Math Money: A simple introduction to crypto-currencies

Rohas Nagpal
Asian School of Cyber Laws
rn@asianlaws.org

Abstract: Crypto-currencies (bitcoin et al) have caught the attention of Governments, enforcement agencies, geeks and the general public. This document provides a simple introduction to crypto-currencies and briefly introduces terms such as cryptography, hash functions, proof-of-work, digital signatures, mining, merkle root & tree, crypto-currency addresses and wallets. This document is intended for the novice reader and may suffer from errors inherent when a complex topic is (over?) simplified.

Note: Although this document mentions Bitcoin, most of it applies to any system that “uses public key cryptography, peer-to-peer networking and proof-of-work to process and verify payments”.

Note: This document is intended for the novice reader and may suffer from errors inherent when a complex topic is (over?) simplified.

1. Evolution of money: from cowry shells to the blockchain

Our ancestors started off with the barter system - something like "I will give you 2 buffaloes in return for 5 shiny new super-sharp axes". Soon they realised that the barter system had too many limitations - everyone didn’t want buffaloes, buffaloes were neither divisible (not too many people would want 0.35 buffaloes) nor very portable (imagine having to carry a buffalo on your shoulders while going shopping).

So they moved on to more acceptable, divisible, homogeneous and portable forms of money - cowry shells, salt, gold, silver and lots more. The Chinese invention of paper eventually led to the birth of paper currency, which was initially backed by gold or other precious metals. Then the world moved on to fiat money - currency that’s declared as legal tender by a government but not backed by a physical commodity¹.

¹ Have a look at a 100-rupee note. It caries a promise signed by the Governor of the Reserve Bank of India (RBI) – “I promise to pay the bearer the sum of one hundred rupees”. If you were to take this note to the Governor of the RBI, he would give you coins or one-rupee notes totaling 100 rupees. The RBI gets the power to issue currency notes by section 31 of the Reserve Bank of India Act, 1934. This section states that “No person in India other than the Bank or, as expressly authorized by this Act, the Central Government shall draw, accept, make or issue any bill of exchange, hundi, promissory note or engagement for the payment of money payable to bearer on demand, or borrow, owe or take up any sum or sums of money on the bills, hundis or notes payable to bearer on demand of any such person...”
This brings us to an essential question – what is money? Money’s a matter of functions four, a Medium, a Measure, a Standard, a Store. So goes the couplet based on William Stanley Jevons analysis of money in 1875. This meant that for something to be called as money, it must function as a medium of exchange, a measure of value, a standard of deferred payment and a store of value.

The birth of computers and the Internet brought in many electronic payment systems including debit cards, stored value cards, giro transfers, credit cards, net-banking, electronic bill payments, electronic cheques, mobile wallets, digital gold currencies, digital wallets, electronic funds transfer at point of sale, mobile banking, SMS banking, online banking, payment cards, real-time gross settlement systems, SWIFT, wire transfers and more.

And then came Satoshi Nakamoto’s path breaking whitepaper - Bitcoin: A Peer-to-Peer Electronic Cash System in October 2008. This brought the world its first truly peer-to-peer electronic currency. Bitcoin earned a lot of notoriety primarily because of its use by members of the now shut-down Silk Road - an illegal online marketplace that facilitated the sale of hundreds of millions of dollars worth of drugs, guns, stolen financial information, counterfeit documents and more. All Silk Road transactions were conducted exclusively in bitcoin.

A lot of crypto-currencies piggybacked on Bitcoin’s underlying innovation – the blockchain. In fact we now have more than 650 virtual currencies being used.

