ACCENTS Transactions on Information Security, Vol 2(6)
ISSN (Online): 2455-7196
http://dx.doi.org/10.19101/TIS.2017.26003

Research Article

Weighted threshold ECDSA for securing bitcoin wallet

Pratyush Dikshit[*] and Kunwar Singh
Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, Tamil Nadu, India

Abstract

Bitcoin is a digital currency based on cryptographic algorithms. All the transactions of this currency are recorded and stored in a publically available database called block chain. Since, these transactions are available to everyone; bitcoins must be stored in a secured wallet. These bitcoin wallets can be opened only by its secret key. And if once the secret key of the wallet is lost, it cannot be recovered because of the irreversible nature of bitcoin transaction. To root out this problem, researchers have proposed a solution of threshold signature scheme compatible with bitcoins signature by using elliptic curve digital signature algorithm (ECDSA) providing security policy of shared control of a wallet. In that scheme, the number of players in reconstruction phase was important for recovering the signature. Our contribution is to present a weighted threshold scheme with bitcoins ECDSA signature, in which all the players/participants do not have the same weight. More exactly, a positive weight is associated to each player and the signature can be reconstructed if and only if the sum of the weights of all shares is greater than or equal to a fixed threshold. In our scheme, the number of shares in reconstruction phase is important for recovering the signature. We extend the threshold ECDSA scheme to weighted threshold ECDSA scheme.

Keywords

Cryptography, Shamir secret sharing, Bitcoin, Threshold signature scheme, ECDSA, Bitcoin wallet.

1.Introduction

Bitcoin was introduced in a self-published paper by Satoshi Nakamoto in October, 2008[1, 2]. Bitcoin is a decentralized system which requires no central authority. In recent years, bitcoin has become increasingly accepted and used in many fields in place of physical cash. Bitcoin is a peer-to-peer network of nodes that distribute and record transactions [3]. Bitcoin transaction is a statement that Player 1 (address 1) would like to transfer some bitcoin values v to Player 2 (address 2), signed by Player 1 by his private key. Transactions are verified by network nodes and confirmed in a public distributed ledger called the block chain. The block chain consists of a series of blocks in which each block contains the hashed value of subsequent block. Every bitcoin block contains a set of verified transactions that are collected from the bitcoin broadcast network. It is assumed that the majority of nodes in the bitcoin network are honest. This makes the verification done by the nodes is correct with high probability. More technically, bitcoin is an electronic-cash system based on cryptographic algorithms.

Although, similar in functionality and purposes, bitcoin transactions are different from traditional banking system in many ways, such as-

- **Irreversibility**: Once a bitcoin transaction is updated in the ledger of block chain, that transaction is irreversible even if it is shown later that the transaction is not correct. (e.g., a stolen private key was used).
- **Automation**: Unlike traditional banking transaction system, bitcoin transaction of any size can be fully automated and can be authorized only with a digital signature.
- **Pseudonymity**: Traditional transaction system carries same name of the user for multiple transactions. In bitcoin context, users transact with different addresses that make them pseudonym. And to achieve privacy, bitcoin addresses transactions do not link together.
- **Negligible transaction fees**: On traditional transaction system, merchants can charge transaction fee that can range from 0:5% to 5%, for each transaction made. Bitcoin transaction can be made at a negligible cost or none at all, as bitcoin fees are based on the amount of bitcoin sent.

[*] Author for correspondence

- **Financial fairness**: Inflation can have a huge impact on traditional banking system, but it doesn’t affect bitcoin system at all.
- **Financial freedom**: Bitcoin provides freedom to users regarding their financial status because of decentralization of bitcoin network, unlike traditional banking system which is totally centralized.

One of the key components of bitcoin is Bitcoin Wallet. By name, it seems that ’wallet’ stores bitcoins, but due to the nature of bitcoin transaction, only the details of bitcoin transaction can be saved but not the bitcoin. So, one can describe the bitcoin wallet as a software or a hardware that stores the digital credentials for his bitcoin holdings and allows him to access (and spend) them. All the necessary information to transact bitcoins is stored in a wallet. Bitcoin uses public-key cryptography, in which basically two keys, one public and one private, are generated. Fundamentally, bitcoin wallet is a collection of these keys [4-6].

By using elliptic curve cryptosystem, wallet generates private keys and then derives the corresponding public keys. These public keys are then converted into hashed values. These hash values are the addresses of wallet. Wallet monitors for outputs spent to those addresses, creates and signs transactions spending those outputs, and broadcasts the signed transactions.

