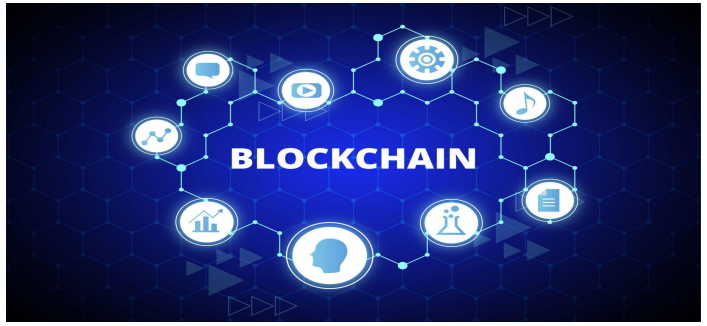
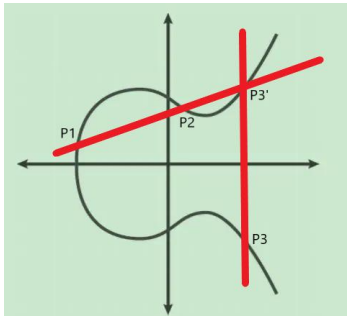
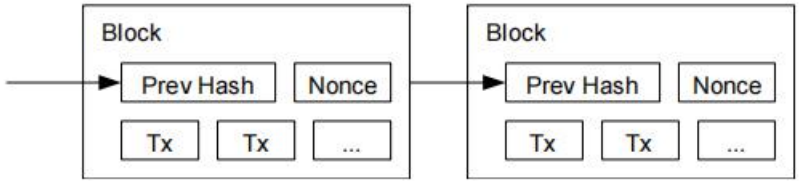
How Cryptography Benefits our Daily Life:

An Overview on **Blockchain System** and **Related Cryptographic Methods**







Author & Speaker:

Zerui Cheng (2019012355)

IIIS, Tsinghua University

Catalogue

1

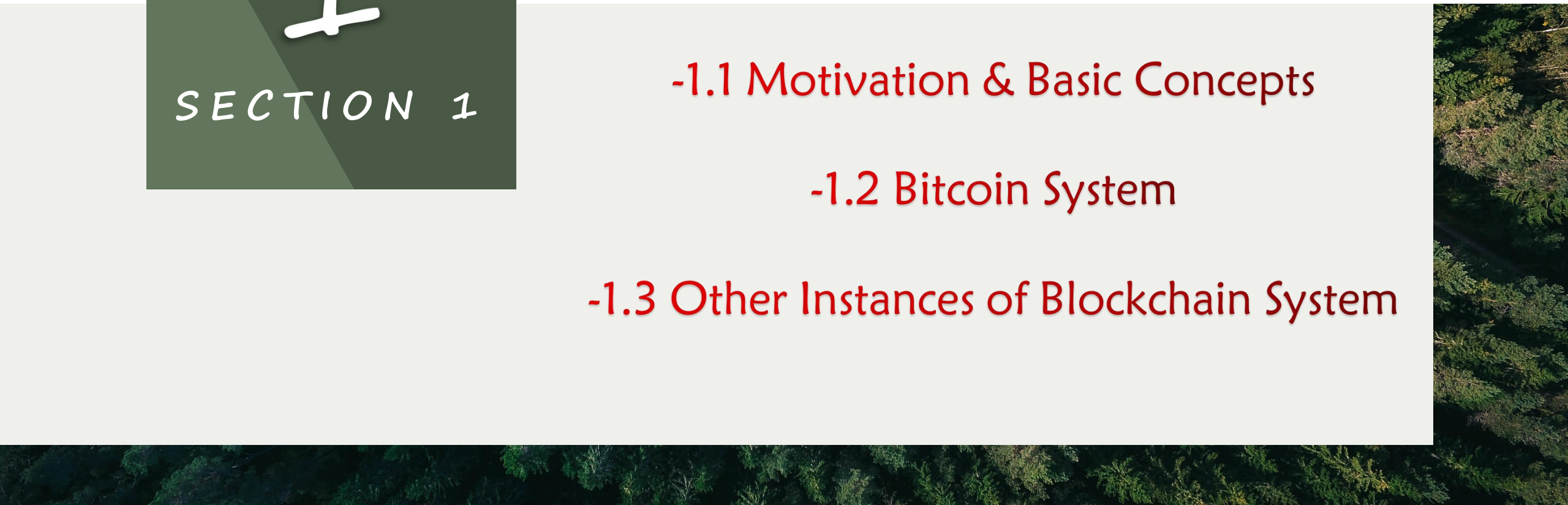
SECTION 1

Introduction to Blockchain System

-1.1 Motivation & Basic Concepts

-1.2 Bitcoin System

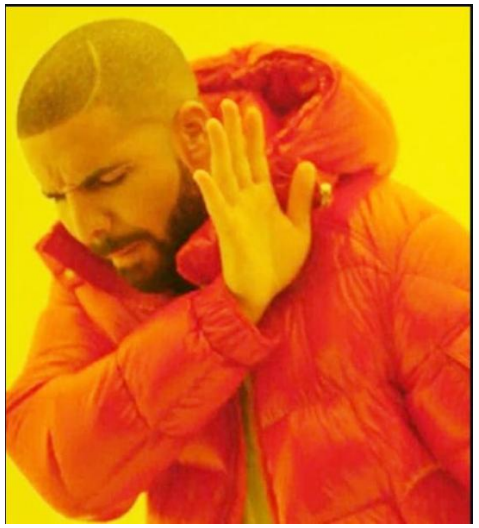
-1.3 Other Instances of Blockchain System