---

2 Virtual currency is a digital representation of value that can be digitally traded and functions as (1) a medium of exchange; and/or (2) a unit of account; and/or (3) a store of value, but does not have legal tender status (i.e., when tendered to a creditor, is a valid and legal offer of payment) in any jurisdiction. It is not issued nor guaranteed by any jurisdiction, and fulfills the above functions only by agreement within the community of users of the virtual currency. Virtual currency is distinguished from fiat currency (a.k.a. “real currency,” “real money,” or “national currency”), which is the coin and paper money of a country that is designated as its legal tender; circulates; and is customarily used and accepted as a medium of exchange in the issuing country. It is distinct from e-money, which is a digital representation of fiat currency used to electronically transfer value denominated in fiat currency. E-money is a digital transfer mechanism for fiat currency—i.e., it electronically transfers value that has legal tender status. [Source: FATF report on Virtual Currencies - Key Definitions and Potential AML/CFT Risks]

3 Cryptocurrency refers to a math-based, decentralised convertible virtual currency that is protected by cryptography. - i.e., it incorporates principles of cryptography to implement a distributed, decentralised, secure information economy. Cryptocurrency relies on public and private keys to transfer value from one person (individual or entity) to another, and must be cryptographically signed each time it is transferred. The safety, integrity and balance of cryptocurrency ledgers is ensured by a network of mutually distrustful parties (in Bitcoin, referred to as miners) who protect the network in exchange for the opportunity to obtain a randomly distributed fee (in Bitcoin, a small number of newly created bitcoins, called the “block reward” and in some cases, also transaction fees paid by users as an incentive for miners to include their transactions in the next block). Hundreds of cryptocurrency specifications have been defined, mostly
around the world. And now we have become a world where bankers wake up each morning wondering – "has the meaning of money and banking changed while I slept".

This rapid change in the global money ecosystem has implications for all of us - from Governments looking to clamp down on money laundering, tax evasion and terrorist funding to banks looking to understand the implications of the blockchain technology. From law enforcement looking to clamp down on the Mafia using Bitcoin to businesses looking for faster and cheaper ways to receive and transfer money globally.

2. The mathematics of it all

Sanya’s a naughty young girl who’s been grounded for a week. She wants to sneak out for desert with her friends but obviously can’t let her dad know about it. She’s not allowed to use her cellphone, so the only way for her to call her friends is using the good old landline in her dad’s room.

Since she regularly gets grounded, she and her friends have worked out a simple system for sharing secrets. When she says, “have you read the book I told you about” she actually means “let’s sneak out tonight”. When she says something about “page 10” of the book, she means “pick me up at 10 pm”. Continuing the logic, page 11 would mean 11 pm and so on.

So on the phone she asks her friend “Have you read the book I told you about? Page 12 is really funny”, she means, “Let’s sneak out tonight, pick me up at midnight”.

What we have just seen is cryptography (and a rebellious teenager) in action in the real world.

The sentence “Let’s sneak out tonight, pick me up at midnight” is plain text – what Sanya actually wants to convey. The sentence “Have you read the book I told you about? Page 12 is really funny” is the cipher text – something that an adversary (her dad in this case) should not be able to understand.

Encryption is the process of converting plain text to cipher text. The reverse process is decryption.

derived from Bitcoin, which uses a proof-of-work system to validate transactions and maintain the block chain. While Bitcoin provided the first fully implemented cryptocurrency protocol, there is growing interest in developing alternative, potentially more efficient proof methods, such as systems based on proof-of-stake. [Source: FATF report on Virtual Currencies - Key Definitions and Potential AML/CFT Risks]

This science of encrypting and decrypting messages (cryptography) has been used for thousands of years. It is believed that when Julius Caesar sent messages to his generals, he replaced every A in his messages with a D, every B with an E, and so on through the alphabet. Only someone who knew the “shift by 3” rule could decipher his messages.

For example, if we want to encode the word “SECRET” using Caesar’s key value of 3, we offset the alphabet so that the 3rd letter down, (D), begins the alphabet.

So starting with ABCDEFGHIJKLMNOPQRSTUVWXYZ
and sliding everything up by 3, you get
DEFGHJKLMNOPQRSTUVWXYZABC
where D=A, E=B, F=C, and so on.

