Basic Cryptographic Primitives in Blockchain

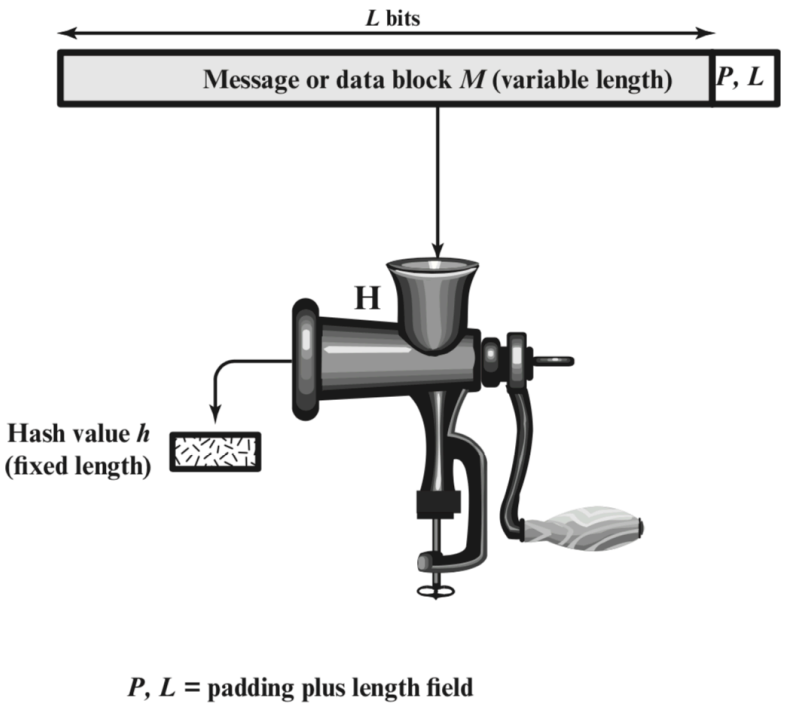
Outline

01 Part One

Hash Function

Overview of hash function

Definition of Hash Function



- A cryptographic hash function is a public function, usually denoted as H, it accepts a variable-length block of data M as input and produces a fixed-size hash value H(M).
- The function value H(M) is often referred to as a hash value or message digest, and so on.
- For large sets of inputs, the resulting output is evenly distributed and seemingly random.

A Simple Example

- For any input positive number **n**, output 4-digit number **h**

- **Example** n = 10
  - Step 1. m = n * n + 37 137 = 10 * 10 + 37
  - Step 2. p = m * m 18769 = 137 * 137
  - Step 3. h = p % 10000 8769 = 18769 % 10000

- Given the output **h**, it is hard to calculate input **n**

This example is not secure and random, but used for understanding

Application in the Internet

How to verify a large file has not been tampered with?

1. Verify the backup of the file. The complexity of space is high.

2. Verify the hash value of the file. The complexity of space is low.

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		d348d978a5387512fbc7d7d52dd3a5ef	23161893	SIG
XZ compressed source tarball	Source release		172c650156f7bea68ce31b2fd01fa766	17268888	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	47b06433e242c8eb848e035965a860ac	29163525	SIG
Windows help file	Windows		89bb2ea8c5838bd2612de600bd301d32	8183265	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	6aa3b1c327561bda256f2deebf038dc9	7444654	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	e0c910087459df78d827eb1554489663	26797616	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	d1d09dad50247738d8ac2479e4cde4af	1348896	SIG
Windows x86 embeddable zip file	Windows		29672b400490ea21995c6dbae4c4e1c8	6614968	SIG
Windows x86 executable installer	Windows		e9db9cf43b4f2472d75a055380871045	25747128	SIG
Windows x86 web-based installer	Windows		8b326250252f15e199879701f5e53c76	1319912	SIG

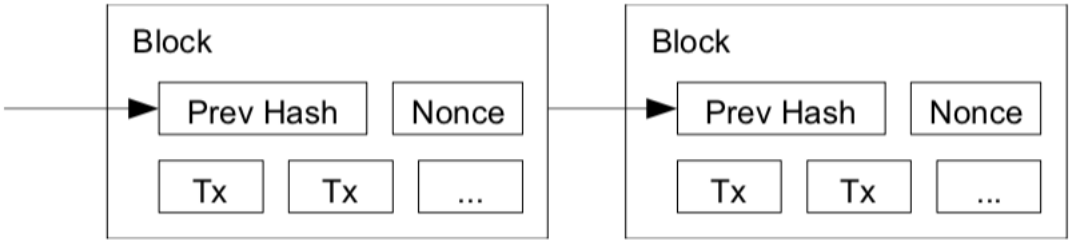
Application in the Blockchain

How to verify all transaction history of the bitcoin not been tampered with?