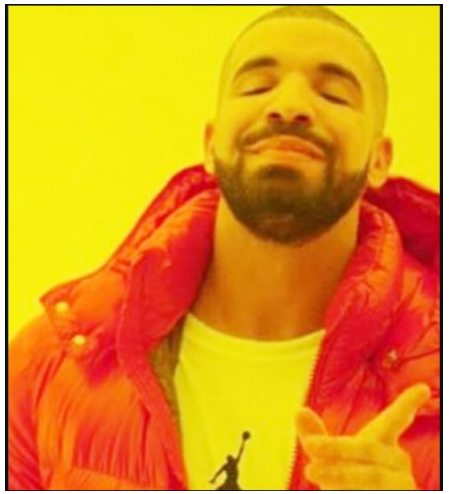


δ 1.1 Motivation & Basic Concepts

- Inherent disadvantages of centralized system：

Under the assumption that everybody is selfish and interest-driven, there’ s no reliable and unconditional trust. Thus centralized systems can be easily attacked by interest-driven malicious users, especially when the malicious users become authorities or coordinators.



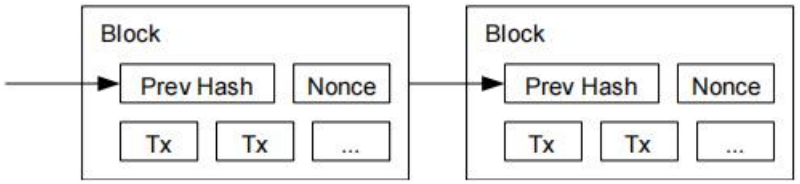


- Concept of blockchain system:

A digital trust platform, where the trust is based on computational hardness, which came into prominence through the birth of Bitcoin, a cryptocurrency introduced by S.Nakamoto in 2008.

δ 1.1 What is Blockchain?

- The answer is simple: blockchain = block + chain



- Blockchain as a data structure:

A linked list that uses hash pointers instead of regular pointers.

Here, a block is a data type that contains ‘hash pointer’ and some ‘data’.

The hash pointer field is simply the hash of another block, which we call its parent. A sequence of such blocks form a chain called blockchain.

- Blockchain system:

More complicated. Generally, it’s a digital trust system based on blockchain data structure. And we take Bitcoin as an example to elaborate it (Note that, Bitcoin is just a good instance of blockchain, regarding blockchain as Bitcoin or cryptocurrency is one-sided and incorrect).

δ 1.2 Appetizer: Bitcoin System

This part is generally based on this paper (the white book of Bitcoin) :

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Spoiler: As time flies, some proof of this paper are shown to be insufficient and some conjectures are shown to be incorrect. Bitcoin system suffers from some attacks such as selfish mining, block-withholding attack, eclipse attack and so on, which contradicts the assumption made by Nakamoto that any malicious user can't gain profit if their computation power is less than 50%, but good news is that the threshold doesn't decrease too much (e.g. 25.7% for selfish mining).

China

China crypto mining business hit by Beijing crackdown, bitcoin tumbles

Reuters

The news appears malapropos this morning. Nevertheless, we're talking about pure tech.

δ 1.2 A Sketch of Bitcoin

- **Idea of bitcoin**:
- an electronic payment system based on **cryptographic proof instead of trust**
- transactions are **computationally impractical to reverse**
- **don't need a trusted third party** any more

- **Implementation**:
a **P2P distributed** timestamp server
generate **computational proof** of the chronological order of transactions

Timestamp Server: Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones before it.

δ 1.2 Definition of a Coin and Transfer of a Coin

- **An electronic coin** : a chain of **digital signatures**

- **Transfer of a coin**:
  - digitally sign **a hash of previous transaction and public key of next owner**
  - add to the end the coin (for the concrete scheme, wait for Section 2.4)

  Here, the signatures indicate chain of ownership and can be verified by the payee.

- **Prevent double-spending:**

  A common solution: an intermediate central authority (not decentralized--->**weakness**)

  New solution in this paper:

  The **earliest transaction counts** and don't care about later attempts to double-spend.

  Participants **agree on a single history** of the order in which they were received.

δ 1.2 PoW (Proof-of-Work) System

- To implement the well-known "one-CPU-one-vote" mode, we need each user of the system to provide a proof for how much computational power it has.

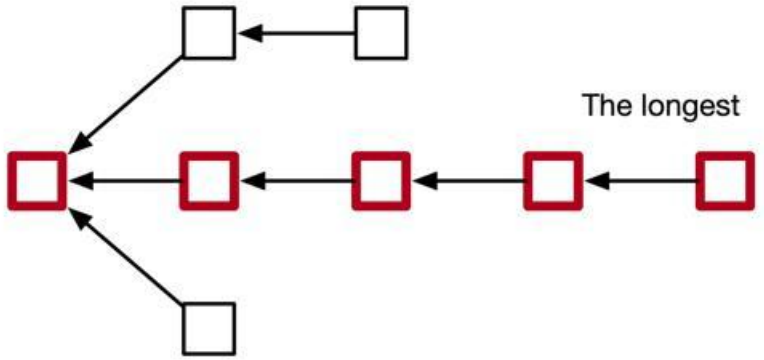
- In Bitcoin system, miners are asked to scan for a value (called nonce) such that, when hashed by SHA256D algorithm, it begins with a number of 0-bits (adjustable,about 70-80 nowadays). Only with a valid nonce value (easy to verify by the public), its block will be accepted by the public and transactions in it will be valid, according to the consensus of the community.