Using this scheme, the plaintext, “SECRET” encrypts as “VHFUHW”. To allow someone else to read the cipher text, you tell him or her that the key is 3. This method is called symmetric cryptography and involves using the same key for encrypting as well as decrypting a message. This naturally poses a serious problem – what if an adversary gets hold of this key? At some point of time the sender and receiver need to exchange the key. That’s when an adversary could get hold of the key. In modern cryptography, keys are really really large numbers.

The secure-key-exchange problem was solved with the birth of asymmetric key cryptography – in which two different but related keys are used - the public key to encrypt data and the corresponding private key to decrypt the data. If Sanya were to send an encrypted message to Karan, she would encrypt the message using his public key (which is available to the world). Once encrypted, the message can only be decrypted using Karan’s private key (which would only be available to Karan).

To understand how this works, let’s look at the RSA algorithm (named after its inventors Ron Rivest, Adi Shamir, and Leonard Adleman).

The RSA public-key encryption algorithm works in the following manner:

1. Generation of a public-private key pair.
2. Encryption of a message (plain text) with the public key generated in step (1) to get the cipher-text.
3. Decryption of the cipher-text by using the corresponding private key generated in step (1).

Step 1: Generation of a key pair

1. Select two large integer primes p and q.
2. Multiply p and q to get a number n, that means, \( pq = n \).
3. Obtain \( \varphi \) which is the product of \( (p-1) \) and \( (q-1) \), that means \( \varphi = (p-1)(q-1) \).
4. Select $e$ such that $1 < e < \varphi$ and the greatest common divisor of $e$ and $\varphi$ is 1. That means $e$ and $\varphi$ are coprime.

5. Compute $d$ such that $1 < d < \varphi$ and $ed \equiv 1 \mod \varphi$. This means that the value of $d$ must be such that $ed - 1$ should be completely divisible by $\varphi$ or $(ed - 1)/\varphi$ should be an integer.

6. The public key is $(e, n)$ and the corresponding private key is $(d, n)$.

Step 2: Encryption process
Suppose the message to be encrypted is $m$. The cipher-text $c$ is obtained by raising the message to the value of $e$ and finding out its modulo $n$.

That means

$$c = m^e \mod n.$$  

Step 3: Decryption process
Decryption is achieved by raising the cipher-text $c$ obtained in step 2 to the value of $d$ and finding out its modulo $n$.

That means $m = c^d \mod n$.

Let’s try the algorithm with really small prime numbers$^5$: 3 and 11. (In reality the primes chosen would be really really large).

1. Choose $p = 3$ and $q = 11$

2. Compute $n = p \times q = 3 \times 11 = 33$

3. Compute $\varphi = (p - 1) \times (q - 1) = 2 \times 10 = 20$

4. Choose $e$ such that $1 < e < \varphi$ and $e$ and $\varphi$ are coprime. Let $e = 7$

5. Compute a value for $d$ such that $1 < d < \varphi$ and $ed \equiv 1 \mod \varphi$.
One solution is $d = 3$.

6. Public key is $(e, n) \Rightarrow (7, 33)$
Private key is $(d, n) \Rightarrow (3, 33)$

7. Suppose the plain text is 2.
The cipher text will be $c = m^e \mod n$.
That's $2^7 \mod 33 = 128 \mod 33 = 29$

8. The decryption will be

$$c^d \mod n$$

$^5$Source: https://www.cs.utexas.edu/~mitra/honors/soln.html
\[ \begin{align*} 29^3 \mod 33 &= 24389 \mod 33 \\ &= 2 \end{align*} \]

The security of the RSA cryptosystem is based on the integer factorization problem. Any adversary who wishes to decipher the cipher-text \( c \) must do so by using the publicly available information \((n, e)\). One possible method is to first factor \( n \), and then compute \( \varphi \) and \( d \) just as was done in the above mentioned steps. The factoring of \( n \) is currently computationally infeasible (provided sufficiently large prime numbers are chosen as \( p \) and \( q \)) and therein lays the strength of the RSA cryptosystem.

