# Bitcoin Cryptography - Elliptic Curve Digital Signature Algorithms

| | |
|---|---|
| **Author** | josh |
| **Date** | 2020-12-26 14:52:11 |

# Overview

At the heart of Bitcoin, Ethereum, and all *crypto*currencies lies the fascinating art of *cryptography*. In a previous tutorial, we discussed the basics of *hashing algorithms* - one way functions used in key parts of the Bitcoin system such as address generation and mining. Now, let's discuss another important type of cryptography used in cryptocurrency applications known as *Elliptic Curve Digital Signature Algorithms.*

# ECDSA - Bitcoin's "Signature" Crypto

## The Necessity of Public Key Cryptography

Bitcoin does not rely on trust for transactions, not at all. Users can send funds to anyone, anywhere without having to go through a trusted intermediary like a bank or government payment processor. So how can we *prove* that a user actually has money they are trying to send and prevent fraud? This is done through the beautiful science of elliptic curve cryptography, a subset of what is known as public key cryptography or (tangentially) asymmetric encryption.

Public key cryptography uses mathematics to prove ownership of a specified *public key* without ever revealing the secret *private key*. This is done through what is called a *digital signature.*

Let's say you want to share a very important message with the world - "I love Bitcoin". You want to share this far and wide, and you want to ensure that nobody can tamper with that message. It would stink if the message got intercepted by an attacker and changed to "I don't think Bitcoin is very good". So what you can do is *digitally sign* that message using public key cryptography!

You generate a private, public key pair and distribute your public key far and wide. This exercise assumes that everyone has your legitimate public key and not an imposter (key agreement is a whole other topic beyond the scope of this tutorial). You use your private key to *sign* the message. By providing the message "I love Bitcoin", and your digital signature, they can use mathematics to prove that you are in fact the rightful owner of the private key that signed the message ***without ever knowing the private key*.**

## Public Key Cryptography and Bitcoin

This digital signature example is neat, but how does this all fit in with cryptocurrencies? Bitcoin uses this exact sort of digital signature algorithm with respect to addresses and transaction signing.

Every Bitcoin wallet is fundamentally a *private key store*. Your wallet contains a bunch of private keys - in a modern wallet these are derived from the seed phrase generated on wallet creation, but used to be generated in an entirely random fashion and stored in a wallet data file.

These private keys are used to derive public keys using Bitcoin's specified Elliptic Curve algorithm known as secp256k1. The public keys are then hashed and encoded in a special format to give the final address you receive coins at.

So let's say you receive 0.5 BTC at your wallet address. You would now like to send that 0.5 BTC off to your friend as a gift, how generous! Your wallet constructs a *transaction* that specifies her address as the receiver. Your wallet then provides a *digital signature* for that transaction using your private key, and includes your address public key in the transaction data. So we have now given the Bitcoin network a *message, public key, and digital signature.* ***We have proven to the network we are the rightful owner of the 0.5 BTC we are trying to send!***

## Why Elliptic Curves?

Elliptic curves are not the only type of public key cryptography. One of the most well known alternatives is called *RSA*, and it is also widely used. However, there are certain advantages to elliptic curves that make them very well suited to systems like Bitcoin.

The primary reason is that elliptic curves require far smaller keys for the same level of security as RSA. To achieve an equivalent level of security, an RSA key of 3072 bits would only require an EC key of 256 bits. This may not seem like much, but it is within the context of cryptocurrencies. *Every single transaction* is added to the blockchain, and *every single transaction* requires one or more digital signatures. The 2816 bit savings adds up tremendously over time.

## The Security of Elliptic Curve Cryptography

This is a much more secure way of spending money than legacy systems such as credit cards! When you use a credit card, you give each and every merchant you are doing business with private information (your credit card number) that can be used to draw from your account. You have to trust that the merchant won't steal more from you than you authorized, and trust that they will not leak your credit card information to hackers. And guess what, they often do.

In contrast, you never have to reveal any secret information to conduct business via Bitcoin. The receiver gives you their public address, and you simply provide a digital signature and public key from *your* address to prove you can spend the funds in your "account". *Secret information is never, ever shared* in a Bitcoin transaction.

There are known vulnerabilities in this system, however. It's not perfect. When constructing an elliptic curve digital signature, one must use a secret and *truly random* value called a *k value*. This number must be generated using a cryptographically secure random number generator (just like a private key), and can never be reused. Some poorly implemented Bitcoin wallets have accidentally reused k values for digital signatures, and thus leaked the user's private keys to attackers. I have a code project demonstrating this vulnerability called NotOkReuse written in Rust, if you are curious about this topic.

# Elliptic Curves - Security without Shared Secrets

Elliptic curve cryptography is a critical part of the Bitcoin system, as it provides the means for securing transactions without trust. Instead, we rely on the provable mathematics of elliptic curves and public key cryptography to secure transactions. By using these mathematical methods instead of trusted institutions, users can participate in a truly decentralized and peer-to-peer protocol for money!