- Thus how much computational power one agent has determines how significant he/she is in the system. Detailed analysis on how it preserves security and yields the "one-CPU-one-vote" manner will be shown in Section 2.3.

δ 1.2 Longest Chain Rule (Nakamoto Consensus)

- **Ideal case: All blocks will form a chain and every block will be confirmed at last.**

- **Unfortunate Reality: Branches may occur in the system due to network delay.**



Longest Chain Rule: At any time, for a miner, he/she always mines on the end of the longest chain from his/her view of the whole system (may be different from the exact state of the state of the system, due to network delay). The scheme guarantees that the consensus is based on majority of the community. Transactions only packed in orphaned blocks won't be confirmed and need a re-broadcasting process.

δ 1.3 Other Instances of Blockchain System

Ethereum (By V.Buterin):
- concept of uncle block as a generalization of parent block
- gas limit and gas fee
- decreases the latency, increases the throughput
- second generation of blockchain

Conflux (By IIIS members Li Chenxing and Long Fan):
- Generate a topological order to avoid any orphaned blocks, and apply Link-Cut Tree algorithm to speed up the process, which increases the throughtput in order of magnitude
- GHAST protocol: decrease confirmation time

CONFLUX

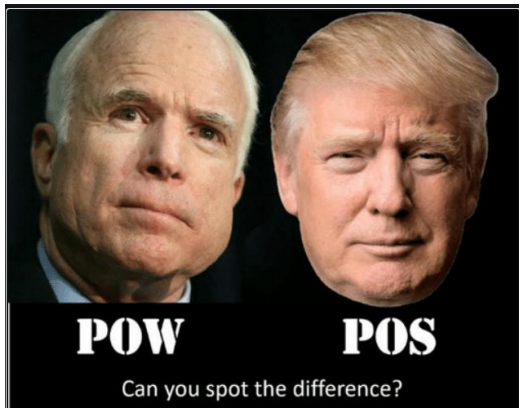
δ 1.3 Other Instances of Blockchain System

Hyperledger (By Linux Community):

- No correspsonding cryptocurrency, thus hard to be intervened by market
- Users should be granted access to the system instead of free entering
- Widely used in enterprise-grade blockchain deployments and building distributed system

PPCoin (themed Proof-of-Stake (PoS) System):

- PoW System is eco-unfriendly (power-consuming)
- Decide users' significance of vote by amount of coins they have: "one-coin-one-vote" mode
- Security: Those who are more significant hold more coins and thus has less motivation to attack (which will only result in devaluation of their coins in stake)
- Fatal Weakness: Nothing-at-Stake Attack



2

SECTION 2

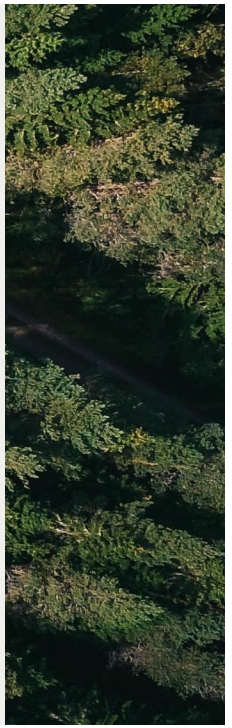
How Cryptography Applies in Blockchain System

