Blockchains with Proof-of-Stake

Hao Chung (鍾豪)

July 29. 2019

Outline

1. PoS-based Blockchains
   - overview to Bitcoin
   - Ouroboros Praos
   - Algorand
2. Two Crypto Notions
   - security parameter and asymptotic behavior
   - pairing
3. Experience in Industry

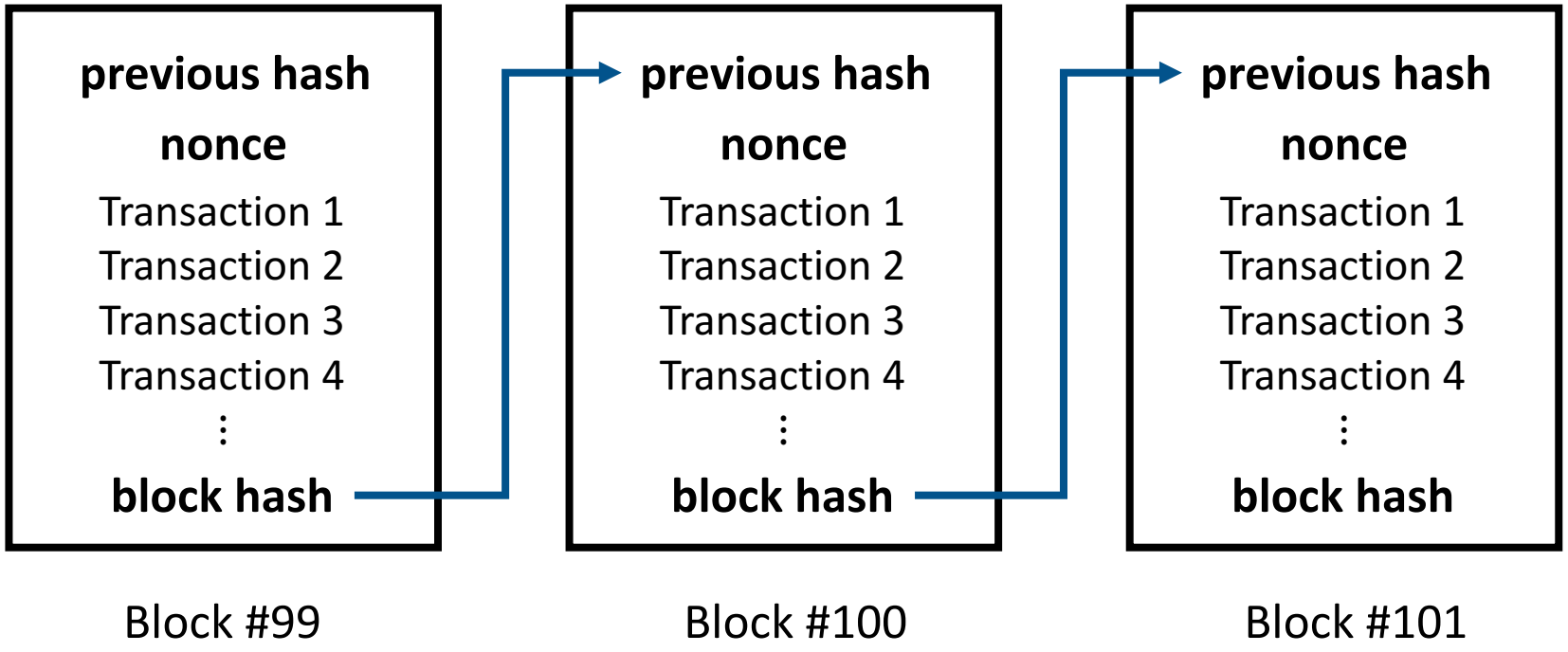
What is a blockchain?

What kind of functionality it wants to achieve?

What kind of data structure it uses?

What kind of method (algorithm) it uses to achieve the functionality?

