Real World Applications that use Elliptic Curve Cryptography

Francisco Rodríguez-Henríquez
CINVESTAV-IPN

Havana 2016 - September 2016

A brief introduction

1. Elliptic curve cryptography (ECC) is used in practice to instantiate public-key cryptography protocols, such as:
    - Digital signatures
    - Key agreement
2. The benefits of using ECC are:
    - Small key sizes
    - Efficient implementations
3. An important number of real world applications use ECC nowdays.
    - Bitcoin
    - SSL/TLS
    - WhatsApp
    - e-passport

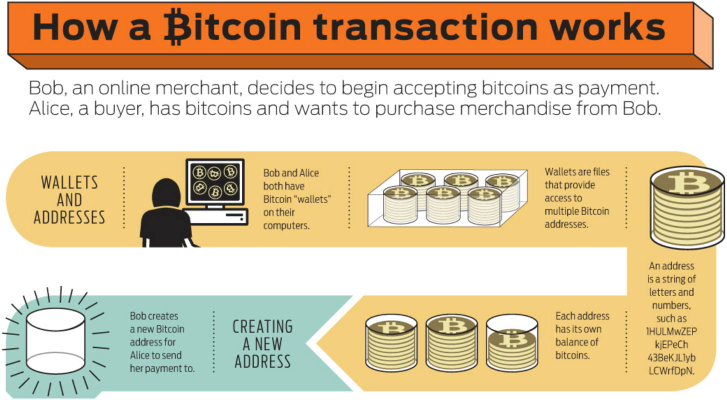
Case 00: Bitcoin

1. In November 2008, a paper was posted on the internet under the name Satoshi Nakamoto titled bitcoin: A Peer-to-Peer Electronic Cash System.
2. According to Nakamoto, bitcoin is a digital currency which allows online payments from one party to another without going through a financial institution.
3. In January 2009, bitcoin network came into existence.
4. Nakamoto mining the first block of bitcoins ever (known as the genesis block).
5. As of 6 February 2016, there were 15.2 millions bitcoins circulation of a capped total of 21 millions.
6. A bitcoin dollar value is arround $573.39.

Case 00: Bitcoin technical details

1. A Bitcoin Block Chain is a journal of all the transactions ever executed.
2. A User Account is tipically a ECDSA private key.
3. A bitcoin transaction is realized by attaching a digital signature of the hash of:
   - The previous transaction
   - The public key of addresses user
4. Each block in the journal contains the SHA-256 hash of the previous block, hereby chaining the blocks together starting from the so-called genesis block.

Case 00: Bitcoin



How a ₿itcoin transaction works

Bob, an online merchant, decides to begin accepting bitcoins as payment. Alice, a buyer, has bitcoins and wants to purchase merchandise from Bob.

WALLETS AND ADDRESSES

Bob and Alice both have Bitcoin "wallets" on their computers.

Wallets are files that provide access to multiple Bitcoin addresses.

An address is a string of letters and numbers, such as 1HULMwZEPkjEPeCh43BeKJL1ybLCWrfDpN.

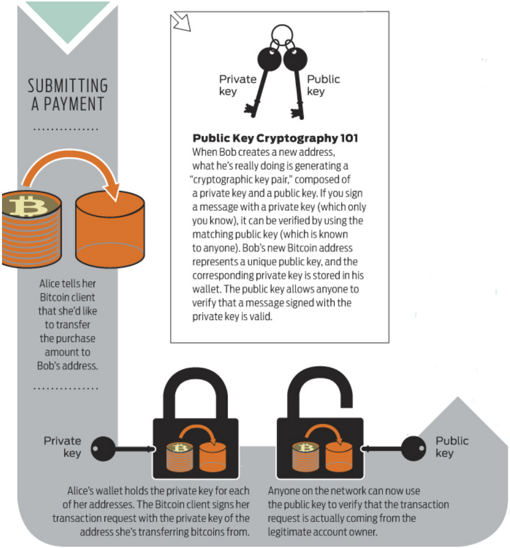
Each address has its own balance of bitcoins.

CREATING A NEW ADDRESS

Bob creates a new Bitcoin address for Alice to send her payment to.

Source: spectrum.ieee.org

Case 00: Bitcoin



SUBMITTING A PAYMENT

Alice tells her Bitcoin client that she'd like to transfer the purchase amount to Bob's address.