Bitcoin wallet can be accessed only by the specific private key. The one, who has that private key, can only open the wallet and construct a transaction. But the main problem is if the private key is lost, it cannot be recovered because there is no authority to control the details of keys. Wallets have been attacked by adversaries many times. And because of irreversibility, these attacks pose many security problems. This results into the decreasing user confidence in bitcoin transaction as well as in bitcoin wallets and could prevent the digital currency from going mainstream. Various solutions of this problem has been proposed by many researchers till date, such as 2-factor authentication, cold storage, multi-signature, etc. But these schemes have serious problems regarding anonymity and privacy, which is undesirable for the goal of bitcoin.

So, the appropriate solution to the problem of bitcoin wallet security, maintaining the requirement of anonymity and privacy, is to have joint control on bitcoin wallet, i.e., combination of multiple participants/players. Before the signature is considered valid, multiple players construct their signatures to form a joint control. Joint control is very much beneficial to eliminate the risk of internal fraud as no one alone gets the full access of signing.

Goldfeder et al. [7] proposed a scheme as a solution to the problem of bitcoin wallet. That scheme uses elliptic curve digital signature algorithm using threshold signature protocol. In threshold signature scheme, the access of constructing a signature is distributed among n players. Each player receives a share of the private signing key. Out of n players, any t or more players are required to sign. Threshold signature scheme distributes the signing power among n players such that any subset of t can jointly sign, but any $t-1$or smaller subset cannot. To comply with threshold signature scheme, elliptic curves can provide various kinds of public key methods that are faster and use much smaller key size, providing equivalent level of security at the same place. That’s why elliptic curve has been chosen to sign a message in distributed (shared) manner.

Our contribution is to provide a scheme which is more practically applicable in organizations. Our scheme is based upon different weightage to different players according to their priorities. The scheme proposed by Goldfeder et al. [4] distributes the secret with equal shares among all the players. But in practical environment, people have different priority according to their posts or ranks in an organization. Our scheme distributes the secret among all the players according to their weightage/priority. And for accessing the wallet, one needs collective sum of shares equal to or more than the threshold value. Our paper extends the threshold ECDSA scheme [4] in order to realize weighted threshold ECDSA scheme. Actually, this can be considered as the weakness of this scheme since each player possesses one or more shares which require more space to store them by the players.

2.Preliminaries and related work

A. Bitcoin

Bitcoin is a decentralized digital currency with no central authority or bank [5]. Bitcoin uses peer-to-peer technology to operate. Bitcoins are assigned to (and redeemed from) transactions and not addresses, but conceptually they can be thought of as belonging to the addresses named in those transactions. Actually, bitcoins do not exist anywhere, even on a hard drive. Someone has bitcoins does not mean that he has some coins collected in digital form. There is no digital bitcoin held in bitcoin addresses, like traditional currency held in bank account. Instead, there are only records of transactions between different addresses, with balance information that can increase and decrease depends on transaction. Every transaction that ever took place is stored in a huge public ledger called the *block chain*. If someone wants to find out the balance of any bitcoin address, he won't find balance at that address, instead he must reconstruct it by looking at the block chain.

Bitcoin transactions are sent from and to electronic bitcoin wallets [3]. Bitcoin wallets are a software abstraction that can manage multiple addresses. Users just see their total balance, and when they want to transfer bitcoins to another address, they specify the amount to be transferred. The bitcoin wallet software chooses the input addresses and change addresses and constructs the transaction. The standard bitcoin wallet implementation generates a new change address for every transaction. The main purpose of choosing this change address is anonymity. Adversary cannot link different transactions of a single user having unique address for each transaction. Everyone on the network knows about a transaction, and one can get the history of transaction back to the point where the bitcoins were produced.

To transfer bitcoins from one address to another, a transaction is constructed that specifies one or more input addresses from which the funds are to be sent, and one or more output addresses to which the funds are to be received [1]. Bitcoin transactions are digitally signed by the private key associated with each input address in order to make the transaction valid. If player 1 wants to send some bitcoins to player 2, the transaction will have three pieces of information:

- An input: record of transaction containing address of player 1 (he received bitcoins from his friend, player 3).
- An amount: the number of bitcoins that player 1 is sending to player 2.
- An output: bitcoin address of player 2.