The history of bitcoin is constantly increasing, so the hash value of the history changes after each increment. It is too complicated to calculate the hash value after each increment.

How to solve this?

The history of the bitcoin is divided into several blocks, and each block contains the hash value of the previous one. After each increment, the new transactions are stored in a new block. We can only store the latest hash value to verify the all history not been tampered with.

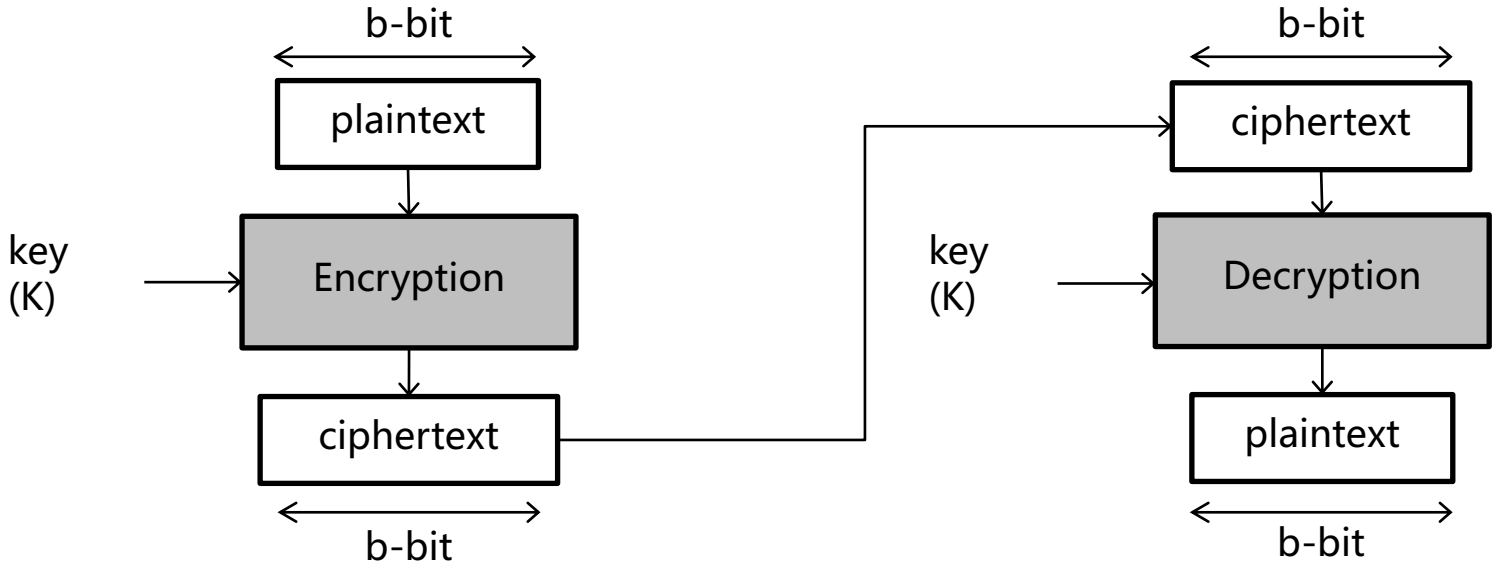


02 Part Two

Asymmetric Cryptosystems

Review Symmetric Cryptography

In symmetric cryptography, the encryption key is the same as the corresponding decryption key (or the corresponding decryption key can be easily deduced from the encryption key).



The sender and receiver must transmit the key through a secure channel prior to transmission of the ciphertext.

Compare with Symmetric Cryptography

Problems	Symmetric Cryptography	Asymmetric Cryptography
Key distribution	Need for prior key distribution	No need for key distribution
Key management	Each user needs to keep n keys with n friends; The number of key pairs to manage in the network is 1+2+3+…+(n-1)=n*(n-1)/2. When a user joins, each user in the system needs to share a new key with the new user.	Each user only needs to keep 1 key-pair with any number of friends; The number of key pairs to manage in the network is *n*. When a user joins, the system needs to produce one pair of public-private key.
Signature	Digital signature capability can not be provided.	Digital signature capability is provided.