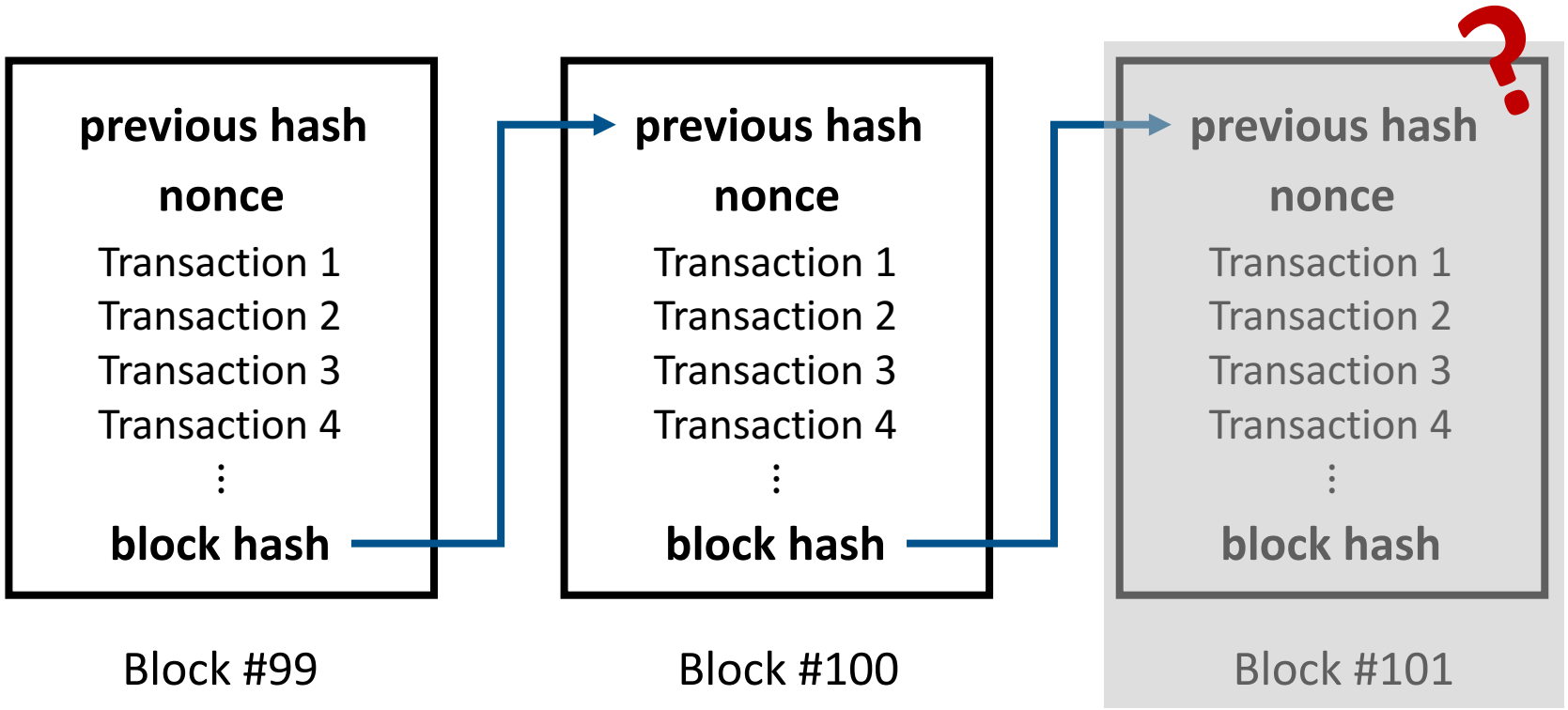
Structure of Blockchains



In this view, a blockchain is a distributed ledger linked by hash value.

Let's recall how Bitcoin works

How do the miners decide "who can issue Block #101?"



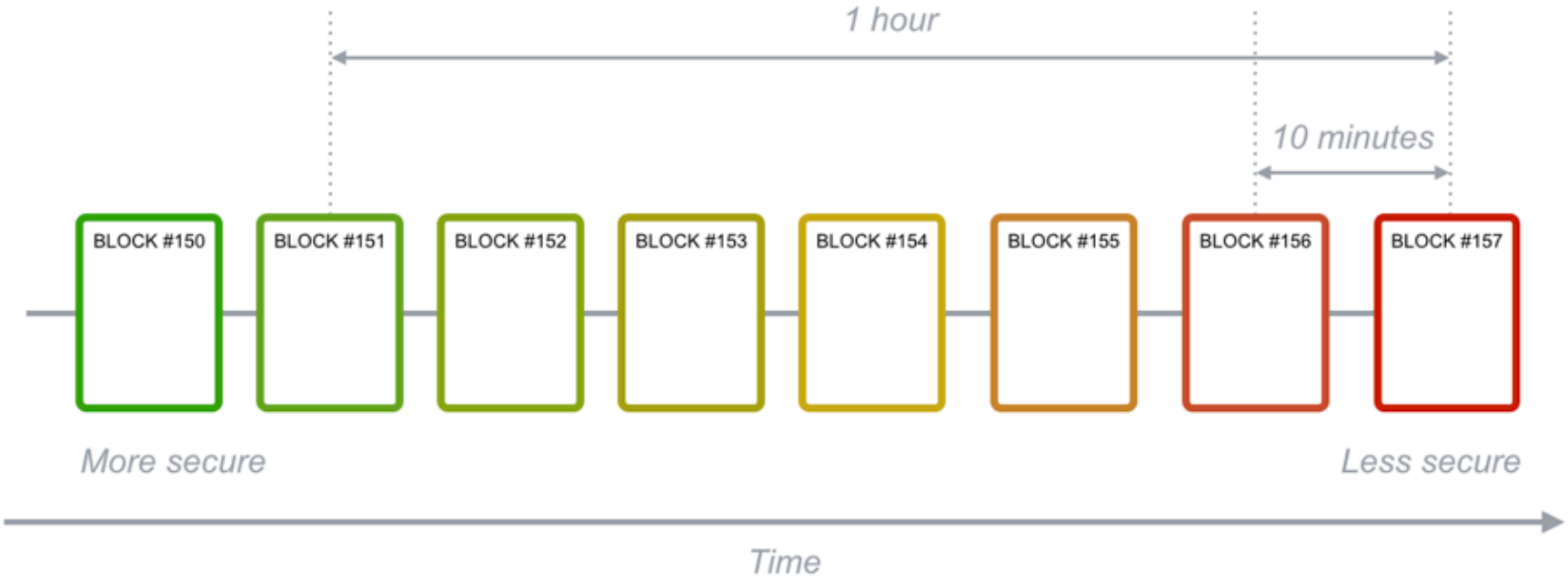
In Bitcoin, the block hash is restricted below a threshold.