To send bitcoins, one needs two things: a bitcoin address and a private key. A bitcoin address is a randomly generated alphanumeric sequence. The private key is another alphanumeric sequence. The private key is kept secret, unlike the bitcoin address. Each output of a bitcoin transaction refers to the input of the next transaction. That's why, it is required to spend whole output at a time. If someone wishes to spend a part of the output that was received in a previous transaction. This can be done by means of a change address where one mentions his own address as one of the output addresses of that transaction. For example, if player 1 received 7 bitcoins in a transaction and wants to transfer 5 bitcoins to player 2, he constructs a transaction in which he transfers 5 to the address of player 2 and the remaining 2 to his own changed address. Bitcoin address can be thought of as a safe deposit box with a transparent glass front. Everyone can see what is inside the safe, but only the person having private key of that safe can unlock it. When player 1 wants to send bitcoins to player 2, he uses his private key to do signature on a message with the input (the source of transaction(s)), amount (number of bitcoins), and output (address of player 2). Then, player 1 sends the mentioned amount from his bitcoin wallet to the bitcoin network. From there, some specified nodes on the network verify the transaction and putting it into a transaction block. New block is added into block chain through a rigorous competition among nodes which are actually, called bitcoin transaction verifiers. This competition requires each node or set of nodes to solve a puzzle called proof-of-work. This activity is called mining and the miners are rewarded with transaction fees and newly created bitcoins. If the block is valid, then the new block is accepted as the head of the block chain.

Miners work on a distributed consensus system that is used to confirm already constructed transactions by including them in the block chain. People send bitcoins to each other over the bitcoin network, but a system needs to keep a record of all these transactions so that people would be able to keep track of who had paid what. The bitcoin network deals with this by aggregating all of the transactions made during a set period (around 10 minutes) into a list, called a block. Then it is the responsibility of miners to confirm those transactions, and update them into a general ledger. As soonas a particular transaction is added to the block chain, it is called successful transaction.

B. Existing methods for the security of bitcoin wallet

There are some already existing methods for the security of bitcoin wallet [1, 7] used in current scenario. Some of them are explained below:

- **2-factor authentication schemes**: [3] Along with the password, the user must provide at least one more authentication step, either by replying to an email or using a smartphone authentication app or using a messaging service on cell phone. But, this will be accomplished at the cost of anonymity and privacy of users.

- **3-factor authentication scheme**: Users can take the help of methods that include biometrics. But again, anonymity is compromised.
- **Cold storage**: Cold Storage signifies keeping the main bitcoin wallet on an offline device i.e., the device which is not connected to the Internet, and moving only the funds needed for daily expenses to online storage, i.e., Hot Storage. Often it seems too much of a hassle, and obviously, it takes much more time to transact which is not in favor of goals of bitcoin.
- **Multi Signature** [4, 7]: Multi-signature (multisig) wallets offer a better solution. A multisig transaction requires the agreement of the required number of authorizedsignatories, for example a 2-of-3 transaction will require two signatories out of three. However, the paper shows that multisig transactions have some serious usability problems, and anonymity and confidentiality drawbacks. Three of the most prominent techniques to preserve the anonymity are Mixcoin, CoinJoin, and the use of change addresses. The problem is none of these techniques are compatible with multi signatures, while they all are compatible with threshold signatures. Suppose a player uses multi signature-based security and makes a purchase at an online store. Then the spending address (address of player) and change address will all have the same t-of-n access control structure, whereas the destination address (address of store) most likely will not. This provides clues to adversaries to link player's input and output addresses. On the other hand, with threshold signatures, change addresses will not be linkable by the adversaries when sending bitcoin to any regular (single-key) address or other threshold address.

Example: In a 2-out-of-3 threshold signature scheme, the ability to construct a signature is distributed among different devices (for example an office computer, a home computer and a smartphone), and each device receives a share of the private signing key. Here, any two devices together can do signature. So, a single compromised device cannot put the money at risk.

3.Secret sharing

A.Shamir secret sharing

Liu considered the following problem. Eleven scientists are working on a secret project. They wish to lock up the documents in a cabinet so that the cabinet can be opened if and only if six or more of the scientists are present. What is the smallest number of locks needed? That is the smallest number of keys to the locks each scientist must carry? [8] Answer of these problems is 462 locks and 252 keys per scientist. Obviously, this is not a practical solution to these kind of problems. Shamir's secret sharing scheme gives the solution for generalization of above problem. Shamir's secret sharing considers the secret as some data $D$and divides the data D is revealed from $t-1$ pieces or less. Shamir's secret sharing is based on the following theorem.

Theorem:
Given t points in the 2-dimensional plane $(x_1, y_1), \ldots, (x_t, y_t)$ with distinct $x's,$ there is one and only one polynomial of degree $t-1$ such that $q(x_i) = y_i$for all i.

a) Shamir's Sharing Protocol: Our goal is to create $u -$ secret shares of the secret s such that at least t shares are required to compute D.