Cryptography has freed itself from the need to secure the transmission of keys and its application prospects are suddenly clear.

Principles of Asymmetric Cryptosystems

- Asymmetric cryptosystem is an important milestone in cryptography.
- Asymmetric cryptography is also known as public key cryptography or double key cryptography.
- In 1976, Diffie and Hellman first proposed the idea of asymmetric cryptography in their article “New directions in cryptography”, which played an extremely important role in promoting the development of cryptography.

-W.Diffie and M.E.Hellman, New Directrions in Cryptography, IEEE Transaction on Information Theory, V.IT-22.No.6, Nov 1976, PP.644-654

Basic Features of Asymmetric Cryptosystem

- The key pair includes a public key *pk* and a secret key *sk*, the *sk* is kept by the user secretly and the *pk* is public to all.
- The message encrypted by *pk* can be decrypted by *sk*, also, the message encrypted by *sk* can be decrypted by *pk*

  Which can be shown as： $E_{pk}\left(E_{sk}(M)\right) = E_{sk}\left(E_{pk}(M)\right) = M$
- When we want to send a private message to a user, we use the public key of the user to encrypt the message. So only the user can decrypt the ciphertext with his/her secret key. $M \rightarrow E_{pk}(M) \rightarrow M$
- When we want to verify the source of a message, the sender uses his/her secret key to encrypt the message. So all of us can verify the ciphertext, also called signature, with his/her public key. $M \rightarrow E_{sk}(M) \rightarrow M$

03 Part Three

Digital Signature

Introduction to Digital Signature

Digital signature, also called electronic signature

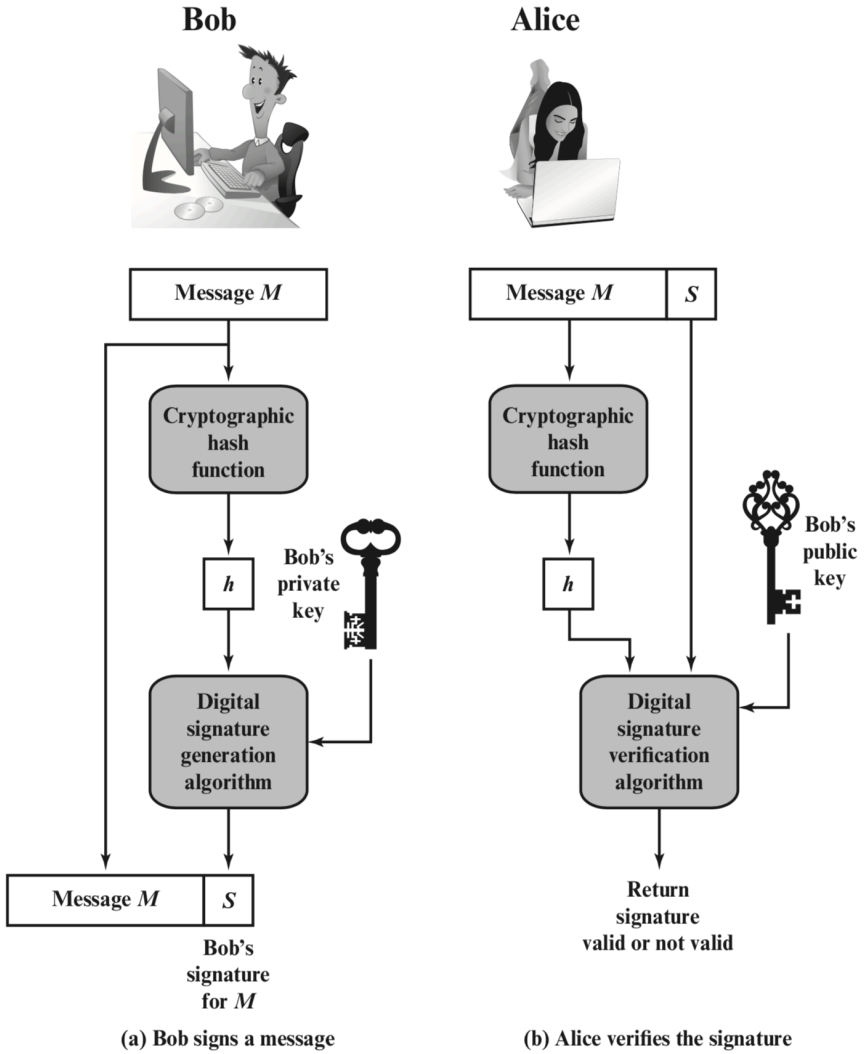
- Sign the electronic documents using electronic means;
- Electronic signature must possess two properties of handwritten signature, which are verifiability and unforgeability.

