Blockstream

Blockstream RESEARCH

Cryptography Basics for Bitcoiners

Jonas Nick
nickler.ninja
@n1ckler

2023-05-30

Kyiv Bitcoin Meetup

Cryptography shifts the balance of power

What is needed is an electronic payment system based on cryptographic proof instead of trust [...].

- Satoshi


BITCOIN
BITCOIN COMMUNITY
imgflip.com

Blockstream

RESEARCH

Blockstream

RESEARCH

- Focus: Cryptography & formal languages

Blockstream
RESEARCH

- Focus: Cryptography & formal languages
  - scientific publications, specifications (BIPs), development of free open source software

Blockstream
RESEARCH

- Focus: Cryptography & formal languages
  - scientific publications, specifications (BIPs), development of free open source software
  - for Bitcoin protocol, wallets, Elements/Liquid Sidechain, Lightning Network, Federated E-cash, etc.

Blockstream

RESEARCH

- Focus: Cryptography & formal languages
  - scientific publications, specifications (BIPs), development of free open source software
  - for Bitcoin protocol, wallets, Elements/Liquid Sidechain, Lightning Network, Federated E-cash, etc.
- blog.blockstream.com
- twitter.com/blksresearch

Warm Up

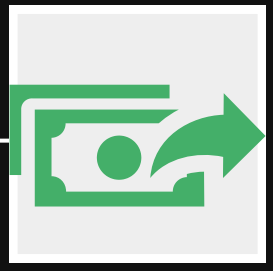
bitcoin.pdf

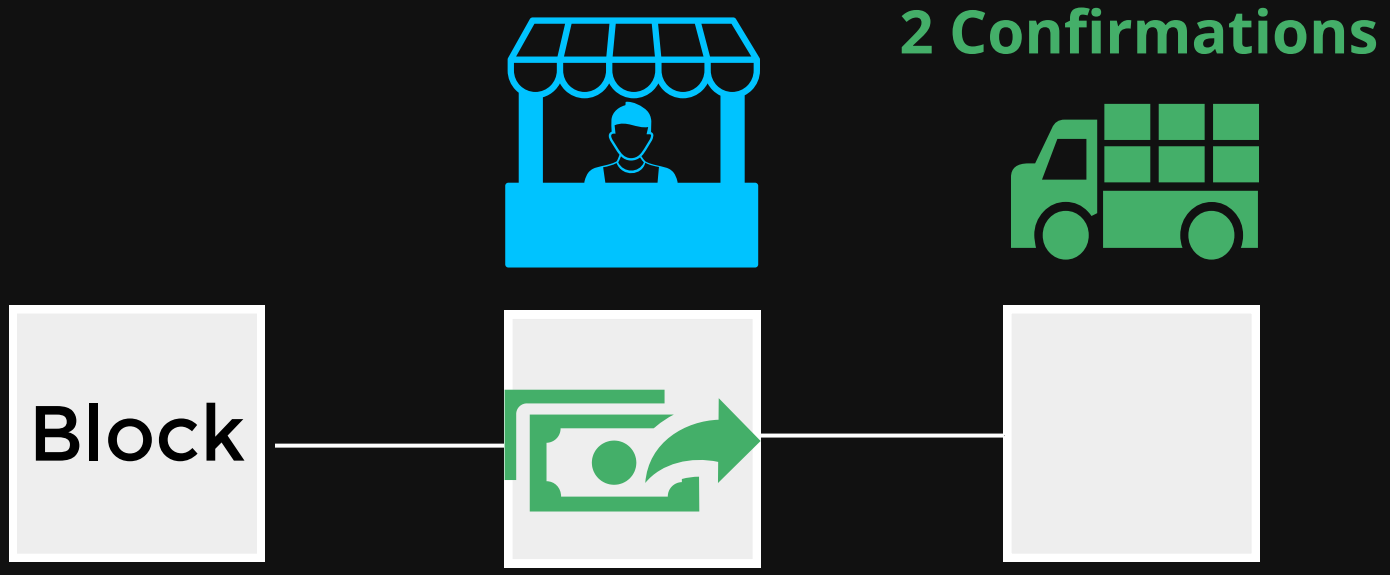
11. Calculations

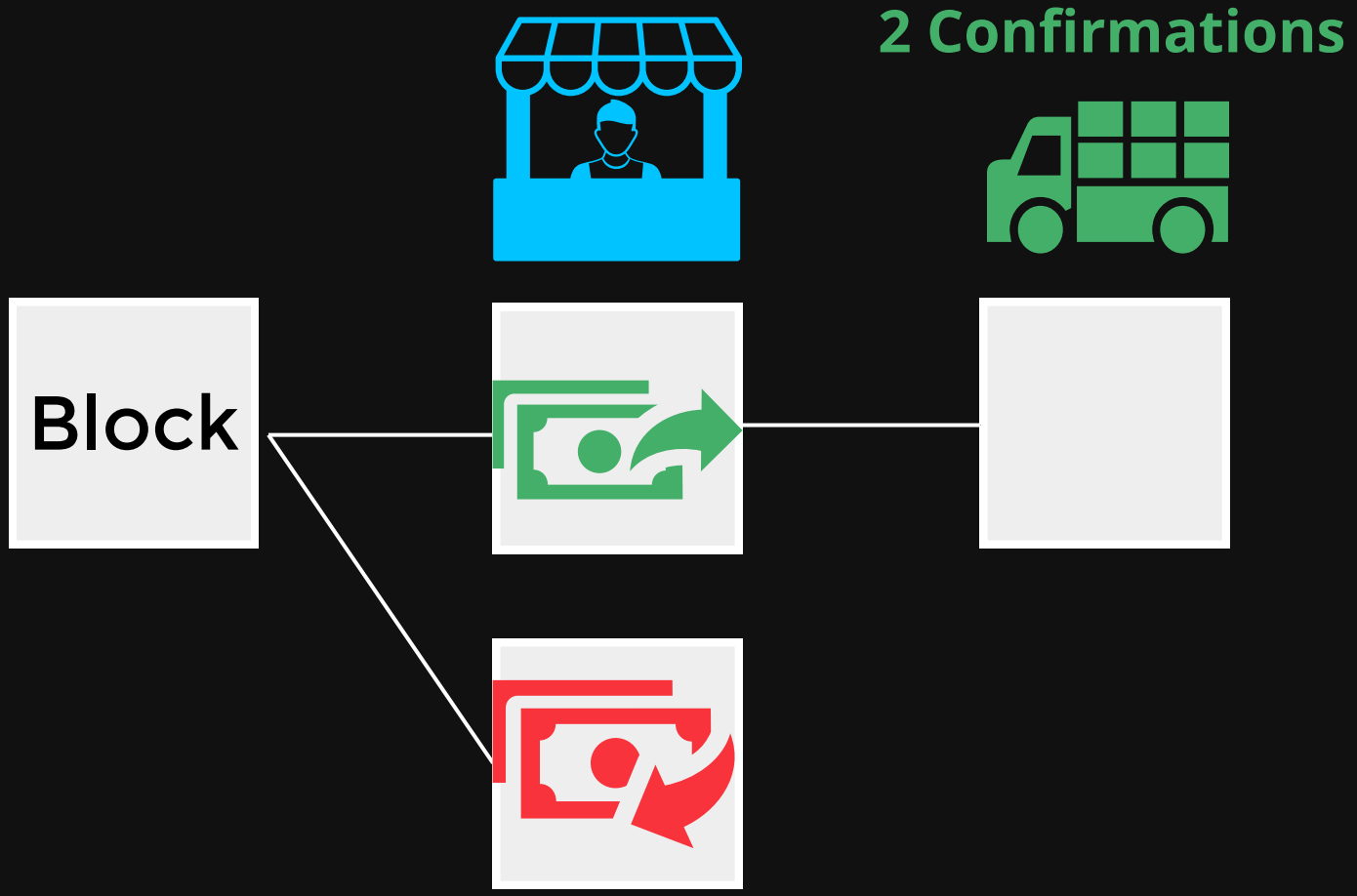
We consider the scenario of an attacker trying to generate an alternate chain faster than the honest chain. Even if this is accomplished, it does not throw the system open to arbitrary changes, such