1) Dealer D pick a random $t-1$ degree polynomial $q(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1}$ which $a_0 = s$.Here all coefficients $a_i (0 \le i \le t-1)$ are from field $F_P{:}\, prime\ p$).
2) Dealer D computes $q(1), q(2), \ldots, q(u)$and secretly distributes each player j the share $q(j)$. Hence the shares are denoted as $q(1), q(2)., \ldots, q(u)$.

From these $t-$ points we can construct polynomial $q(x)$ of degree $t-1$ and can find the secret $s = q(0)$. One can construct polynomial by using Lagrange interpolation.

b) Lagrange polynomial.: Given t points in the 2-dimensional plane $(x_1, y_1), \ldots, (x_t, y_t)$ with distinct $x's,$ then the unique polynomial passing through these points in the Lagrange form is a linear combination $q(x) = L(x) = \sum_{i=1}^{t} y_i l_i(x)$ of the Lagrange basis polynomials:

$$l_i = \prod_{1 \le m \le t, m \ne i} \frac{(x - x_m)}{(x_i - x_m)} = \frac{(x - x_1)}{(x_i - x_1)} \cdots \frac{(x - x_t)}{(x_i - x_t)}$$

c) Properties of Shamir's secret sharing

1) **Perfect security:** Adversary with knowledge of $t-1$ shares or less cannot find any information regarding secret.
2) **Ideal:** Size of each share is exactly the same as the size of the secret.
3) **Extendable:** By calculating the polynomial in additional points, additional shares may easily be created.

4) Homomorphic property:

- If we add/multiply a constant to all secret shares (y-values) then this constant will be added/multiplied to the secret to get new secret.
- Suppose we have two secrets s and t. Their corresponding shares are $f(1), \dots, f(n)$ for polynomial $f(x)$ and $g(1), \dots, g(n)$ for polynomial $g(x)$. Now we define j^{th} share as $f(j)+g(j), (j \in [1 \dots n])$. New secret will be $s+t$ for the new function $h(x)=f(x)+g(x)$since $h(0)=f(0)+g(0)$.

B. Secret Sharing in threshold signature manner

Threshold secret sharing is a scheme to distribute a secret value into shares that can be given to different players, with the following two properties [4]:

(1) any subset of shares of size equals or more than threshold can reconstruct the secret
(2) any subset of shares smaller than this threshold together yields no information about the secret.

The secret can be framed as a polynomial of degree $t-1$and a randomly selected point on the polynomial is given to each of n players, any $t-of-n$ can be used to reconstruct the polynomial using Lagrange Interpolation.

Secret sharing schemes are fundamentally one-time use. In that, once the secret is reconstructed, it is known to all the players who took part in reconstructing it. A more general approach is threshold cryptography, where a threshold number of shares would be needed, out of all the shares, to reconstruct the secret. A (t,n)- threshold signature scheme distributes signing power to n players. Any group of at least t players can generate a signature, whereas a group of less than t cannot. A key property of threshold signatures is that the private key need not ever be reconstructed. Even after repeated signing, adversary cannot learn anything about the private key that would allow them to produce signatures without a threshold sized group.

C. Sharing Secret: with dealer vs. without dealer

Here, dealer means an authorized and trusted system who manages the distribution of secret. Here, it is described how shares are generated and distributed. One way to do this is by using a trusted dealer who has a randomly generated key [9]. He generates the shares and distributes them to each player. But, this scheme has a major drawback of dependency on a single point for all shares. A more sophisticated scheme eliminates the use of trusted dealer and allows the players to generate shares of a key in a distributed manner without ever constructing the key in the process.

Both the approaches have their own strengths and weaknesses. None of them is strictly better than the other. Although having a trusted dealer is a weakness, but in some cases it is strictly necessary. A dealer less protocol allows the parties to generate a new key, but it does not allow players to distribute an already existing key. In the Bitcoin context, if someone already has an address and later he wants to add threshold security to that address, he needs a trusted dealer protocol to generate shares from the existing secret key.

However, when generating a new address, a dealer less protocol is generally superior. This explains the need of the proposed scheme to include both approaches in two different levels.

D. Joint random secret sharing (without a dealer) [JRSS]

JRSS provides freedom to the players to choose their secrets on their own. This scheme doesn't require a dealer or a third party to generate shares of a secret. Each player chooses secret using Shamir Secret Sharing. Then all $n-$ players distribute the shares to all other players. Finally,t (threshold) players with their shares can compute combined secret key. This protocol is free from single point of failure. It also verifies the correctness and consistency of shares of the players without sharing original secret to anyone. The proposed solution is preferable when generating a new address.

