
- A’s public key is $(E, P, n, Q_1, Q_2)$, A’s secret keys are $d_1$ and $d_2$.

#### Signature generation

- Select a statistically unpredictable and unique integer $k$ in the interval $[1, n-1]$.
- Calculate $kP = (x, y)$ and $r = x \mod n$. If $r = 0$, then go to step 1.
- Calculate $k^{-1} \mod n$.
- Calculate $s = k^{-1} \{ 1 + d_1 r + h(m)d_2 \} \mod n$. Where $h$ is the secure Hash Algorithm such as SHA-1 or SHA-2.
- If $s = 0$ go to step 1.
- The message $m$ signature is the integers pair $(r, s)$.

#### Signature verification
- Gain an authentic of A's public key \((E, P, n, Q_1, Q_2)\). Verify that \(r\) and \(s\) are integers in the interval \([1, n-1]\).
- Calculate \(w = s^{-1} \mod n\) and \(h(m)\).
- Calculate \(u_1 = w \mod n, u_2 = rw \mod n\) and \(u_3 = h(m)w \mod n\)
- Calculate \(u_1 P + u_2 Q_1 + u_3 Q_2 = (x_0, y_0)\) and \(v = x_0 \mod n\).
- The signature is accepted if and only if \(v = r\).

In our algorithm even if the same random number \(k\) and public key is used to generate signatures on two different messages, the equations will be as follows:

\[
s = k^{-1} \{1 + d_1 r + h(m) d_2\} \mod n
\]

Take the form like

\[
s_1 = k^{-1} \{1 + d_1 r + h(m_1) d_2\} \mod n 
\]

\[
s_2 = k^{-1} \{1 + d_1 r + h(m_2) d_2\} \mod n 
\]

By rewriting the equations (3.1) and (3.2) we get

\[
s_1 k = \{1 + d_1 r + h(m_1) d_2\} \mod n 
\]

\[
s_2 k = \{1 + d_1 r + h(m_2) d_2\} \mod n 
\]

And

\[
d_1 r = s_1 k - 1 - d_2 h(m_1) \quad (3.5)
\]

\[
d_1 r = s_2 k - 1 - d_2 h(m_2) \quad (3.6)
\]

Also

\[
d_2 h(m_1) = s_1 k - 1 - d_1 r \quad (3.7)
\]

\[
d_2 h(m_2) = s_2 k - 1 - d_1 r \quad (3.8)
\]

By subtracting (3.3) from (3.4) we get

\[
k = (s_1 - s_2)^{-1} (d_2 (h(m_1) - h(m_2))) \mod n. \quad (3.9)
\]

By subtracting (3.5) from (3.6) we get

\[
d_1 = r^{-1}(k(s_1 - s_2) - d_2 (h(m_1) - h(m_2))) \mod n. \quad (3.10)
\]

By subtracting (3.7) from (3.8) we get

\[
d_2 = (h(m_1) - h(m_2))^{-1}(k(s_1 - s_2)) \mod n. \quad (3.11)
\]

Here, computing the secret key \(d_1\) is impossible without the knowledge of the random number \(k\) and the other secret key \(d_2\). It is also not possible to compute the other secret key \(d_2\) without the knowledge of the random number. Moreover, it is hard for the adversary to compute the random number \(k\) without the knowledge of the secret key \(d_2\).

4. Security Analysis

We simply summarize improved ECDSA security as follows:
Here if the attacker tries to find the random number \( k \) then he needs to find \( d_2 \) first. He also cannot find \( d_2 \) because he does not have \( k \). It is also impossible to find \( d_1 \) because he need to calculate \( d_2 \) first.

- ECDLP: it is impossible, in some way or another, for an attacker, who is trying to forge a signature, as a result of needing to solve the ECDLP in many steps, which is difficult because there is no suitable method to solve these types of problems. Thus, he could not be able to deduce the value of \( d_1 \) and \( d_2 \) from the public keys due to the complexity of the DLP.

5. Conclusion

In this paper, the random number exposing probability is reduced, therefore, the secret key revealing probability also reduced by using two elliptic curve secret keys. Due to the complexity of DLP of the elliptic curves it would be impossible for the attacker to find the secret keys. He also cannot compute the secrets from equations we explained previously, and by knowing two messages with repeated secrets. Because, to find \( k \) the attacker requires \( d_2 \) first, and vice versa. In this improvement there is an increase in the cost of computations, but it is the security of any algorithm which can never be compromised. Now, this algorithm can be applied securely in the applications of the blockchain.

References

1. Nakamoto S, Bitcoin A. A peer-to-peer electronic cash system. Bitcoin–URL: https://bitcoin.org/bitcoin.pdf. 2008.
2. Narayanan A, Clark J. Bitcoin's academic pedigree. Communications of the ACM. 2017;60(12):36-45.
3. Gupta BB. Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives: CRC Press; 2018.
4. Zhou X, Tang X, editors. Research and implementation of RSA algorithm for encryption and decryption. Proceedings of 2011 6th international forum on strategic technology; 2011: IEEE.
5. Marchesan GC, Weirich NR, Culau EC, Weber II, Moraes FG, Carara E, et al., editors. Exploring RSA Performance up to 4096-bit for Fast Security Processing on a Flexible Instruction Set Architecture Processor. 2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS); 2018: IEEE.
6. Liao H-Z, Shen Y-Y. On the elliptic curve digital signature algorithm. Tunghai Science. 2006;8:109-26.
7. Galbraith SD, Gaudry P. Recent progress on the elliptic curve discrete logarithm problem. Designs, Codes and Cryptography. 2016;78(1):51-72.
8. Jyotiyan D, Saxena VP. A Fault Attack for Scalar Multiplication in Elliptic Curve Digital Signature Algorithm. Computing and Network Sustainability: Springer; 2017. p. 283-91.
9. Gunjan VK, Diaz VG, Cardona M, Solanki VK, Sunita KVN. ICICCT 2019 – System Reliability, Quality Control, Safety, Maintenance and Management: Applications to Electrical, Electronics and Computer Science and Engineering: Springer Singapore; 2019.
10. Ouaddah A, Abou Elkalam A, Ouahman AA. Towards a novel privacy-preserving access control model based on blockchain technology in IoT. Europe and MENA cooperation advances in information and communication technologies: Springer; 2017. p. 523-33.
11. Pornin T. Deterministic usage of the digital signature algorithm (DSA) and elliptic curve digital signature algorithm (ECDSA). Internet Engineering Task Force RFC. 2013;6979:1-79.
12. Wang Z, Yu H, Zhang Z, Piao J, Liu J. ECDSA weak randomness in Bitcoin. Future Generation Computer Systems. 2020;102:507-13.