The digital signature has the following functions:

- Any one can verify whether the message is sent by the signer;
- The signer cannot deny that the signature was sent by himself. Also means nobody else can sign the same signature;
- It can confirm that the message had not been altered from the time it was signed to the time it was received.

Implementation

- In brief, digital sign is a process of encryption, and the verification of digital signature is a process of decryption.
- Public key cryptosystem can provide digital signature :
- Sign the document with private key d,
- Verify the signature with public key e.



Application in the Internet

How to verify the hash on the webpage has not been tampered with?

Using the signature of the corresponding public key generated by the website.

How to verify the public key has not been tampered with?

Check the certificate of the website which is signed by the root server.

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		d348d978a5387512fbc7d7d52dd3a5ef	23161893	SIG
XZ compressed source tarball	Source release		172c650156f7bea68ce31b2fd01fa766	17268888	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	47b06433e242c8eb848e035965a860ac	29163525	SIG
Windows help file	Windows		89bb2ea8c5838bd2612de600bd301d32	8183265	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	6aa3b1c327561bda256f2deebf038dc9	7444654	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	e0c910087459df78d827eb1554489663	26797616	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	d1d09dad50247738d8ac2479e4cde4af	1348896	SIG
Windows x86 embeddable zip file	Windows		29672b400490ea21995c6dbae4c4e1c8	6614968	SIG
Windows x86 executable installer	Windows		e9db9cf43b4f2472d75a055380871045	25747128	SIG
Windows x86 web-based installer	Windows		8b326250252f15e199879701f5e53c76	1319912	SIG

04 Part Four

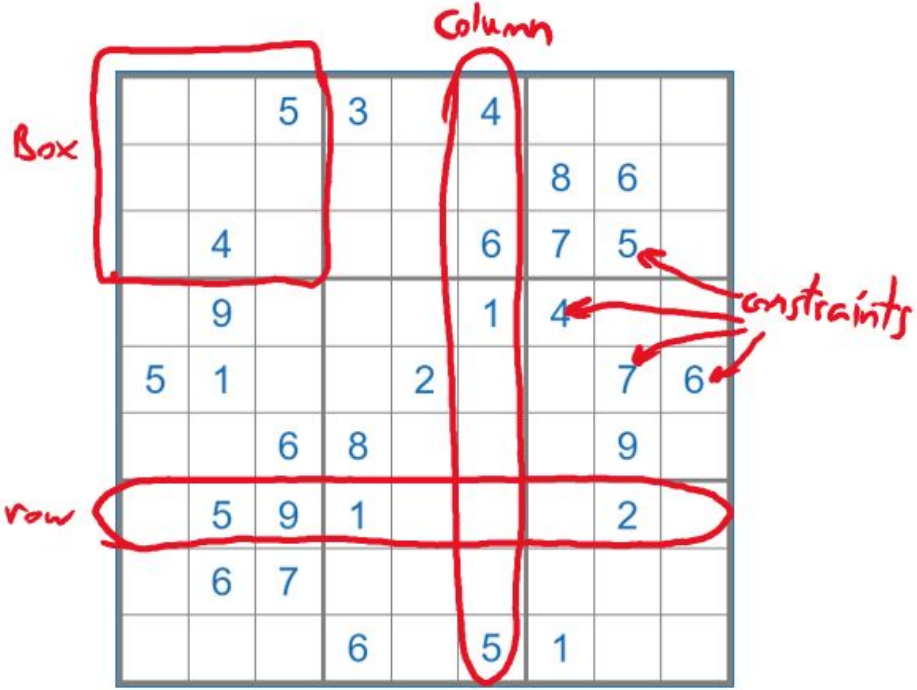
Zero-Knowledge Proof

Introduction to Zero-Knowledge Proof

Sudoku Game

- This is a Sudoku question. Let's assuming it's too hard for you to solve it. So you want me to prove there is an answer for this question.