Firstly, all the players agree on the following setup [9]. Given an elliptic curve Edefined over field Z_p(prime p). The base point $G \in EZ_p$ of large cyclic sub-group of order $r\,(prime\,r)$ that divides the number of points in $E(Z_p)$. Given a threshold t and the total number of players $n \geq 2t+1$.

Now, each player $P_i$does the following:

A. Secret Sharing among all the players

1) Selects a random polynomial $f_i(x)$ of degree t, such that $f_i(0)$ is the secret value. For example, $f_i(x) = a_0^i + a_1^i x + \cdots + a_t^i x^t$. Here a_i^0 is the secret of player i, where $i \in [1, \ldots, n]$
2) Secretly sends $f_i(j)$ to player $P_j, \forall j = [1, \ldots, n]$
3) Compute $y_i = a_k^i G, \forall k = [0, \ldots, t]$.
4) Compute $z_i = f_i G, \forall\, j = [1, \ldots, n]$

B. Verification of shared secret by all the players

1) For verification purpose, each player $P_i$broadcasts y_i and z_i to all other players.

2) Each $P_{j \neq i}$ verifies that $\sum_{k=0}^{t} j^k y_i = f_i(j)G$ or not, and that $f_i(j)G$ is consistent with his share. For example, player 2 verifies the share of player 1 on a polynomial of degree $t = 2$. Player 2 computes $f_2(1) = (a_0^2 + a_1^2 + a_2^2)G$. Player 2 also computes $\sum_{k=0}^{2} 1^k a_k^2 G$. Since both the values come out to be equal, player 2 accepts the signature as valid.

If any player is found guilty, then the decision would be taken on the basis of majority voting. Once the above scheme is completed successfully, each player P_i can safely calculates his share as $\sum_{j=1}^{n} f_j(i) mod\, r$.

The combined secret key of n players is $a_0^1 + a_0^2 + \cdots + a_0^n$. Out of n players, any t -players with their shares can compute the combined secret key.

E. The degree reduction protocol

This protocol shows that a polynomial of degree $2t$ can be reduced to polynomial of degree t while keeping the free coefficient unchanged [11].

Let $h(x) = a_0 + a_1 x + \cdots + a_{2t} x^{2t}$ and let $s_i = h(b_i) = f(b_i) g(b_i)$, for $i = 0, \ldots, n-1$ be the shares of $h(x)$. Each player P_i holds an s_i. So, $h(x)$ can be truncated to be $k(x) = a_0 + a_1 x + \cdots + a_t x^t$ and $r_i = k(b_i)$ for $i = 1, \ldots, n-1$. This can be proved as follows:

CLAIM: let $S = (s_0, \ldots, s_{n-1})$ and $R = (r_{0,\ldots,} r_{n-1})$ then there exists a constant $n \times n$ matrix A such that $R = S.A$

Now

Let H be the n vector $H = (h_0, \ldots, h_t, \ldots, h_{2t}, 0, \ldots, 0)$ and let K be the n-vector $K = (h_0, \ldots, h_t, 0, \ldots, 0)$. Let $B = (c_{i,j})$ be the $n \times n$ (Vandermonde) matrix, where $c_{i,j} = b_j^i$ for $i, j = 0, \ldots, n-1$. Furthermore, let P be the linear projection $P(x_0, \ldots, x_{n-1}) = (x_0, \ldots, x_t, 0, \ldots, 0)$.

We have

$$H.B = S$$
$$H.P = K$$
$$K.B = R$$

Since B is not singular (because $b_i s\ are\ distinct$), we have $S(B^{-1}PB) = R$, but $A = B^{-1}PB$ in some fixed constant matrix. This proves our claim.

F. Standard ECDSA

Firstly, the usual ECDSA signature generation scheme is presented below.

Parameters

Given an elliptic curve E over Z_p (prime p) [2]. Given base point G of order n, the private key d, and the message m to be signed.

Signature generation

1) Compute $e = SHA - 1(m)$. Convert e to an integer using the method in ANSI X9.62. With reference to ANSI X9.62, given an input message, SHA-1 gives the output in the form of hexadecimal which can be further converted into integer easily.
2) Select a random integer k such that $1 \leq k \leq n - 1$.
3) Compute $(x_1, y_1) = kG$.
4) Convert x_1 to an integer using the method in ANSI X9.62. Compute $r = x_1\, mod\, n$. If $r = 0$, return tostep 2.
5) Compute $s = k^{-1}(e + dr)\, mod\, n$. If $s = 0$, return to step 2.
6) The signature for m using the key d is the pair(r, s).