The first miner create a block with small block hash can issue a block.

↑ try many possibilities of nonces → Proof-of-Work (PoW)

Let’s recall how Bitcoin works

When does a block become “confirmed?”



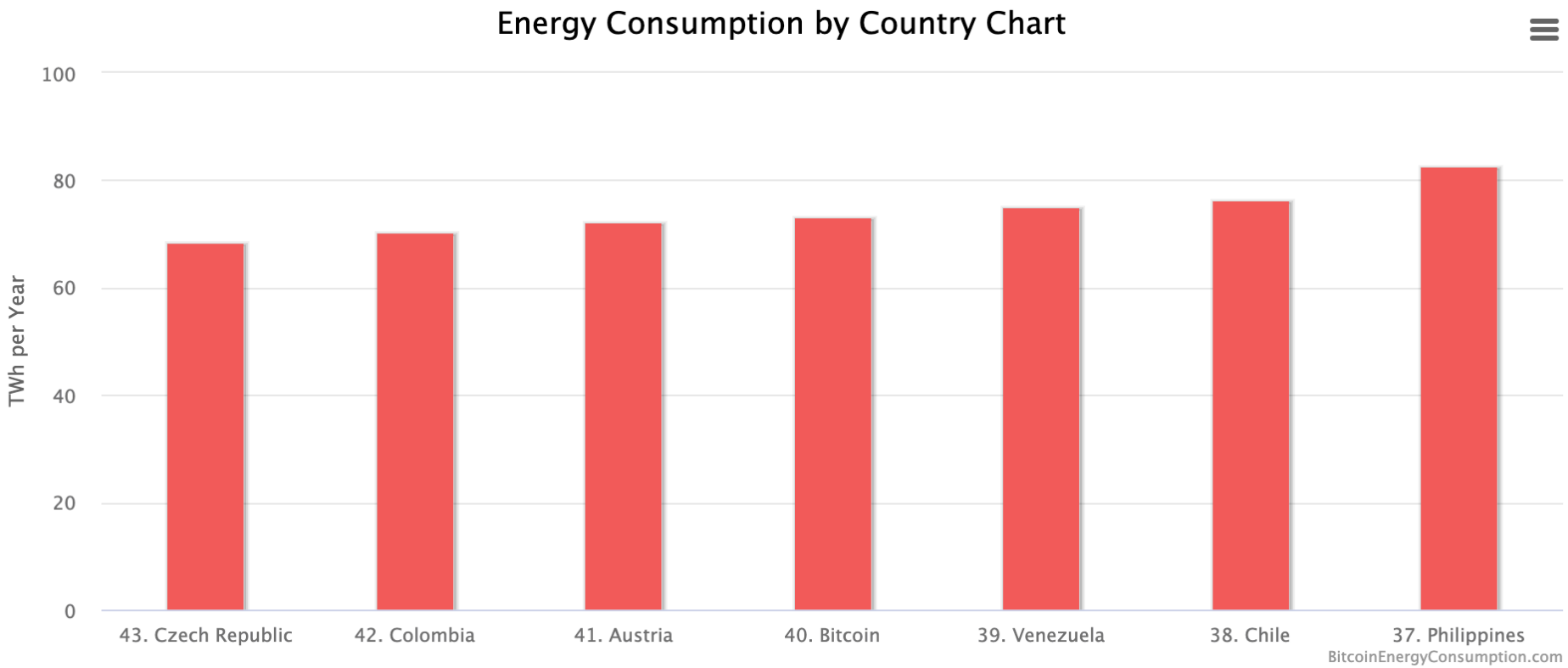
In practice, a block with six successors is recognized as confirmed.

In short, Nakamoto consensus has two main components.

limited resources that resist dummy accounts

the way that each miner reach a consensus

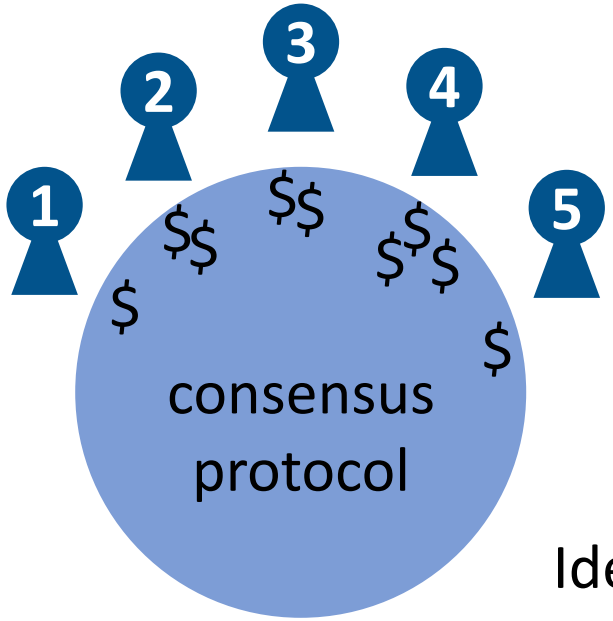
Energy Consumption of Bitcoin



at July 27. 2019

Proof-of-Stake

- Stake is the currency that a participant lock up in the protocol as a guarantee.
- The right to issue a block is related to the stakes that participants own.