Before we get into the nuts and bolts of how crypto-currencies work, we need to understand some more concepts including hash functions. A one-way hash function takes an input (e.g. a PDF file, a video, an email, a string etc.) and produces a fixed-length output e.g. 160-bits.

The hash function ensures that if the information is changed in any way – even by just one bit – an entirely different output value is produced. The table below shows some sample output values using the sha1 (40) hash function\(^6\).

<table>
<thead>
<tr>
<th>Input</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>sanya</td>
<td>c75491c89395de9fa4ed29affda0e4d29cbd290</td>
</tr>
<tr>
<td>SANYA</td>
<td>33fef490220a0e6dee2f16c5a8f78ce491741adc</td>
</tr>
<tr>
<td>Sanya</td>
<td>4c391643f247937bee14c0bca9ff985fc0d0ba</td>
</tr>
</tbody>
</table>

It can be seen from the table above that by changing the input from sanya to SANYA, an entirely different hash value is generated. What must be kept in mind is that irrespective of the size of the input, the hash output will always be of the same size.

Two things must be borne in mind with regard to one-way hash functions:

1. It is computationally infeasible to find two different input messages that will yield the same hash output.
2. It is computationally infeasible to reconstruct the original message from its hash output.

Having understood hash functions, let’s have a look at another interesting concept called proof-of-work. This is a way to reduce spam and denial of service attacks by requiring a computer to spend some time and processing power to solve something.

One such proof-of-work system that is used in crypto-currencies is hashcash. The basic premise of hashcash is that if the sender of an email can prove that she has

\(^6\) Computing hash of an electronic record is a very simple process. E.g. in php it can be done with: `hash_file('sha256', $filename).`
spent reasonable time and computational power to solve some puzzle, it can be believed that the sender is not a spammer. The logic is that spamming would be economically infeasible if a spammer had to spend non-trivial time and computational power for every single email being sent.

Let’s develop an elementary proof-of-work system, based on hashcash, which can be used to control spam. Let’s presume that rn@asianlaws.org is sending an email to info@lexcode.com. The sender must include something similar to the following in the header of the email:

```
rn@asianlaws.org: info@lexcode.com:18032016:xxxx
```

That’s 4 pieces of information separated by colons. The first piece is the sender's email address, the second is the receiver's email address and the third is the current date in DDMMYYYY format. The fourth piece is something that needs to be calculated by the sender's computer. Let’s call it a **nonce**.

The objective is to find an input that would result in a sha256 hash which begins with 5 zeros.

So we start the nonce at a value of 0 and then keep incrementing it (1, 2, 3 ...) and calculating the hash. Something like this:

<table>
<thead>
<tr>
<th>Input</th>
<th>sha256 hash</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:rn@asianlaws.org">rn@asianlaws.org</a>:<a href="mailto:info@lexcode.com">info@lexcode.com</a>:18032016:1</td>
<td>288721860bec3a490811981c831702d4f41e54c3f8c183c5650ac73f231659c</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input</th>
<th>sha256 hash</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:rn@asianlaws.org">rn@asianlaws.org</a>:<a href="mailto:info@lexcode.com">info@lexcode.com</a>:18032016:2</td>
<td>11caf434535c35cdc843e801382f0a8643a03500649a9bfa41c8e6a4be65a413</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input</th>
<th>sha256 hash</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:rn@asianlaws.org">rn@asianlaws.org</a>:<a href="mailto:info@lexcode.com">info@lexcode.com</a>:18032016:3</td>
<td>aad80b9c58e977a5da9081b2667af443b50425876920528f237df0a6fe1aa4</td>
</tr>
</tbody>
</table>

And so on till .. 1580661

<table>
<thead>
<tr>
<th>Input</th>
<th>sha256 hash</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:rn@asianlaws.org">rn@asianlaws.org</a>:<a href="mailto:info@lexcode.com">info@lexcode.com</a>:18032016:1580661</td>
<td>0000080602f705257e74a4e847e9ed23ab61be5b2ba4263fbc90bd7c7c7ab4</td>
</tr>
</tbody>
</table>

Calculating this may not take a genuine sender a lot of time and computational power but if a spammer were to make these calculations for millions of emails, it will take a non-trivial amount of time and computational power.