4.Proposed scheme

Now, the proposed scheme for weighted threshold signature using ECDSA is explained below:

Our Scheme: Weighted Threshold ECDSA Signature Our proposed scheme is similar to the scheme explained by Goldfeder et al. [4]. The key difference is that, all the players get equal shares in the scheme proposed in [4], whereas, in this scheme, each player is given one or more shares of the secret key according to his weightage/priority$w$, that makes this scheme more realistic. The sharing of secret among all players is done on the basis of Shamir's Secret Sharing Scheme [10], but with different weights. For example, player P_1 with weightage $w_1 = 1$ gets share $g(x_1^1)$, player P_2 with weightage $w_2 = 3$ gets the share $g(x_1^2)$,$g(x_2^2)$,$g(x_3^2)$, likewise player P_l with weightage w_l gets the share $g(x_1^l)$,$g(x_2^l)$,…, $g(x_{w_l}^l)$. Here, x_j^lmeans l^{th}player has

j different shares of secret. We assume that all x_j^l are different. One more difference is that the scheme proposed in [4] requires (threshold) t players out of (total) n players, whereas this scheme requires j shares out of n shares of all the players. The total number of players (m) may be less than total number of shares (n).
$w_1 + \cdots + w_m = n$, where $m \leq n$.

This scheme is divided into two phases - setup phase and signature generation phase. The set up phase is totally handled by a dealer to distribute the shares of each player secretly.

Signature generation phase is further divided into two sections. First section calculates the value of first part of signature i.e., r. This section does not include the role of secret shared by the dealer. Every player j for every share k computes $f_k^j(1)$, $f_k^j(2)$, …, $f_k^j(n)$ and gives to players having share 1, share 2, ..., share n. For example, suppose a player P_2has weightage 3, he selects 3 random polynomialssuch that $f_1^2(x) = (a_0^2 + a_1^2 x + \cdots + a_t^2 x^t)$, $f_2^2(x) = (b_0^2 + b_1^2 x + \cdots + b_t^2 x^t)$, $f_3^2(x) = (c_0^2 + c_1^2 x + \cdots + c_t^2 x^t)$. Then he calculates $f_k^2(1)$, $f_k^2(2)$, …, $f_k^2(n)$and distribute to players having share 1, share 2, ..., share n. The free term a_0^2, b_0^2, $c_0^2$is the secret of player 2. This is done almost in the same manner as the dealer did in the previous phase, but without any intervention of the dealer in this phase. Dealer does not have any information about the value of individual shares of players. The second section of this phase calculates the value of second part of signature i.e., s with the help of secret shared by the dealer. Combining both the parts, we get the required signature i.e. (r,s). Signature can be constructed by any number of players whose combined shares equals or more than the threshold (t) value.

The base condition for this scheme is that total number of shares must be twice more than the threshold value, i.e., $n \geq 2t + 1$.

Parameters

An elliptic curve E defined over field Z_p (prime p). A base point $G \in E(Z_p)$ is a generator of large cyclic sub group of order r (prime r) that divides the number of points in elliptic group $E(Z_p)$.

Setup Phase

A dealer D selects a random polynomial g of degree $t - 1$. The dealer calculates shares of all the players. The player(s) having minimum weightage possess a single share and those of higher weightage possess more than one share. Each player P_l gets w_l share as $g(x_j^l)$, where $1 \leq j \leq w_l$and w_l is the weightage of the player l. Shares are numbered from 1 to n, the value of i_{th} share for player k issued by dealer is defined as d_i. Then the dealer distributes shares to corresponding players secretly. For example, player P_1 with weightage $w_1 = 2$ gets share $g(x_1^1)$and $g(x_2^1)$, likewise player P_l with weightage w_l gets shares as $g(x_1^l), g, \dots, g(x_{w_l}^1)$.

Public Key $Q = \sum_{i=1}^{n} d_i G$

Once this is complete, players can do signature on message m as follows:

Signature Generation

First Part of Signature: This part of signature is computed without using the shares issued by the dealer.

1) Since $m$is public, each player computes $e = SHA - 1(m)$ and then $e$is converted to an integerusing ANSI X9.62.