Ideally,

$$\Pr(p_i \text{ issues a block}) \propto \Pr(p_i' s \text{ stake})$$

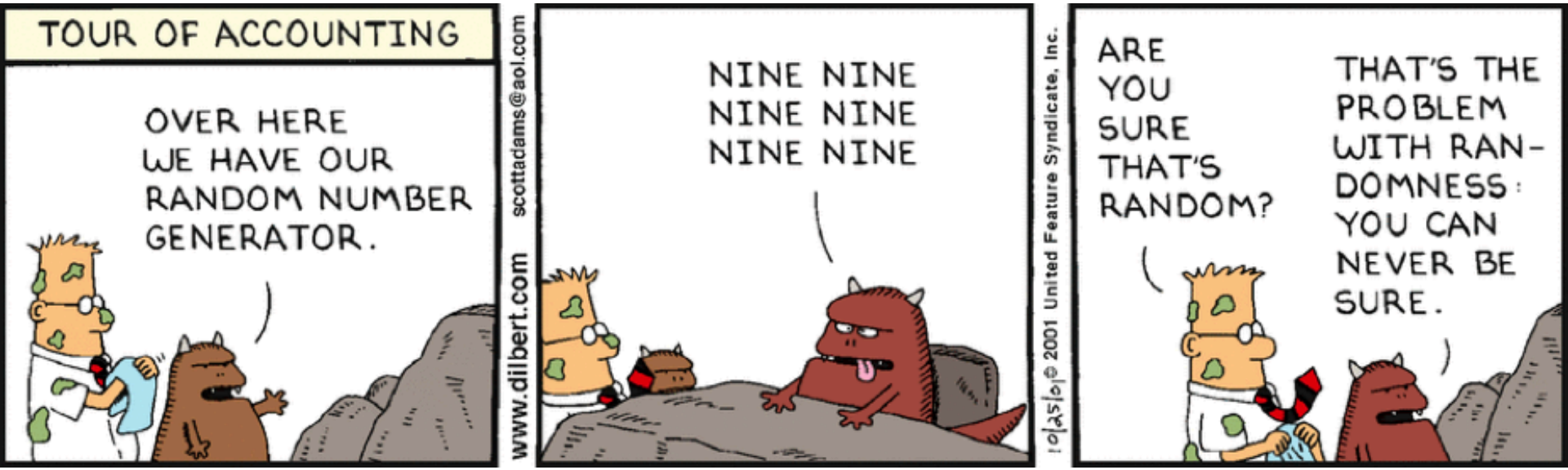
Ouroboros Praos

Proposed by Bernardo David, Peter Gazi, Aggelos Kiayias, and Alexander Russell in 2017.

Accepted by EuroCrypt 2018.

Compare to Bitcoin, Ouroboros Praos consists of

Proof-of-Stake ✚ **Longest Chain Rule**



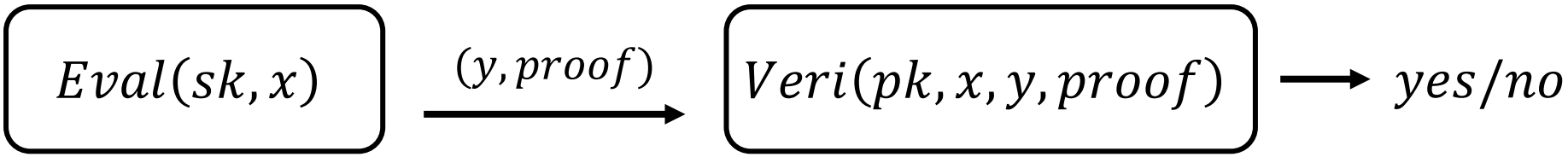
Can we verify the authentication of the generation of the random number?

Verifiable Random Function (VRF)

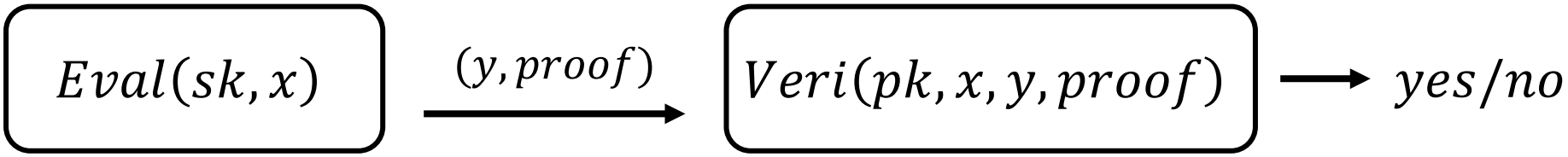
VRF is a function that generates a random number, where the computation can be verified.