Public Key Cryptography 101

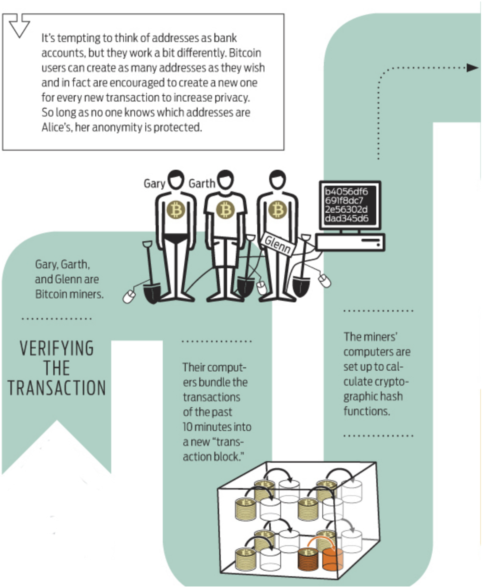
When Bob creates a new address, what he's really doing is generating a "cryptographic key pair," composed of a private key and a public key. If you sign a message with a private key (which only you know), it can be verified by using the matching public key (which is known to anyone). Bob's new Bitcoin address represents a unique public key, and the corresponding private key is stored in his wallet. The public key allows anyone to verify that a message signed with the private key is valid.

Alice's wallet holds the private key for each of her addresses. The Bitcoin client signs her transaction request with the private key of the address she's transferring bitcoins from.

Anyone on the network can now use the public key to verify that the transaction request is actually coming from the legitimate account owner.

Source: spectrum.ieee.org

Case 00: Bitcoin



It's tempting to think of addresses as bank accounts, but they work a bit differently. Bitcoin users can create as many addresses as they wish and in fact are encouraged to create a new one for every new transaction to increase privacy. So long as no one knows which addresses are Alice's, her anonymity is protected.

Gary, Garth, and Glenn are Bitcoin miners.

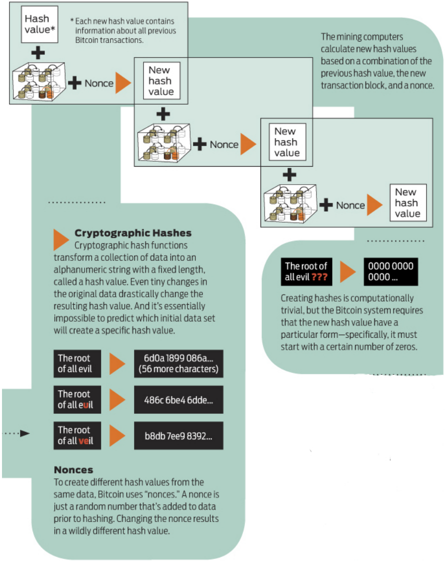
VERIFYING THE TRANSACTION

Their computers bundle the transactions of the past 10 minutes into a new "transaction block."

The miners' computers are set up to calculate cryptographic hash functions.

Source: spectrum.ieee.org

Case 00: Bitcoin



Hash value* + Nonce ▶ New hash value + Nonce ▶ New hash value + Nonce ▶ New hash value

* Each new hash value contains information about all previous Bitcoin transactions.

The mining computers calculate new hash values based on a combination of the previous hash value, the new transaction block, and a nonce.

Cryptographic Hashes

Cryptographic hash functions transform a collection of data into an alphanumeric string with a fixed length, called a hash value. Even tiny changes in the original data drastically change the resulting hash value. And it's essentially impossible to predict which initial data set will create a specific hash value.

The root of all evil ▶ 6d0a 1899 086a... (56 more characters)

The root of all euil ▶ 486c 6be4 6dde...

The root of all veil ▶ b8db 7ee9 8392...

Nonces

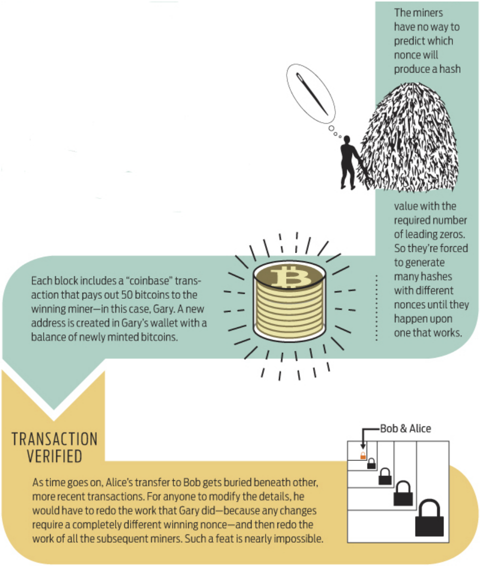
To create different hash values from the same data, Bitcoin uses "nonces." A nonce is just a random number that's added to data prior to hashing. Changing the nonce results in a wildly different hash value.