2) Players run Joint Random Secret Sharing without a Dealer [Refer section III.D] as follows:

- •Each player P_l with weightage w_l selects random polynomial(s) $f_j^l(x)$ each of degree t,such that his chosen secret is the free term of the polynomial, where $1 \leq j \leq w$. for example, suppose P_2 has weightage 3 so, $f_1^2(x) = (a_0^2 + a_1^2 x + \cdots + a_t^2 x^t)$, $f_2^2(x) = (b_0^2 + b_1^2 x + \cdots + b_t^2 x^t)$, $f_3^2(x) = (c_0^2 + c_1^2 x + \cdots + c_t^2 x^t)$ the free term a_0^2, b_0^2, c_0^2 is the secret of player 2.

- •Compute $k_i = \sum_{j=1}^{w_1} f_j^1(x_i) + \sum_{j=1}^{w_2} f_j^2(x_i) + \dots + \sum_{j=1}^{w_i} f_j^m(x_i)$,where k_i is the shared secret of player having share i.

3) Each player having shared secret i computes the value of the Lagrange basis polynomial

$$\mathrm{b_i(x)} = \prod_{\mathrm{j \neq i, j \in B}} \frac{\mathrm{j - x}}{\mathrm{j - i}}$$

But in order to consider the free term of polynomial, put x = 0 in above equation

$$\mathrm{b_i}(0) = \prod_{\mathrm{j \neq i, j \in B}} \frac{\mathrm{j}}{\mathrm{j - i}}$$

B is the set of indices of any number of shares out of nshares.

4) Each player having share i computes $y_i = b_i k_i$.
5) Each player having share i broadcasts $V_i = y_i G = b_i k_i G$. It is very hard to find out y_i from $V_i$because of elliptic curve discrete logarithm problem.
6) According to the homomorphic property of Shamir secret sharing, if we add/multiply a constantto all secret shares then this constant will be added/multiplied to secret to get new secret[Refersection III.A]. All players can now compute $(x_1, y_1) = kG = \sum_{i \in B} V_i$, where $k = a_0^1 + a_0^2 + \cdots + a_0^n \bmod n$.

7) Convert x_1 to an integer using the method in ANSI X9.62. Then, compute $r = x_1 \bmod n$. If $r = 0$, then return to setup phase.

Second part of signature
1) It is required to compute $k^{-1} \bmod n$ from shares of k without revealing any information about k.
2) The players run the Joint Random Secret Sharing protocol [Refer section III.D] to distribute a share c_i of c to each player having share i, where $c_i = \sum_{a=1}^{w_1} f_a^1(x_i) + \sum_{a=1}^{w_2} f_a^2(x_i) + \ldots + \sum_{a=1}^{w_i} f_a^m(x_i)$.
3) A simple multiplication protocol can be employed here, such that$u_i = c_i k_i$. But the result is automatically a share on a polynomial of degree $2t$. Moreover, the resulting polynomial is not completely random which may weaken the security of the scheme. Consequently, the Joint Random Zero Secret Sharing is employed to add a sort of randomization to the process.
4) Players for each share run the Joint Random Zero Secret Sharing. This scheme is a special case of JRSS. In this scheme each player for each share chooses his secret as zero. Hence for all the players in this scheme must agree with $a_0 G = 0 \; \forall i = [1, \ldots, n]$. The stepwise procedure of this scheme as follows:

- Each player having share i selects a random polynomial f of degree 2t subject to zero as its free term, i.e., $f(x) = 0 + a_1 x + a_2 x^2 + \cdots + a_t x^t + \cdots + a_{2t} x^{2t}$. Each player with weightage w selects random polynomial(s) $f_j^i(x)$, where $j \in [1, \ldots, w_i]$.
- z_i is the secret of player having share$i$, where$z_i = \sum_{a=1}^{n} f_a(i)$.

5) Now, each player having share i locally computes and broadcasts $v_i = k_i c_i + z_i$.
6) Player can interpolate the polynomial of degree $2t$ and compute v. And then, all players can compute $v^{-1} \bmod n$.
7) Each player having share i computes his share of k^{-1} as $l_i = c_i v^{-1}$ on a polynomial of degree $2t$. So, apply secure degree reduction protocol [Refer section III.E] to reduce the degree of polynomial from $2t$ to t.
8) Players now compute the shares of $w_i = d_i k_i^{-1}$ over a degree $2t$ polynomial by multiplying their shares of d and k^{-1}. They run secure degree reduction protocol, to reduce the degree of the polynomial back to t [Refer section III.E].
9) By applying homomorphic property of Shamir Secret Sharing [Refer section III.A], each player now computes the share $s_i = k_i^{-1} e + r w_i = k_i^{-1} e + r(d_i k_i^{-1}) = k_i^{-1}(e + d_i r)$ Players, then, can run secure degree reduction protocol to reduce the degree of the polynomial sharing s back to t [Refer section III.E].
10) Players can now interpolate their shares of s to recover $s = k^{-1}(e + dr)$. If $s = 0$, then return to setup phase.
11) Hence, (r, s) is the required signature on message m using key d.