- 2.1 Recapture of definitions

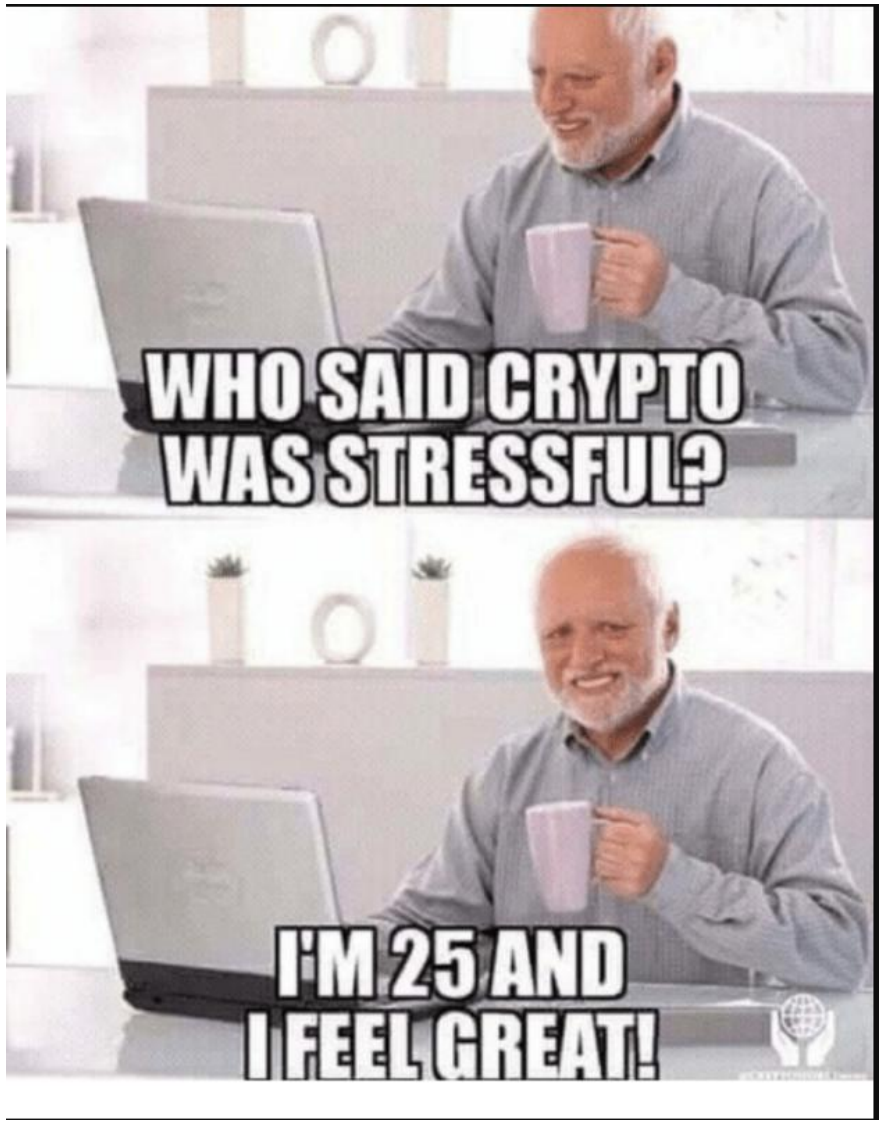
- 2.2 Merkle-Damgard Construction

- 2.3 SHA256D for Incrementing a Nonce

- 2.4 Ellipic Curves for Digital Signatures



δ 3.0 Fake News! (Some common misunderstanding)



δ 3.1 Recapture of Definitions

Definition 3.1. (Hash Function)

A hash function is a function H that converts a binary string x of arbitrary length to a binary string $H(x)$ of fixed length with the following two properties:

- *Easy to compute*: $\exists$ n.u.p.p.t A that computes $H(x)$ for any input x with success probability 1;
- *Collision-reducible* : Suppose the input space of x is X and the output space of $H(x)$ is Y, then $H(x)$ should distribute uniformly on Y when we go through all possible input $x \in X$.

Definition 3.2. (Cryptogragphic Hash Function, CHF)

A cryptographic hash function is a hash function $H : \{0,1\}^m \to \{0,1\}^n$ with the following property in addition:

- *Collision-resistant* : Any n.u.p.p.t *Adv* can't come up with $x \neq x'$ satisfying $H(x) = H(x')$ with non-negligible probability. (i.e. Given $x \in X$, there's no exponential speed-up to find $x' \neq x$ satisfying $H(x) = H(x')$ in contrast to exhaustive search).

Definition 3.3. (Puzzle-friendly CHF, PCHF)

A puzzle-friendly cryptographic hash function is a CHF $H : \{0,1\}^m \to \{0,1\}^n$ with the following property in addition:

- *Puzzle-friendly* : Any n.u.p.p.t *Adv* can't come up with x satisfying $H(x) \in Y \subset \{0,1\}^n$ where $\frac{|Y|}{2^n} \leq \epsilon(n)$ for some negligible ϵ with non-negligible probability. (i.e. It's just a generalization of the "one-way" property. Note that here $|Y|$ doesn't necessarily be polynomial size, for example, $|Y| = 2^{\frac{n}{2}}$).

On the left : Definition of PCHF (used in 3.2&3.3)

Below: Definition of Unforgeable DS (used in 3.4)

Definition 3.4. (Digital Signature)

$(Gen, Sign, Ver)$ is a digital signature scheme over the message space $\{M_n\}_n$ if

- $Gen(1^n)$ is a p.p.t. which on input n outputs a public key p_k and a secret key s_k, i.e. $p_k, s_k \leftarrow Gen(1^n)$
- $Sign$ is a p.p.t. which on input a secret key s_k and message m outputs a signature σ, i.e. $\sigma \leftarrow Sign_{s_k}(m)$
- Ver is a deterministic p.p.t. algorithm which on input a public key p_k, a message m and a signature σ returns either "accept" or "reject". And it satisfies the following property: For all $m \in M$,

$$Pr[p_k, s_k \leftarrow Gen(1^n) : Ver_{p_k}(m, Sign_{s_k}(m)) = accept] = 1$$

Definition 3.5. (Unforgeable Digital Signature)

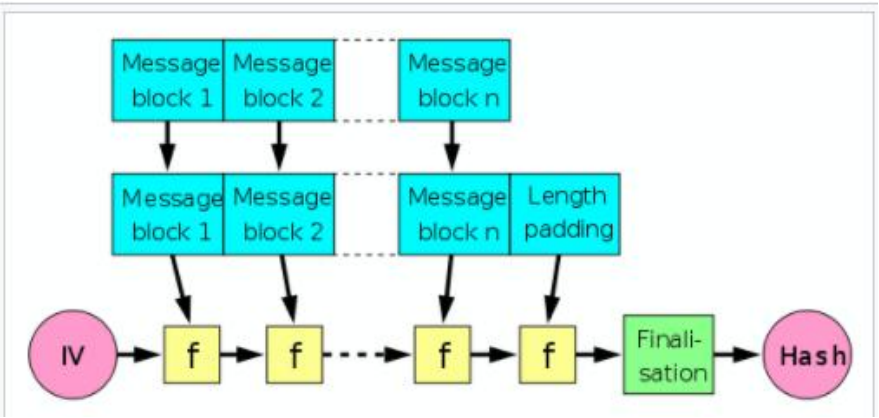
A digital signature scheme $(Gen, Sign, Ver)$ is unforgeable over the message space $\{M_n\}_n$ if it satisfies