The root of all evil ??? ▶ 0000 0000 0000...

Creating hashes is computationally trivial, but the Bitcoin system requires that the new hash value have a particular form—specifically, it must start with a certain number of zeros.

Source: spectrum.ieee.org

Case 00: Bitcoin



The miners have no way to predict which nonce will produce a hash value with the required number of leading zeros. So they're forced to generate many hashes with different nonces until they happen upon one that works.

Each block includes a "coinbase" transaction that pays out 50 bitcoins to the winning miner—in this case, Gary. A new address is created in Gary's wallet with a balance of newly minted bitcoins.

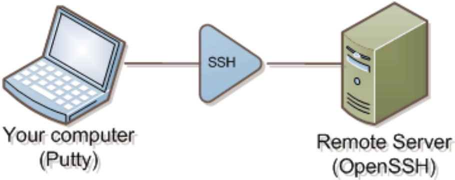
TRANSACTION VERIFIED

As time goes on, Alice's transfer to Bob gets buried beneath other, more recent transactions. For anyone to modify the details, he would have to redo the work that Gary did—because any changes require a completely different winning nonce—and then redo the work of all the subsequent miners. Such a feat is nearly impossible.

Source: spectrum.ieee.org

Case 01: Secure Shell (SSH) and Transport Layer Security (TLS)

1. SSH is a cryptographic network protocol for operating network services securely over an unsecure chanel.
2. The best known example application is for remote login to computer systems by users.



Source:Jason Young

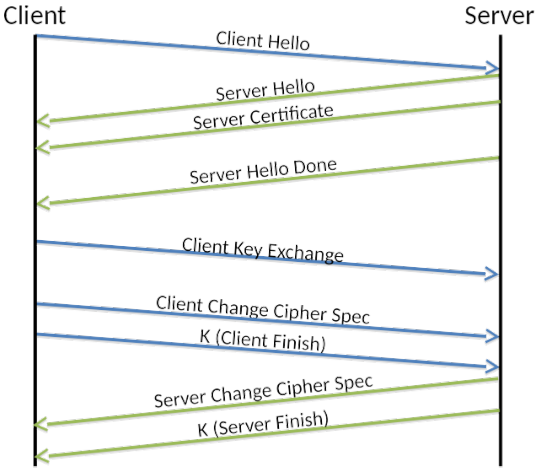
Case 01: Secure Shell (SSH) and Transport Layer Security (TLS)

1. ECC can be used in three times in the SSH protocol:
   - In RFC 5656 specifies the ephemeral Elliptic Curve Diffie-Hellman key exchange method.
   - The server authenticates itself by signing a transcript of the key exchange, this can be done with ECDSA.
   - Finally, clients can use ECDSA public keys for client authentication.

Case 01: Secure Shell (SSH) and Transport Layer Security (TLS)

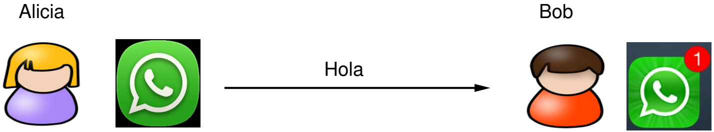
1. TLS is a cryptographic protocol that provide communications security over a computer network.
2. The Transport Layer Security protocol aims mainly to provide privacy and data integrity between two entities.
3. RFC 4492 specifies elliptic curve cipher suites for TLS.
4. Elliptic curves can arise in several locations in the protocol:
   - The elliptic curve Diffie Hellman (ECDH) key exchange.
   - TLS certificates can use either RSA or ECDSA.
   - Also cen be used for encryption.

Case 01:TLS



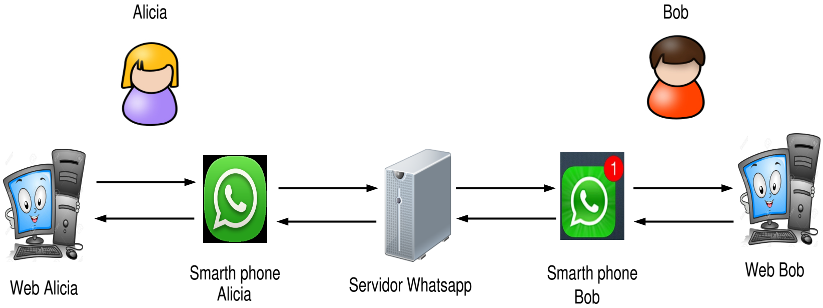
Source: MongoDB Asynchronous Java Driver