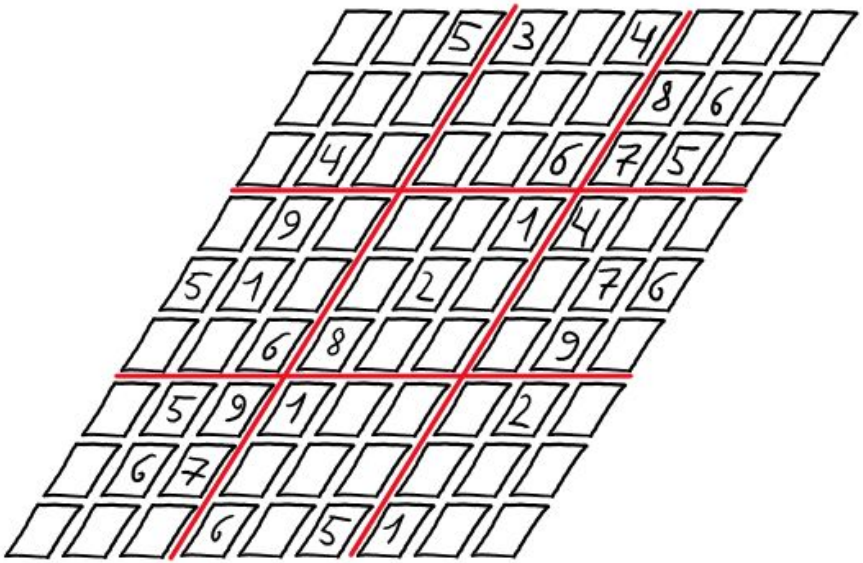
- However, you don't want me to tell you the answer directly, because you want to continue it.



Introduction to Zero-Knowledge Proof

Commit

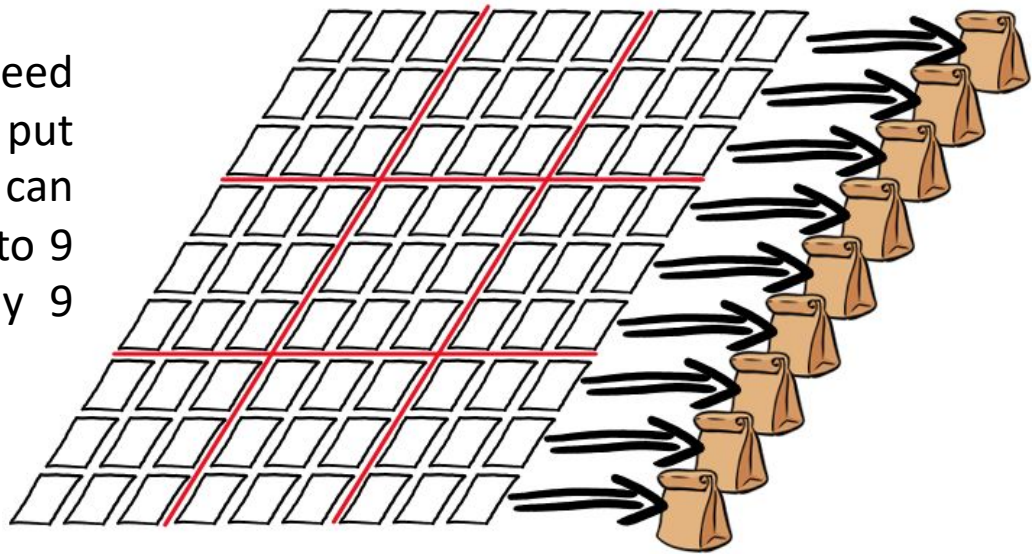
- First, I put 81 pieces of paper on the table, each with a number 1-9. The place of each piece is the same as the solution. And the Numbers in the question face up and the blanks face down.
- You can not turn up the pieces of paper to see the answer !



Introduction to Zero-Knowledge Proof

Challenge and Response

- Choose one way of row, column or box to verify the answer, which is called a challenge.
- Assuming that you choose row, so I need to response your challenge. I will put papers in each row into a bag, so we can see that the 81 pieces are divided into 9 bags and each one contains exactly 9 pieces. Which is called a response.