- (unforgeable) For any n.u,p.p.t Adv, there exists negligible $\epsilon(\cdot)$ such that

$$Pr[p_k, s_k \leftarrow Gen(1^n) : Ver_{p_k}(m, Adv(m)) = accept] \leq \epsilon(n)$$

δ 3.2 Merkle-Damgard Construction

If we already have a PCHF from $\{0,1\}^m$ to $\{0,1\}^n$ where $m > n$, we may want to generalize it to one from $\{0,1\}^t$ to $\{0,1\}^n$ for arbitrarily larger $t > m$ while reserving its property as PCHF. And a general principle used in the construction of many cryptographic hash functions is the Merkle-Damgard construction as follows. For a message $x \in \{0,1\}^t$ for arbitrary long t, we first break x into portions and call them $x_1, x_2, ...$, where x_is are all of length $m - n$ (append 0 on the last if the last portion isn't long enough). Then we can apply the PCHF $F : \{0,1\}^m \to \{0,1\}^n$ recursively, for the first iteration, the input string is $u_0 = l$ (where l is a fixed string) and we get $F(u_0) = y_0 \in \{0,1\}^n$. Then for the i^{th} iteration where $i \geq 1$, the input string is $u_i = y_{i-1} \circ x_i \in \{0,1\}^{n+(m-n)} = \{0,1\}^m$, and the output string is $F(u_i) = y_i \in \{0,1\}^n$. Suppose that the input string is divided into c portions and we get y_c eventually, then we compute $h = F(y_c \circ l)$ where l is the bit representation of t ($t < m - n$ must hold for the process to be carried out in polynomial time where $t = O(logn)$ scale). Finally, the output string of the generalized hash function is h. We can prove that the generalized hash function still holds the property of collision-resistant by contradiction, and thus the Merkle-Damgard construction also results in a cryptographic hash function from $\{0,1\}^t$ to $\{0,1\}^n$.



Key Property:

If the original PCHF is collision-resistant and puzzle-friendly, then the generalization also holds the two properties (proved in a way similar to Hybrid Argument shown in class).

In practice:

- Finalisation function (for better mixing effect and avalanche effect (i.e. Small variation in input will affect the output drastically)).

δ 3.3 SHA256D Algo for Incrementing a Nonce

SHA-256-D =
SHA: Secure Hash Algorithm + 256: Bits of the image of the function + D: Double

- Put forward by National Institute of Standards and Technology in 2001

- Based on a compression function from 768-bit to 256-bit and Merkle-Damgard Construction

- Its concrete scheme is complicated and tedious bitwise operations. Moreover, it lacks proof for its security, but there doesn't exist known attack to it, either. Thus we omit it here. Usually, in related research area, we often regard it as an oracle with the property as PCHF.

- It's used to increment a nonce value in a block in Bitcoin system, where the hash value should have enough number of 0s in its prefix to be confirmed by the community.

- Criterion for a Valid Block:
  SHA256D(Version,hashPreBlock,hashMerkleRoot,Timestamp,Bits,Nonce)<=MAXTARGET/Diff

δ 3.3 Analysis on Why it Implies "one-CPU-one-vote"

- Here we only care about each round of competing to generate a new block.

- It's essentially a Poissonian process.
  - Reason: For miners with computational power x and kx, at the time when a nonce is found, Miner 2 scans k times candidates than Miner 1, thus the probability to own the block is k times Miner 1. Afterwards, the situation comes back to initial and it's an independent new round.

- One-CPU-One-Vote: The expected proportion of the miner's blocks on the whole chain is determined by and proportional to its computational power, and the true proportion will converge to the expectation in the long term from the Central Limit Theorem.

- Security: As a result, if most of the community is honest, a malicious miner can hardly hurt the system since the authenticated blocks produced by him is very limited, and his malicious behavior can be easily compensated by the following block, which won't hurt the system at all.

δ 3.4 Elliptic Curves for Digital Signatures

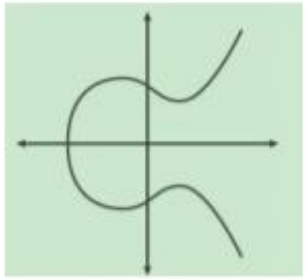
- Elliptic Curve can be used to generate digital signatures, and Bitcoin System applies SECP256K1 in the family of ECSDA. (ECDSA: Elliptic Curve Digital Signature Algorithm)

- Let's first see what elliptic curve is.

As shown in its name, we first need an elliptic curve as the basis of our algorithm. And the elliptic curve is in the form:

$$y^2 \equiv x^3 + ax + b \pmod{p}$$

And the curve is plotted in the following figure.



- Then, let's see the addition on the curve.

Then, some involved maths derivation shows us that, for all points (x, y) on the curves such that $x, y \in F_p$, they essentially form a field. And we use $E(F_p)$ to denote it, which is called the elliptic curve field specified by the function above.