In construction, a VRF is a 3-tuple of algorithms $(Gen, Eval, Veri)$ such that

- $Gen(1^{\lambda}) \rightarrow (pk, sk)$
- $Eval(sk, x) \rightarrow (y, proof)$
- $Veri(pk, x, y, proof) \rightarrow \{yes, no\}$



Verifiable Random Function (VRF)



A secure VRF must satisfy

- Complete Provability

  Suppose $(y, proof) = Eval(sk, x)$. Then
  $$\Pr(Veri(pk, x, y, proof) = yes) = 1.$$

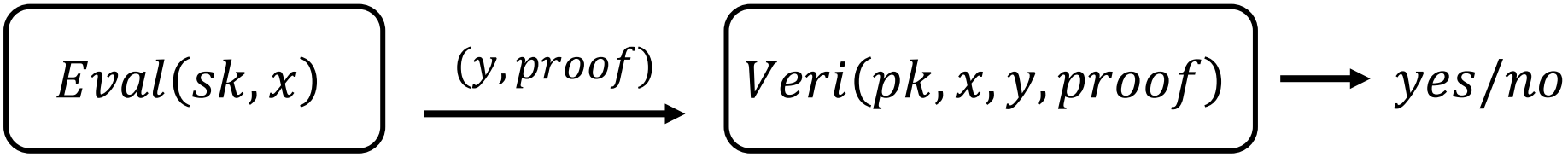
- Unique Provability

  No $(pk, x, proof_1, proof_2, y_1, y_2)$ such that $y_1 \neq y_2$ can satisfy $Veri(pk, x, y_1, proof_1) = Veri(pk, x, y_2, proof_2) = yes$.

- Pseudorandomness

  The generated y should be indistinguishable from a uniformly distributed string.

Verifiable Random Function (VRF)



Remind that the signature has the properties:

1. only the user with the secret key can generate a valid signature
2. everyone with the public key can verify the signature
3. without the secret key, the signature should be unpredictable

In practice, a VRF can be constructed by a unique signature and a random oracle.

That is

$$Eval(sk, x) = Hash\big(Sig_{sk}(x)\big).$$

Ouroboros Praos

In Ouroboros Praos, protocol is executed in "slots."

We define

- $data$ to be slot id and some block information (public)
- α_i to be the relative stake of participant p_i

For each slot, participant p_i is allowed to propose a block if

$$Eval(sk_i, data) < T(\alpha_i).$$

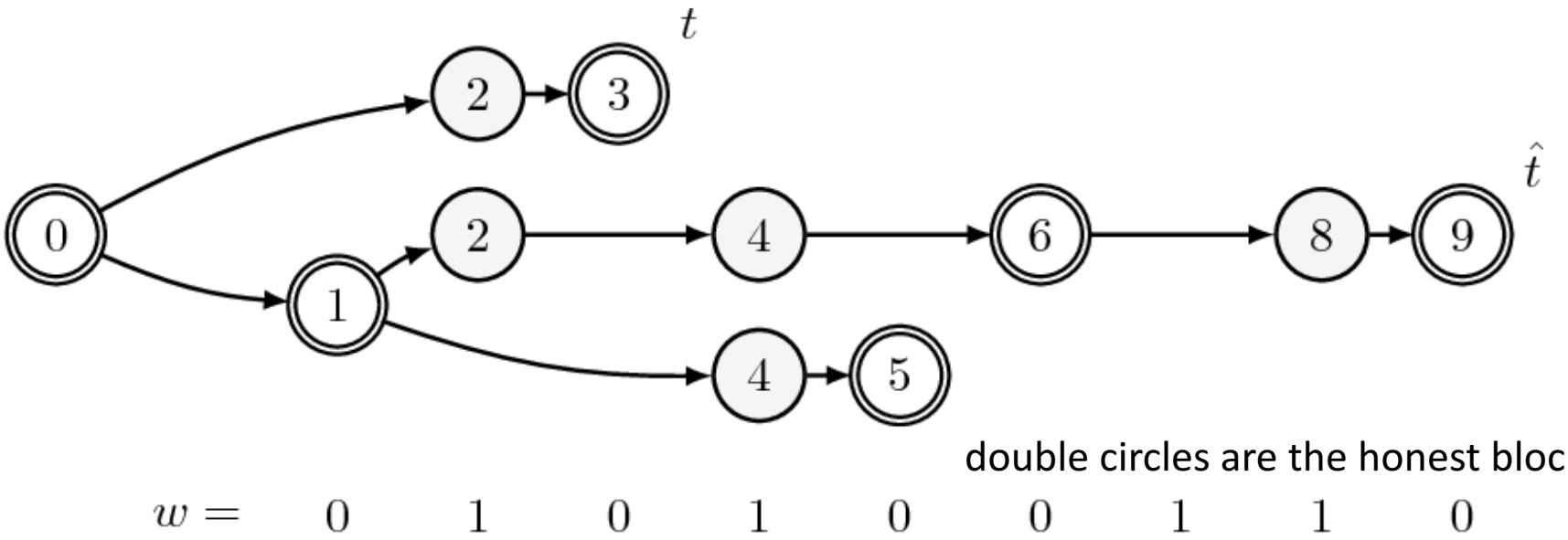
Other participants can easily verify the qualification by

$$Veri(pk_i, data, y, proof).$$

Ouroboros Praos

For each slot, there may be zero, one, or many proposed blocks.