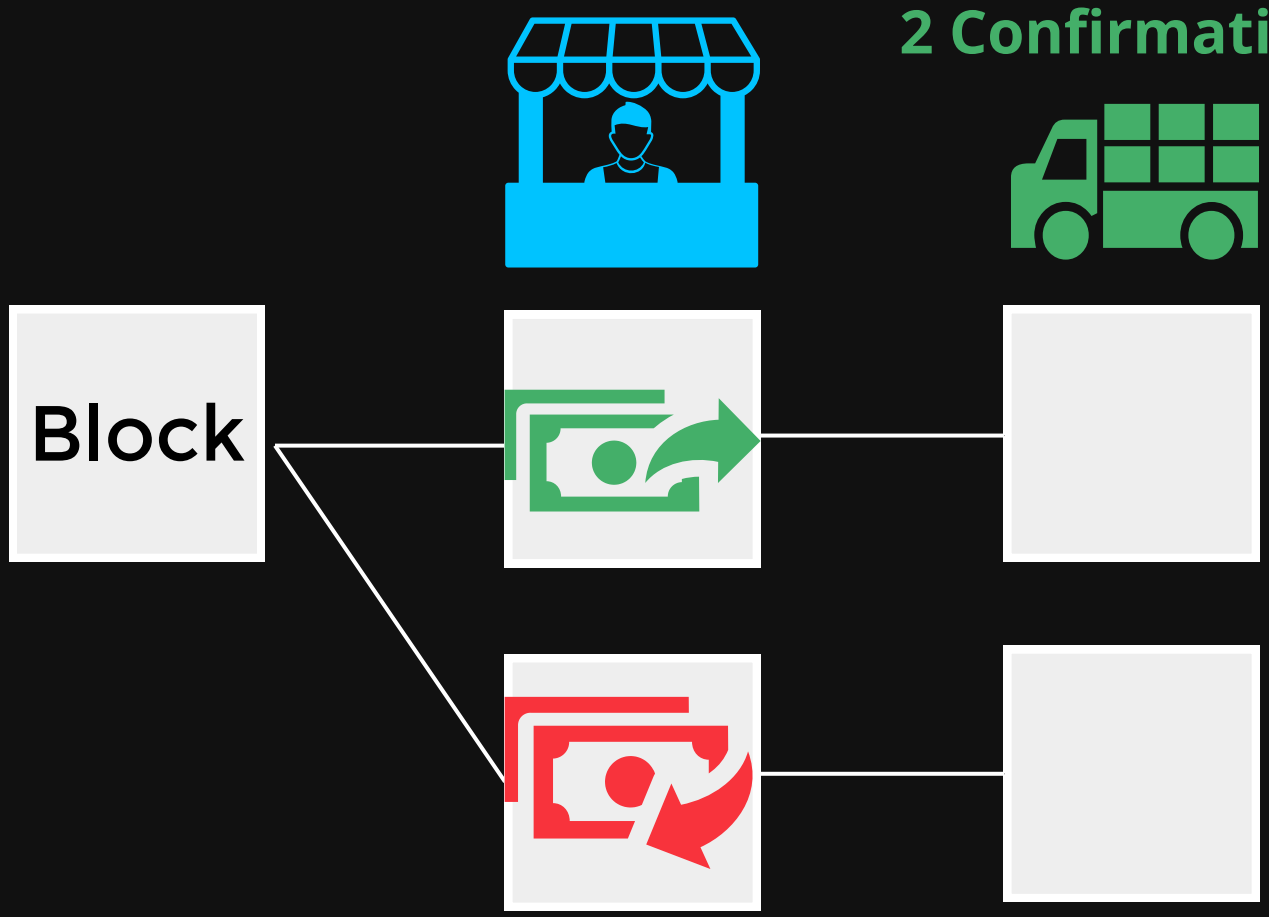
Block