At the receiver's end, the computer will simply take the following line from the header of the email and calculate the hash.

```
rn@asianlaws.org:info@lexcode.com:18032016:1580661
```

If the hash begins with a pre-defined number of zeros (5 in this example), the email would not be considered spam. This will take the receiver a trivial amount of time and computational power since it just has to calculate the hash of one
input. The date can be used as an additional validation parameter – e.g. if the date is within 24 hours of the time of receipt, the email will be approved for download.

A very important application of public key cryptography is a **digital signature**. In this, the signer first calculates the hash of the message she wants to digitally sign. Then using her private key and the hash, she creates a digital signature, using the relevant algorithm.

This digital signature is unique to the message.

The signer then sends the message and the digital signature to the receiver. The receiver re-computes the hash from the message. The receiver also computes another string using the digital signature and the signer's public key (using the relevant algorithm). If this string and the hash match, the digital signature is verified.

**Blind digital signatures** were subsequently developed for use in digital cash and cryptographic voting systems. In this system, the content of the message is disguised before it is signed. The resulting blind signature can be verified against the original, un-blinded message in the manner of a regular digital signature.

However, blind digital signatures do not solve the **double-spending** problem. In case of physical currency notes, you cannot double-spend a note because once you hand the note over to someone, you don't have the note anymore to spend again. Since electronic records are easily duplicated, a “digital coin” can be spent multiple times.

Bitcoin solves the double-spending problem through the **blockchain** - a public ledger containing an ordered and time-stamped record of transactions. In addition to preventing double-spending, the blockchain prevents the modification of previous transaction records.

A block of one or more new transactions is collected into the transaction data part of a block. Copies of each transaction are hashed, and the hashes are then

---

7 Source: [https://en.wikipedia.org/wiki/Blind_signature](https://en.wikipedia.org/wiki/Blind_signature). An often-used analogy to the cryptographic blind signature is the physical act of a voter enclosing a completed anonymous ballot in a special carbon paper lined envelope that has the voter's credentials pre-printed on the outside. The ballot can be marked through the envelope by the carbon paper. The voter hands the sealed envelope to an official who verifies the credentials and signs it. Once signed, the package is given back to the voter, who transfers the now signed ballot to a new unmarked normal envelope. Thus, the signer does not view the message content, but a third party can later verify the signature and know that the signature is valid within the limitations of the underlying signature scheme.
paired, hashed, paired again, and hashed again until a single hash remains, the **merkle root** of a **merkle tree**.  

Let’s consider a simple illustration of how the blockchain works. Consider a block that has 6 transactions a, b, c, d, e and f.

The **merkle tree** is:

\[
\begin{align*}
d1 &= \text{double-hash (a)} \\
d2 &= \text{double-hash (b)} \\
d3 &= \text{double-hash (c)} \\
d4 &= \text{double-hash (d)} \\
d5 &= \text{double-hash (e)} \\
d6 &= \text{double-hash (f)} \\
d7 &= \text{double-hash (d1 concatenated with d2)} \\
d8 &= \text{double-hash (d3 concatenated with d4)} \\
d9 &= \text{double-hash (d5 concatenated with d6)} \\
d10 &= \text{double-hash (d7 concatenated with d8)} \\
d11 &= \text{double-hash (d9 concatenated with d9)}
\end{align*}
\]

Since there are an odd number of hashes, we take d9 twice.

\[
d12 = \text{double-hash (d10 concatenated with d11)}
\]

**d12** is the **merkle root** of the 6 transactions in this block. This is stored in the block header. Additionally, each block also stores the hash of the header of the previous block. This chains the blocks together and ensures that a transaction cannot be modified without modifying the block that records it and all following blocks. Transactions are also chained together.