In this case, the honests follow the longest chain.



David et al. showed that it is secure if the honests own $> \frac{1}{2}$ stakes.

Algorand

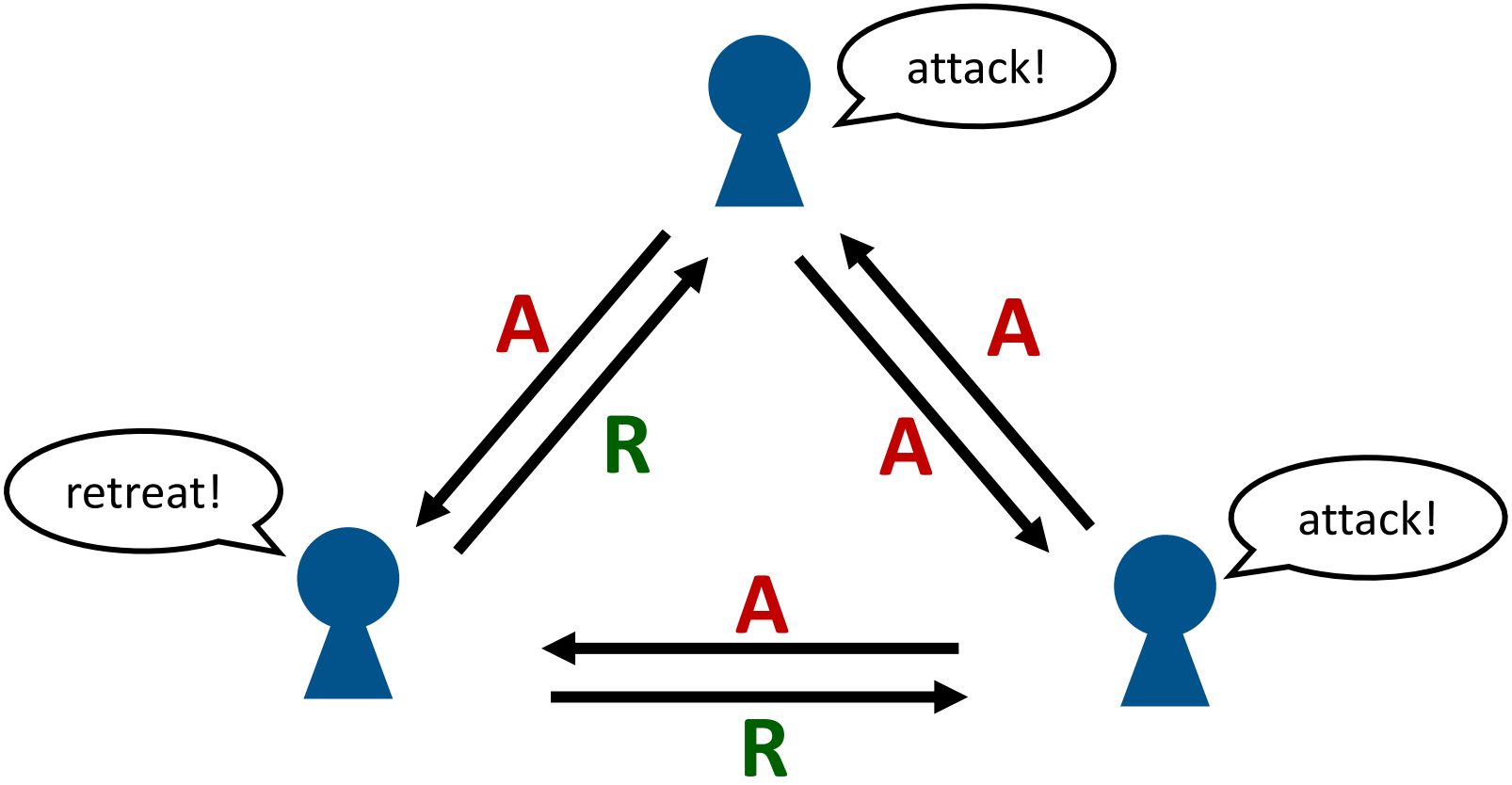
Proposed by Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, Nickolai Zeldovich in 2017.

Accepted by SOSP 2017.