Introduction to Zero-Knowledge Proof

Verification

- Finally, you can open the 9 bags to verify whether there are exactly 9 pieces of paper written 1 – 9.
- If yes, it means that the commit I put on the table is probably right, at least each row of the commit is satisfied. Due to I don't know which kind of challenge you will propose, row, or column, or box.
- If you don't trust me, we can turn back into the commit phase, to start again. If I can pass several rounds of challenge, it means that I know a satisfied answer with high probability

Introduction to Zero-Knowledge Proof

The basic concept of zero-knowledge proof

- There are two parties in the zero-knowledge proof: Prover and Verifier
- The prover wants to convince the verifier that a statement is true without providing any other information to the verifier
- Three important properties are satisfied:
  - Correctness: P can not deceive V. In other words, if P does not know how to prove the statement, then the probability that P convince V that he can prove the statement is very low.
  - Completeness: V can not deceive P. If P knows how to prove the statement, then P can convince V that he can prove it in a high possibility.
  - Perfect Zero knowledge: V can not acquire any additional knowledge rather than the statement.

IZK and NIZK

The classification of zero-knowledge proof

- There are two class of zero-knowledge proof: interactive zero-knowledge proof (IZK) and non-interactive zero-knowledge proof (NIZK)
- In IZK, the verifier proposes challenge for several rounds, and the prover needs to solve the challenge. Each challenge contains some random parameters, so it could not be predicted by the prover.
- However, it is not a good idea to use IZK in blockchain system, because it is hard for a prover to be challenged by every nodes in the network.
- The Fiat-Shamir transform can convert the IZK into NIZK using hash function. The hash value of the commit can be used to generate the challenge.

Current NIZKs in blockchain

- In 2013, the zero coin protocol is proposed using "Dynamic accumulators" to achieve NIZK in blockchain system for the first time. However, the denomination of coin is fixed.

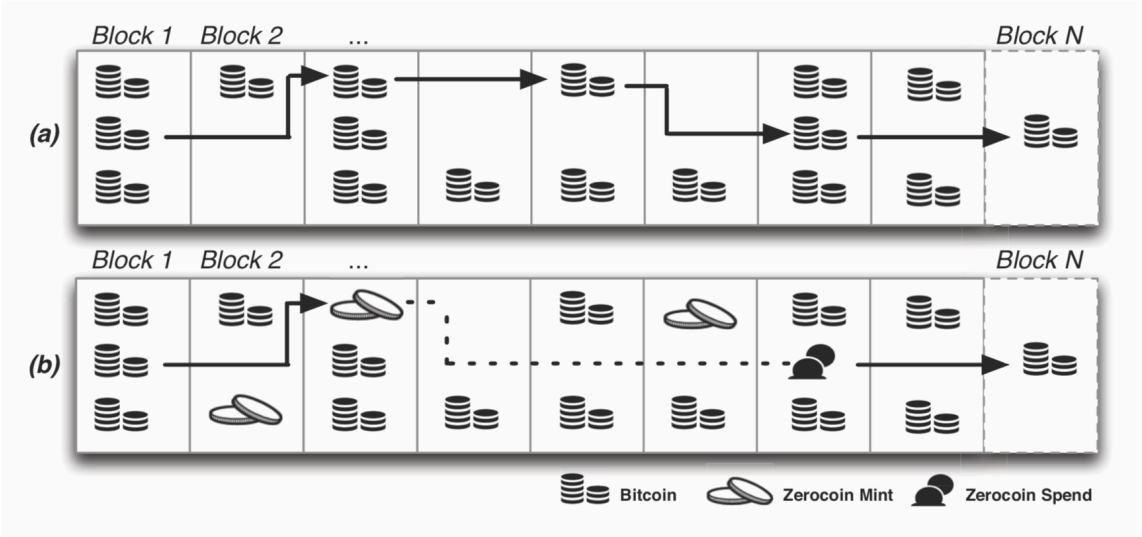


Figure 1: Two example block chains. Chain (a) illustrates a normal Bitcoin transaction history, with each transaction linked to a preceding transaction. Chain (b) illustrates a Zerocoin chain. The linkage between mint and spend (dotted line) cannot be determined from the block chain data.

Current NIZKs in blockchain

- In 2018, zero-knowledge Succinct Non-interactive ARguments of Knowledge (zk-SNARKs) is proposed in Zerocash project.
- Zerocoin and Zerocash need the trusted third party in the system to generate the secret parameters for the protocols, which is hard to accept in blockchain system.
- In 2018, zero-knowledge Scalable Transparent ARguments of knowledge(zk-STARK) is proposed to achieve transparent generation of the zk-proof system.
- The efficiency of zk-STARK is higher than zk-SNARK, however, the size of proof in zk-STARK is larger than that in zk-SNARK.