For the field $E(F_p)$, the addition is defined as follows:

Suppose $P_1, P_2 \in E(F_p)$ where $P_1 = (x_1, y_1), P_2 = (x_2, y_2)$, then $P_1 + P_2 = P_3 = (x_3, y_3)$, where

$$x_3 \equiv \lambda^2 - x_1 - x_2 \pmod{p}$$

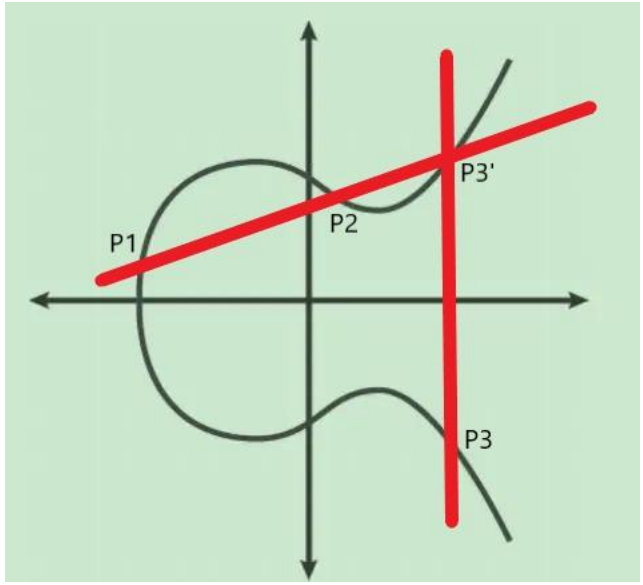
$$y_3 \equiv \lambda(x_1 - x_3) - y_1 \pmod{p}$$

Here $\lambda \equiv (x_2 - x_1)^{-1}(y_2 - y_1) \pmod{p}$ if $P_1 \neq P_2$ and $\lambda \equiv (2y_1)^{-1}(3x_1^2 + a) \pmod{p}$ if $P_1 = P_2$, where x^{-1} denotes the inverse of x modulo p.

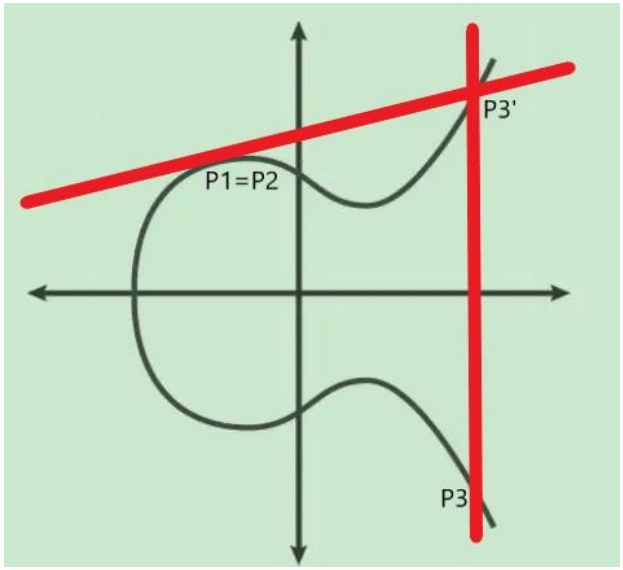
δ 3.4 Elliptic Curves for Digital Signatures

- Intuition for addition operation on elliptic curves

P1 ≠ P2



P1 = P2

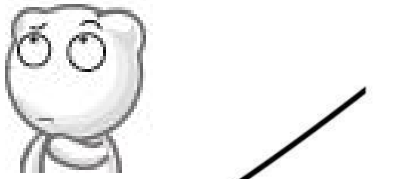


- Multiplication by a constant k for a point on elliptic curves:
  Simply do the addition for k times;
  Can be computed by O(logk) additions by binary speed-up
  (correctness: associative law of the field)

δ 3.4 Elliptic Curves for Digital Signatures

- Then let's focus on the concrete scheme of ECDSA.

Then we select N points on the curve and choose one of the points to be the origin point G, here N is the size of the message space and we can make a projection from all possible messages to the range $[0, N-1]$. The selection of the points will depend on the cofactor $h = \frac{\#(E(F_p))}{N}$ (i.e. density of selected points), where $E(F_p)$ is the elliptic curve field specified by the function above, which is composed of points on the curve. Then the 6-tuple (a, b, p, N, G, h) specifies an elliptic curve, which is also known to the public. Note that p, N should both be prime (we can always find appropriate p and N according to the distribution of primes). In particular, for Bitcoin, we choose $a = 0$, $b = 7$ and p, N are both large primes around 2^{256},laying the foundation of the algorithm.



δ 3.4 Elliptic Curves for Digital Signatures

- Then let's focus on the concrete scheme of ECDSA.