Compare to Bitcoin, Algorand consists of

Proof-of-Stake  **Byzantine Agreement**

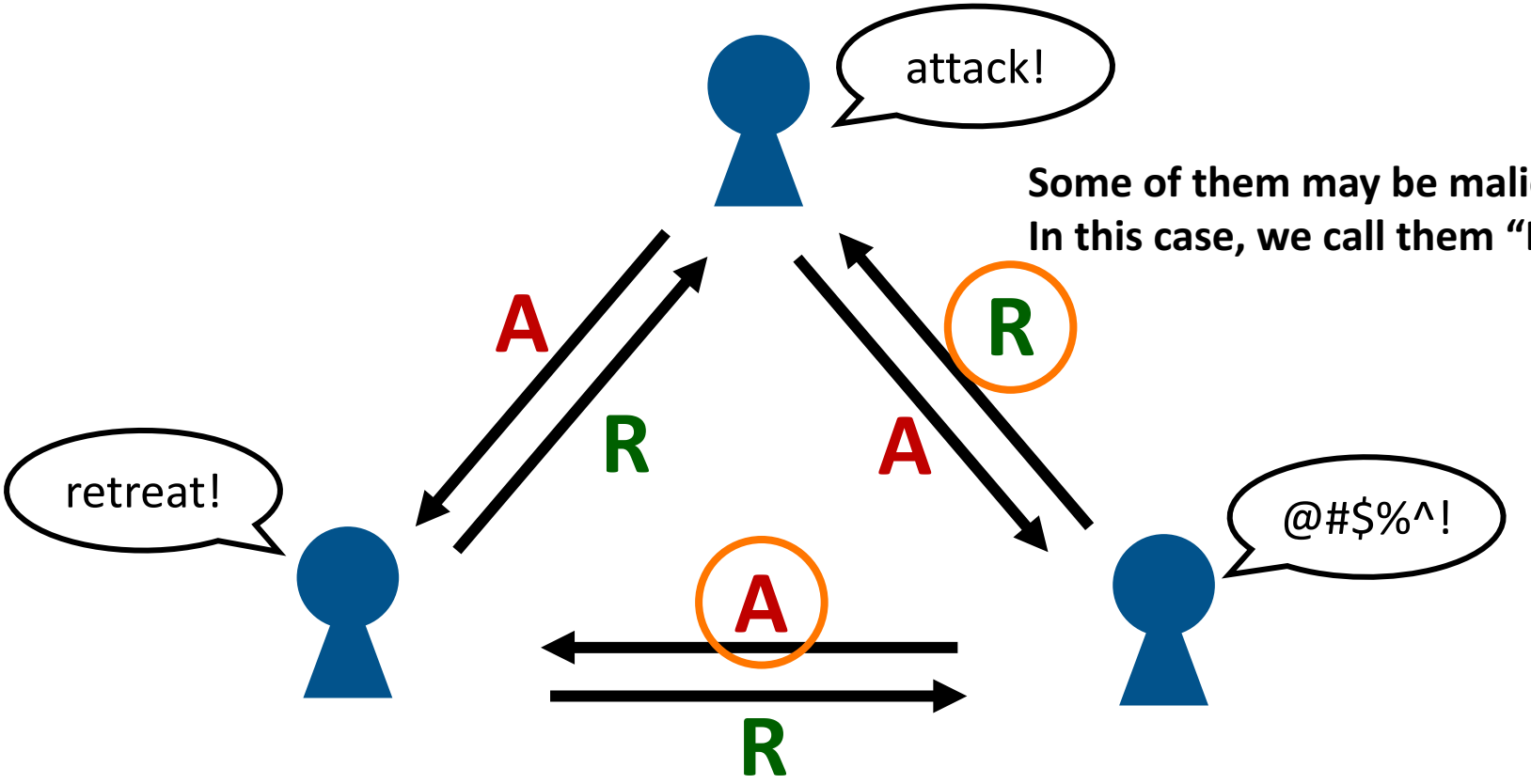
Byzantine Agreement



Suppose there are three generals.

They want to have a consensus that whether they should attack or not.

Byzantine Agreement



Suppose there are three generals.

They want to have a consensus that whether they should attack or not.

Algorand

Like Ouroboros Praos, the qualification of proposing blocks are decided by VRF.

For each block height, participant p_i is allowed to propose a block if

$$v_i = Eval(sk_i, data, coin\ id) < T.$$

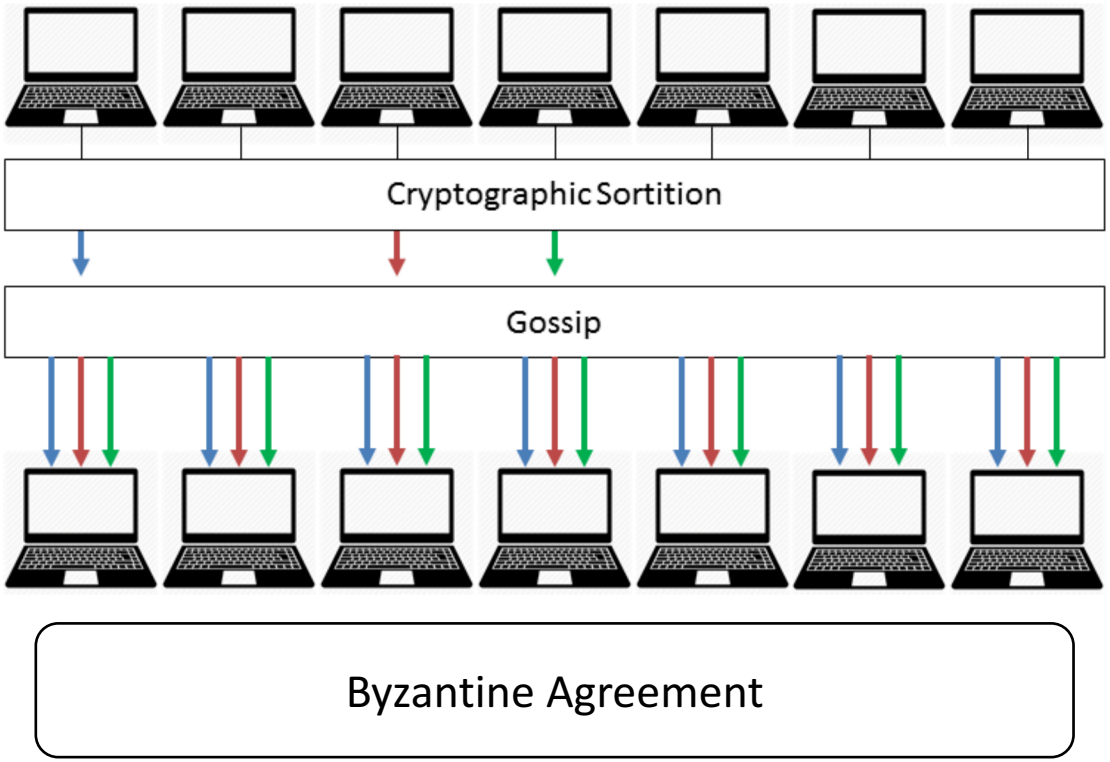
So that the more stake p_i, the more chances p_i can try