Signature verification
Verification to the signature obtained in threshold ECDSA scheme is almost the same as that of standard ECDSA verification procedure. For authenticating signature, one must have a copy of dealer's public key Q. One can verify Q is valid curve point as follows:

1) Check that Q is not equal to the identity element O
2) Check that Q lies on the curve.
3) Check that $n \times Q = O$. Here, n is the order of elliptic curve chosen by dealer.

After this, follow these steps:
1) Check that the two parts of signature obtained above,r and s are integers and are integers and $r, s \in [1, n-1]$.
2) Calculate $\mathrm{SHA}-1\ (m)$ and convert bit string to integer e by using ANSI X9.62.
3) Calculate $w = s^{-1} \bmod n$
4) Calculate $u_1 = ew \bmod n$ and $u_2 = rw \bmod n$.
5) Calculate the curve point $(x_1, y_1) = u_1 G + u_2 Q$.
6) The signature is valid if $r \equiv x_1 \bmod n$. Invalid, Otherwise.

This scheme is of more practical use than that of explained by Goldfeder, et al[4], Gennaro, et al[6], Goldfeder, et al.[7]. Goldfederet al.[4] provided a scheme as all the players were supposed to have equal weightage/priority. But practically, users keep priority to various devices/systems according to their reliability, availability and convenience.

5.Conclusion

We have extended the threshold ECDSA in order to address weighted threshold ECDSA. We explained that our proposed scheme can be implemented in any organization according to its internal hierarchical level. We have given a scheme on how to use weighted threshold ECDSA scheme to realize Bitcoin wallets. Our technique has the potential to dramatically improve bitcoin security. The only drawback in this scheme is that it requires more space for storing secret share of each Player. Here, each player possesses one or more shares and each share is represented by a polynomial value. This requires a lot of space to store shared values corresponding to each player. The concern is to limit the space required by each player to store his secret share. Our future work includes the distribution of shares to all the players in such a way that all the players get a single share with different values. This gives an advantage of storing one polynomial no matter what the Weightage the player possesses.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

References

[1] Androulaki E, Karame GO, Roeschlin M, Scherer T, Capkun S. Evaluating user privacy in bitcoin. In international conference on financial cryptography and data security 2013 (pp. 34-51). Springer Berlin Heidelberg.

[2] Barber S, Boyen X, Shi E, Uzun E. Bitter to better—how to make bitcoin a better currency. In international conference on financial cryptography and data security 2012 (pp. 399-414). Springer Berlin Heidelberg.

[3] Transactions. https://en.bitcoin.it/wiki/Transactions Accessed 11 March 2014.

[4] Goldfeder S, Bonneau J, Felten EW, Kroll JA, Narayanan A. Securing bitcoin wallets via threshold signatures. Princeton University, http://www. cs. princeton. edu/∼ stevenag/bitcoin threshold signatures. pdf. 2014.

[5] Nakamoto S. Bitcoin: a peer-to-peer electronic cash system.

[6] Gennaro R, Goldfeder S, Narayanan A. Threshold-optimal DSA/ECDSA signatures and an application to bitcoin wallet security. In international conference on applied cryptography and network security 2016 (pp. 156-74). Springer International Publishing.

[7] Goldfeder S, Gennaro R, Kalodner H, Bonneau J, Kroll JA, Felten EW, Narayanan A. Securing bitcoin wallets via a new DSA/ECDSA threshold signature scheme. 2015.

[8] Singh K, Rangan CP, Banerjee AK. Lattice-based identity-based resplittable threshold public key encryption scheme. International Journal of Computer Mathematics. 2016; 93(2):289-307.

[9] Ibrahim MH, Ali IA, Ibrahim II, El-sawi AH. A robust threshold elliptic curve digital signature providing a new verifiable secret sharing scheme. In IEEE Midwest symposium on circuits and systems 2003 (pp. 276-80). IEEE.

[10] Shamir A. How to share a secret. Communications of the ACM. 1979; 22(11):612-3.

[11] Ben-Or M, Goldwasser S, Wigderson A. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In proceedings of the twentieth annual ACM symposium on theory of computing 1988 (pp. 1-10). ACM.

This paper is selected from proceedings of National Workshop on Cryptology-NWC 2016 organized at JNN College of Engineering Shimoga, Karnataka, India during 11-13, August 2016.