Bitcoin uses a *proof-of-work* technique similar (but more complex) than the one discussed earlier in this document. Since good cryptographic hash algorithms convert arbitrary inputs into “seemingly-random” hashes, it is not feasible to modify the input to make the hash predictable. To prove that she did some extra work to create a block, a **miner** must create a hash of the block header, which does not exceed a certain value.

The term **miner** must not be compared with a gold or coal miner in the real world. While a gold miner digs into the earth to discover gold, a bitcoin miner uses computational power to calculate hashes. To add an entire block to the

---

block chain, a Bitcoin *miner* must successfully hash a block header to a value below the target threshold. Bitcoin miners spend a lot of money (for computational power and electricity) and are compensated by way of a reward for each block they mine – this was initially 50 bitcoins per block and is halving every 210,000 blocks. Miners also earn transaction fees. Miners usually operate as part of a large pool instead of as individuals.

Interestingly, Bitcoins can be also be mined with a pencil and paper⁹.

The first-ever Bitcoin block is known as the **genesis block**. Each subsequent block is addressed by its **block height**, which represents the number of blocks between it and the genesis block.

New blocks are added to the block chain if their hash is at least as challenging as a **difficulty** value expected by the Bitcoin **consensus protocol**. According to the bitcoin protocol, it should take 2 weeks for 2016 blocks to be generated. If the time taken is more or less than 2 weeks then the difficulty value is relatively decreased or increased.

The overview of the Bitcoin process is explained very well on Ken Shirriff’s blog¹⁰.

---


¹⁰ To simplify slightly, bitcoins consist of entries in a distributed database that keeps track of the ownership of bitcoins. Unlike a bank, bitcoins are not tied to users or accounts. Instead bitcoins are owned by a Bitcoin address, for example 1KKKK6N21XKo48zW67K82Q3BnQK63z. A transaction is the mechanism for spending bitcoins. In a transaction, the owner of some bitcoins transfers ownership to a new address. A key innovation of Bitcoin is how transactions are recorded in the distributed database through mining. Transactions are grouped into blocks and about every 10 minutes a new block of transactions is sent out, becoming part of the transaction log known as the blockchain, which indicates the transaction has been made (more-or-less) official. Bitcoin mining is the process that puts transactions into a block, to make sure everyone has a consistent view of the transaction log. To mine a block, miners must find an extremely rare solution to an (otherwise-pointless) cryptographic problem. Finding this solution generates a mined block, which becomes part of the official block chain.

Miners pick up your transaction, generate a
A Bitcoin address is an identifier of 26 to 35 alphanumeric characters, beginning with the number 1 or 3, which represents a possible destination for a bitcoin payment. Addresses can be generated at no cost by any user of Bitcoin.

mined block containing your transaction, and send this mined block to peers. Eventually your client will receive the block and your client shows that the transaction was processed.

Bitcoin uses digital signatures to ensure that only the owner of bitcoins can spend them. The owner of a Bitcoin address has the private key associated with the address. To spend bitcoins, they sign the transaction with this private key, which proves they are the owner. (It's somewhat like signing a physical check to make it valid.) A public key is associated with each Bitcoin address, and anyone can use it to verify the digital signature.

Blocks and transactions are identified by a 256-bit cryptographic hash of their contents. This hash value is used in multiple places in the Bitcoin protocol. In addition, finding a special hash is the difficult task in mining a block.