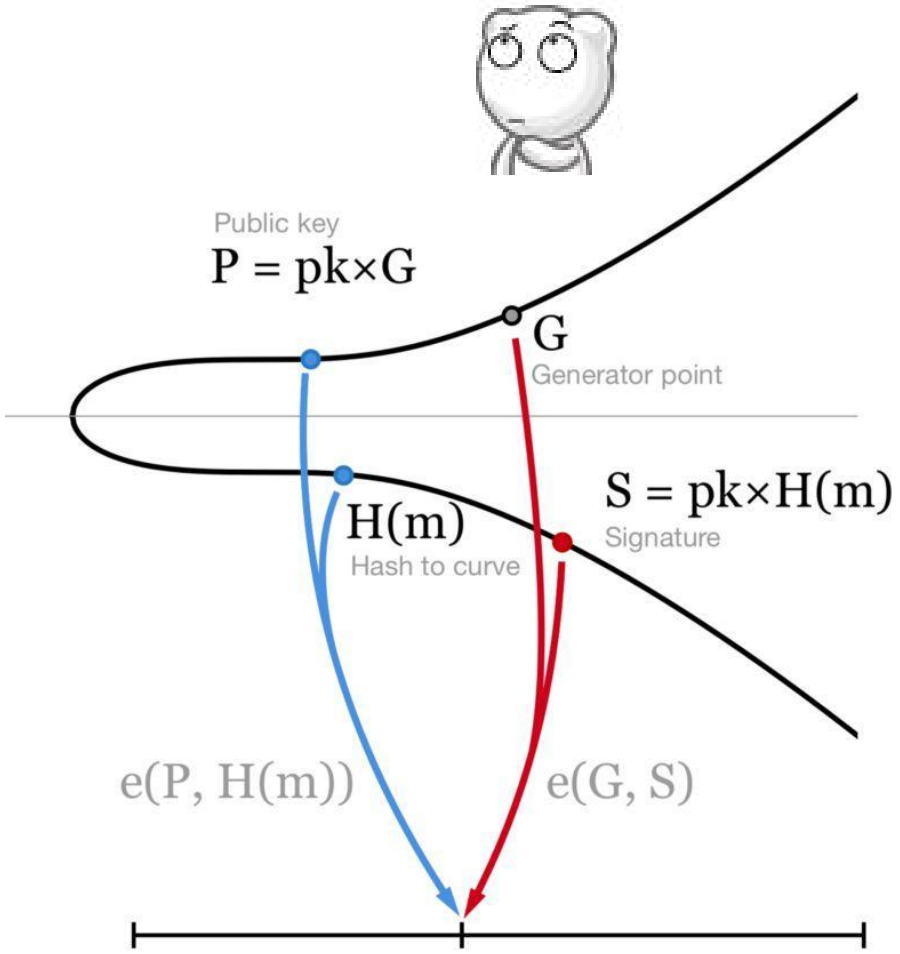
Then we will do some magic on the elliptic curve in ECDSA. For the *Gen* process, the secret key s_k is a purely random number. And we can compute $Q = s_k \times G$ as another point on the elliptic curve and take Q as the public key p_k. Then, for the *Sign* process, first generate a random number k and compute the point $P = k \times G$ on the curve, then suppose z is the hash value of the message M to sign, P_x is the x-coordinate of the point P , then it computes the following value

$$S \equiv k^{-1}(z + s_k \times P_x) \pmod{p}$$

Then it outputs $\{P_x, S\}$ as the signature σ. Finally, for the verifier $Ver_Q(z, \{P_x, S\})$, it can restore the point P computed by *Gen* in the following manner:

$$P = S^{-1} \times z \times G + S^{-1} \times p_k \times Q$$

Then it compares P_x in σ with the x-coordinate of the computed point P. If they're equal, then the signature is valid and $Ver_Q(z, \{P_x, S\})$ returns accept. Otherwise, the signature is an invalid one and $Ver_Q(z, \{P_x, S\})$ rejects.



δ 3.4 Elliptic Curves for Digital Signatures

How ECDSA works in Bitcoin system for verification of a Transation

- In practice, every user of Bitcoin system holds a pair of keys (Si,Pi) where the secret key Si is private and the public key Pi is broadcast to all as the only certificate of identity in the system.

- To sign a transaction, the sender signs the previous hash and the receiver's public key as the message in the digital signature and append it on the end of the coin.

- Then, the miners can verify the authentication of the transaction by running the Verifier. Once the transaction is verified, it goes to the waiting list to be packed in a block. After the block is appended to the chain and gets confirmed by majority of the community (A usual criterion is for Bitcoin that, 6 blocks generated behind the block are all on the longest chain), the transaction eventually comes to a success.

Current Limitations and Directions for Further Research

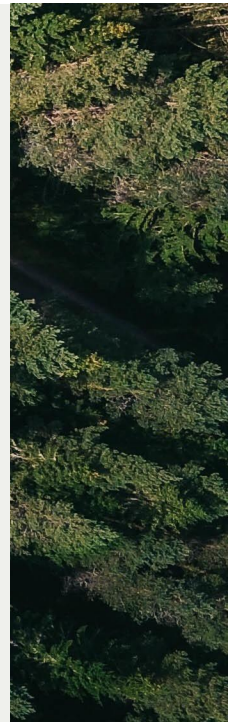
3

SECTION 3

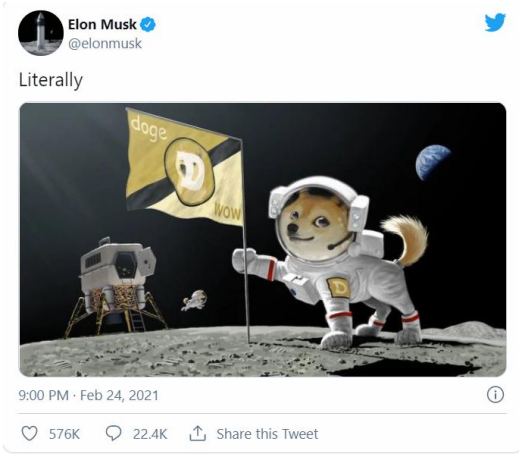
- 3.1 Social Issues

- 3.2 Environmental Issues

-3.3 Technical Issues



δ 3.1 Social Issues



- Roller-coaster style price curve
- Manipulated by some conscienceless capitalists
- Attract gamblers to join in the community (not the ideal audience, however, blockchain should aim at those who need a digital trust platform), and they may become unstable factors of the society once they lose.