Ideally, the winner of the block is the one with smallest VRF value v.

However, how to make sure every participant have a consensus?

=> Byzantine agreement

Algorand


Cryptographic Sortition
Gossip
Byzantine Agreement
red is the winner

Outline

1. PoS-based Blockchains
   - overview to Bitcoin
   - Ouroboros Praos
   - Algorand
2. Two Crypto Notions
   - security parameter and asymptotic behavior
   - pairing
3. Experience in Industry

Is primality test in **P**?

Definition (Primes problem)

Given an integer N, decide whether N is a prime or not.

In 2002, Manindra Agrawal, Neeraj Kayal and Nitin Saxena finally showed that Primes problem is in **P**.

If we can try the division up to $\sqrt{N}$, why Primes problem $\in$ **P** doesn't hold trivially?

Key: the running time is counted in input size.

What is the time complexity of breaking AES-256?

Let‘s define problem of breaking-encryption as follow.

Given an encryption oracle $E_{sk}(\cdot)$, try to find the underling secret key sk.

Now, suppose our target is AES-256.

Then, what is the time complexity of breaking AES-256?

Permutation

How many permutations are there that maps n-bit-long strings to n-bit-long strings?

Let S_n denote the set of all the permutation mapping $\{0,1\}^n$ to $\{0,1\}^n$.

Let K_n denote the random variable that uniformly distributes over S_n.

Definition (uniform permutation ensemble)

The uniform permutation ensemble, denoted $\mathcal{K} = \{K_n\}_{n\in\mathbb{N}}$, has K_n uniformly distributed over the set of all permutations mapping n-bit-long strings to n-bit-long strings.

Pseudorandom Permutation

Definition (permutation ensemble)

A permutation ensemble is a sequence $\mathcal{P} = \{P_n\}_{n\in\mathbb{N}}$ of random variables such that the random variable P_n assumes values in the set of permutations mapping n-bit-long strings to n-bit-long strings.

Definition (Pseudorandom permutation ensemble)

A permutation ensemble $\mathcal{P} = \{P_n\}_{n\in\mathbb{N}}$ is called *pseudorandom* if for every probabilistic polynomial-time oracle machine M, every polynomial $p(\cdot)$, and all sufficiently large n's,

$$\left|\Pr\left(M^{P_n}(1^n)\right) - \Pr\left(M^{K_n}(1^n)\right)\right| < \frac{1}{p(n)},$$

where $\mathcal{K} = \{K_n\}_{n\in\mathbb{N}}$ is the uniform permutation ensemble.

Computational Indistinguishability

When we talk about the asymptotic behavior of an algorithm, we need that algorithm accepts any length of the input.

Many cryptographic definitions rely on the computational indistinguishability.

The formal definition of computational indistinguishability refers to probability ensembles, which are infinite sequences of probability distributions.

Pairing

Definition (Pairing)

A *pairing* is a map

$$e: G_1 \times G_2 \to G_3$$

satisfies the following two conditions:

- Bilinearity

$$e(P + P', Q) = e(P, Q)e(P', Q), \forall P, P' \in G_1, Q \in G_2$$
$$e(P, Q + Q') = e(P, Q)e(P, Q'), \forall P \in G_1, Q, Q' \in G_2$$

- Non-degeneracy

For all non-zero $P \in G_1$, $\exists Q \in G_2$ such that $e(P, Q) \neq 1$

For all non-zero $Q \in G_2$, $\exists P \in G_1$ such that $e(P, Q) \neq 1$

Decisional Diffie-Hellman (DDH) problem

Decisional Diffie-Hellman problem

Let G be a cyclic group. Given the generator P and (aP, bP, Q), try to decide whether $Q = abP$.

Pairings make the DDH problem easy.

Because

$$e(aP, bP) = e(P, P)^{ab} = e(P, abP).$$

It is easily to decide DDH by checking

$$e(aP, bP) \stackrel{?}{=} e(P, Q).$$

Outline

1. PoS-based Blockchains
   - overview to Bitcoin
   - Ouroboros Praos
   - Algorand
2. Two Crypto Notions
   - security parameter and asymptotic behavior
   - pairing
3. Experience in Industry