Source: https://en.bitcoin.it/wiki/Address. The technical process to create Bitcoin addresses is explained as under:

• Step 0: Having a private Elliptic Curve Digital Signature Algorithm key: 18E1A7B6A307F426A94F8114701E7CBEE774E7F9A47E2C2035DB29A206321725
• Step 1: Take the corresponding public key generated with it: 0450B63AD64A87AE822FE83C1AF1A8403CB53F53E48608511DAD8A048B87E5B2352
  2CD470243453A299FA9E772377716103ABC11A1DF3BB55ED6F2EE187E9C582BA6
• Step 2: Perform SHA-256 hashing on the public key 600FFE422B4E00731A59557A5CCA46CC183944191006324A447BDB2D98D4B40B
• Step 3: Perform RIPEMD-160 hashing on the result of SHA-256 010966776006953D5567439E5E39F86A0D273BEE
• Step 4: Add version byte in front of RIPEMD-160 hash (0x00 for Main Network) 00010966776006953D5567439E5E39F86A0D273BEE

(note that below steps are the Base58Check encoding11)
• Step 5: Perform SHA-256 hash on the extended RIPEMD-160 result 445C7A8007A93D873318828BB320A8FE2DEBD2AE1B47F0F50BC10BAE845C094
• Step 6: Perform SHA-256 hash on the result of the previous SHA-256 hash D61967F63C7DD183914A4AE529F6AD54D62CE3D277798075B107615C1ABA30
• Step 7: Take the first 4 bytes of the second SHA-256 hash. This is the address checksum D61967F6
• Step 8: Add the 4 checksum bytes from the previous stage at the end of extended RIPEMD-160 hash from stage 4. This is the 25-byte binary Bitcoin Address. 00010966776006953D5567439E5E39F86A0D273BEE61967F6
• Step 9: Convert the result from a byte string into a base58 string using Base58Check encoding. This is the most commonly used Bitcoin Address format 16UwLL9Risc3QfpqBUvKofHmBQ7wMtjvM

11
There are currently two address formats in common use:

<table>
<thead>
<tr>
<th>Common P2PKH which begin with the number 1</th>
<th>Example: 1BvBMSEYstWetqTFn5Au4m4GFg7xJaNVN2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newer P2SH type starting with the number 3</td>
<td>Example: 3J98t1WpEZ73CNmQviecrnyiWrnqRhWNLy</td>
</tr>
</tbody>
</table>

Bitcoin wallets at their core are a collection of private keys.

These collections are stored digitally in a file, or can even be physically stored on pieces of paper.

The simplest Bitcoin wallet is a program, which performs these functions¹²:

- it generates private keys,
- derives the corresponding public keys,
- helps distribute those public keys as necessary,
- monitors for outputs spent to those public keys,
- creates and signs transactions spending those outputs, and
- broadcasts the signed transactions.

Although it's called a wallet, a Bitcoin wallet does not store bitcoins. The wallet is a collection of public-private key-pairs.

As discussed, the blockchain is a database of transaction information. It is constantly growing and is sent out to all nodes in the Bitcoin network. Every transaction is distributed to the network and all valid transactions are included in the next block, which is mined.

Imagine a real-world transaction where your salary is transferred to your bank account through an online transfer made by your employer. You then use your debit card to pay for dinner. This transfers some of the money to the restaurant’s account. In these 2 transactions, did you see a single currency note? No. So we can say that in today’s world most money exists as a history of transactions and balances.

Bitcoin, or for that matter most virtual currencies work the same way. They don’t actually “exist” in the true sense of the word. They just are there!

A bitcoin can be divided down to 8 decimal places - 0.00000001 is the smallest amount, also referred to as a satoshi. The last block that will generate bitcoins will be block 6,929,999. This is expected to be generated around the year 2140. After that, the total number of bitcoins will remain static at just below 21 million.

3. Additional reading and other resources

1. Blind Digital Signatures

2. Bitcoin: A Peer-to-Peer Electronic Cash System
   https://bitcoin.org/bitcoin.pdf

3. Bitcoin wiki
   https://en.bitcoin.it/wiki/

4. FATF Report on Virtual Currencies Key Definitions and Potential AML/CFT Risks

4. License

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. You are free to:
Share — copy and redistribute the material in any medium or format
Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:
Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Notices:
You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation.
No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.