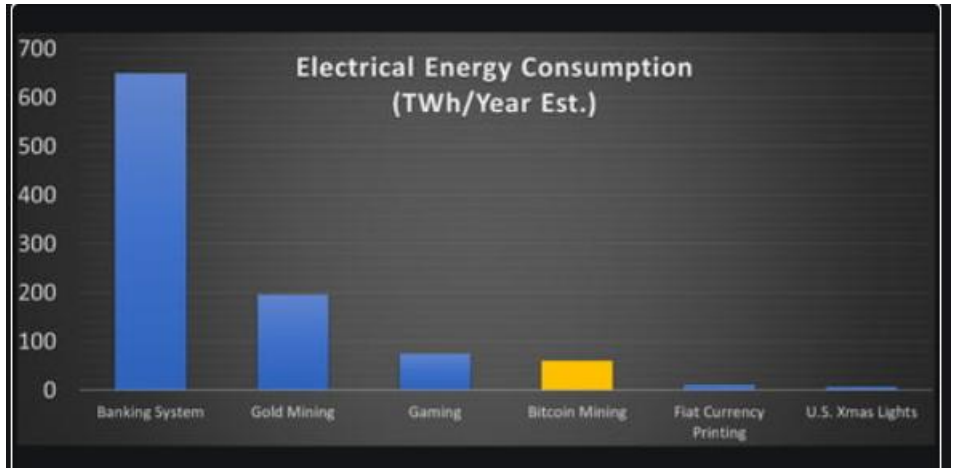
- Activate illegal behaviors such as drug trading, gun trading , kidnapping, extortion etc.

- Lack effective supervision as a decentralized system

- As a kind of cryptocurrency themed decentralization, it's essentially a threat to the authorities of the government.

- Concession: Consortium-chain, a centralized blockchain which provides digital trust for enterprise-grade coopearion. It's a new branch of blockchain and needs further research.

δ 3.2 Environmental Issues



- PoW (Proof-of-Work) system is eco-unfriendly, consuming too much energy and power, and thus violates the purpose of sustainable development.

- Idea for fixing the problem:
  Put forward new consensus to substitute it.

- Here, PoS (Proof-of-Stake) and DPoS are both good attempts, but they still have unsolved weaknesses today (The first suffers nothing-at-stake attack while the second is centralized) . How to fix and strengthen the consensus of PoS and DPoS is a heated topic.

- If we tackle this problem , then PoS can be widely used in blockchain, implying a new generation and a new era of blockchain system.

δ 3.3 Technical Issues

- Performance of blockchain is now still poor.

- Limited throughput and high latency are common problems, which doesn't live up to the expectation of daily use. The maximum capacity of transactions per unit time is limited, and the confirmation time for current blockchain is also too long.

(1) It can't hold the amount of transactions competent to WeChat Pay and Alipay.

(2) Users can't tolerate a long confirmation time, either.

- Now, in Conflux, Link-Cut Tree and GHAST are applied to improve these aspects. Although the effect is prominent, it's far from the expected stage to be commonly used. Thus this problem remains for more researchers to work on and suggests a direction for further research.

PART OF REFERENCES

[1] S.Nakamoto.2008.Bitcoin: A Peer-to-Peer Electronic Cash System.
[2] Coron, Jean-Sébastien, et al. "Merkle-Damgård revisited: How to construct a hash function." Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 2005.
[3] Gilbert, Henri, and Helena Handschuh. "Security analysis of SHA-256 and sisters." International workshop on selected areas in cryptography. Springer, Berlin, Heidelberg, 2003.
[4] Johnson, Don, Alfred Menezes, and Scott Vanstone. "The elliptic curve digital signature algorithm (ECDSA)." International journal of information security 1.1 (2001): 36-63.
[5] Buterin,Vitalik.”A next-generation smart contract and decentralized application platform.” white paper 3.37 (2014).
[6] Androulaki,Elli,et al. ”Hyperledger fabric: a distributed operatingsystem for permissioned blockchains.” Proceedings of the thirteenthEuroSys conference. 2018.
[7] Li,Chenxing,et al.”A decentralized blockchain with high throughput and fast confirmation.” 2020 USENIX Annual TechnicalConference (USENIXATC 20). 2020.
[8] Li,Chenxing, Fan Long, and Guang Yang. "GHAST: Breaking confirmation delay barrier in nakamoto consensus via adaptive weighted blocks." arXiv preprint arXiv:2006.01072 (2020).
[9] King, Sunny, and Scott Nadal. "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake." self-published paper, August 19 (2012): 1.
[10] Lecture notes of ECE 598CV in UIUC: Principle of Blockchains, instructed by Prof.Pramod Viswanath:\\ \em{https://courses.grainger.illinois.edu/ece598pv/sp2021/\#Lectures}

THE END

Thanks for your attention!

June 2, 2021