Case 02: WhatsApp



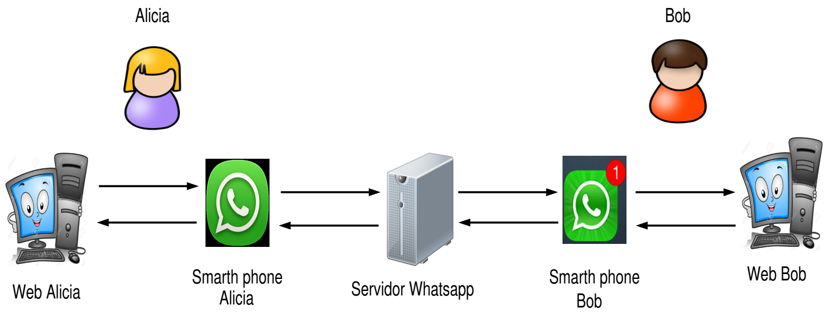
1. WhatsApp allows people to exchange messages and make calls around the world.
2. After March 31, 2016 are end-to-end encrypted.
3. The Signal Protocol, designed by Open Whisper Systems, is the basis for WhatsApps end-to-end encryption.
4. This protocol aims to prevent third parties and WhatsApp from having plaintext access to messages or calls.
5. Even if encryption keys are physically compromised, they cannot be used to go back in time to decrypt previously transmitted messages.

Case 02:Arquitectura de WhatsApp



1. WhatsApp Web is an extension of WhatsApp phone.
2. WhatsApp Web needs a connection to a phone in order to synchronize the messages.

Case 02: Arquitectura de WhatsApp



1. The WhatsApp account needs to be available and it also requires an Internet connection.
2. WhatsApp offers that messages are stored only on the respective phones.
3. However, messages, images, and videos are stored on their servers temporarily, until they are delivered or in a maximum month period.

Case 02: Open WhisperSystem explanation

The protocol used by WhatsApp is based on:

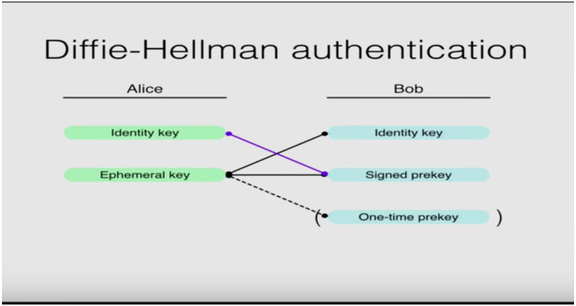
1. *Off the record messaging protocol* (OTR): was proposed by Borisov, Goldberg and Brewer.
2. *Silent circle instant message protocol* (SCIMP): proposed by Vinnie Moscaritolo, Gary Belvin, and Phil Zimmermann.

This protocol has advantages and pitfalls, Open Whisper System worked in a protocol that has the better of two worlds.

Case 02: WhatsApp Security Goals and Trust Model

1. Security goals
   - Privacy and Integrity.
   - *Forward security*.
2. Trust Model
   - Minimize the infrastructure, however, public key directories are required.

Case 02: WhatsApp Triple Diffie-Hellman



Diffie-Hellman authentication

Alice

Identity key

Ephemeral key

Bob

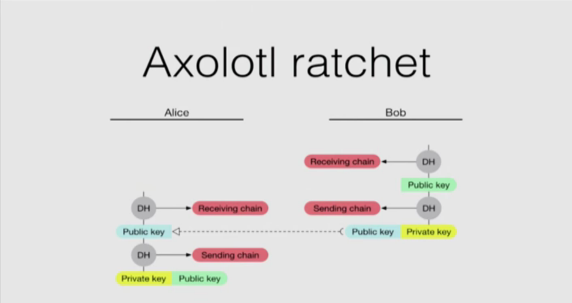
Identity key

Signed prekey

(One-time prekey)

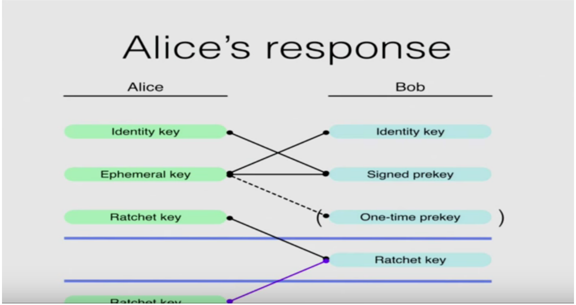
1. The aim is to build a shared secret for each session.
2. This is accomplish using Diffie Hellman over elliptic curves.

Case 02:Ratchet Axolotl



1. In this protocol the keys are updated based on the Diffie Hellmans performed.

Case 02: Complete Protocol



Alice’s response

1. Triple Diffie Hellman
2. Rachet