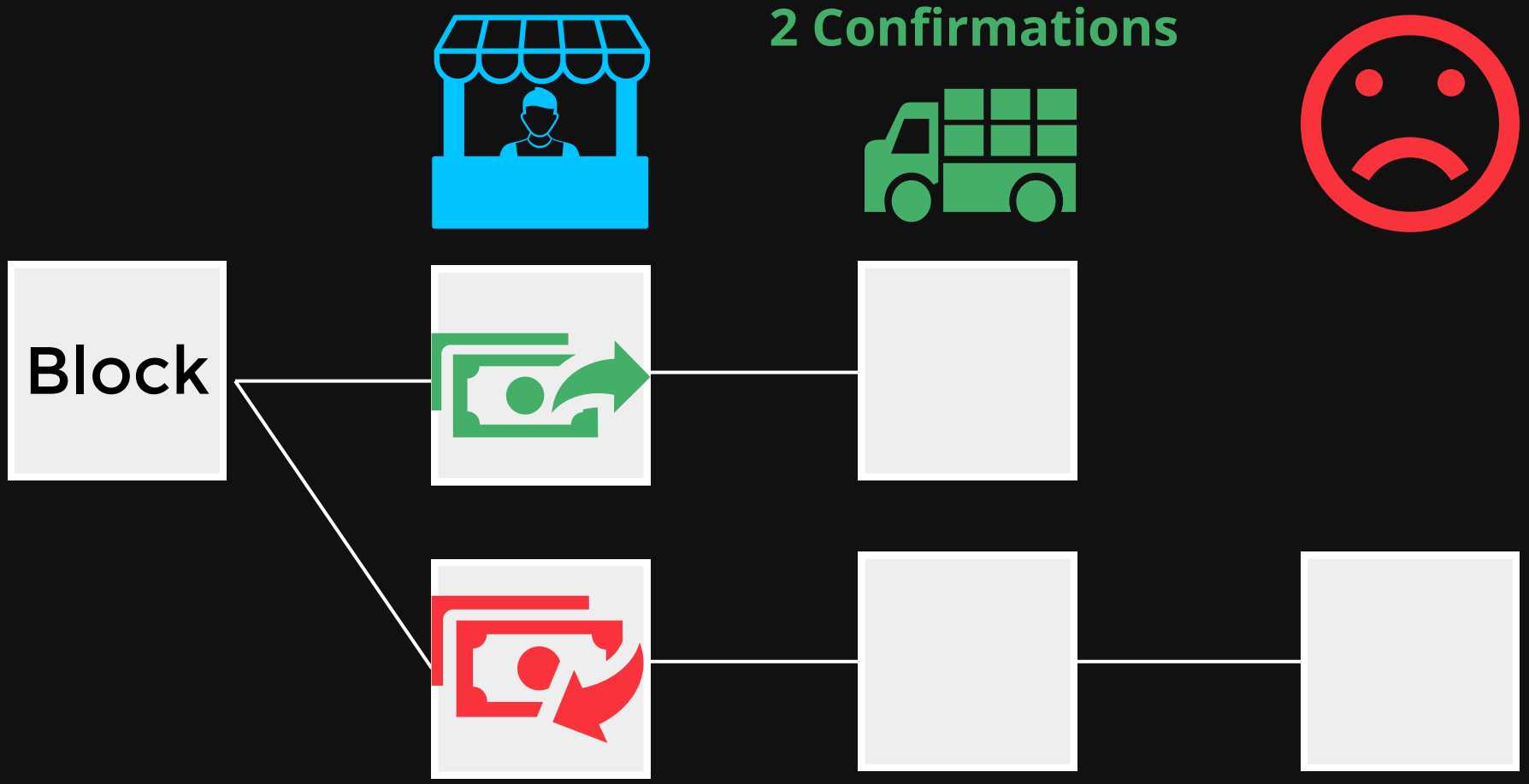
Block



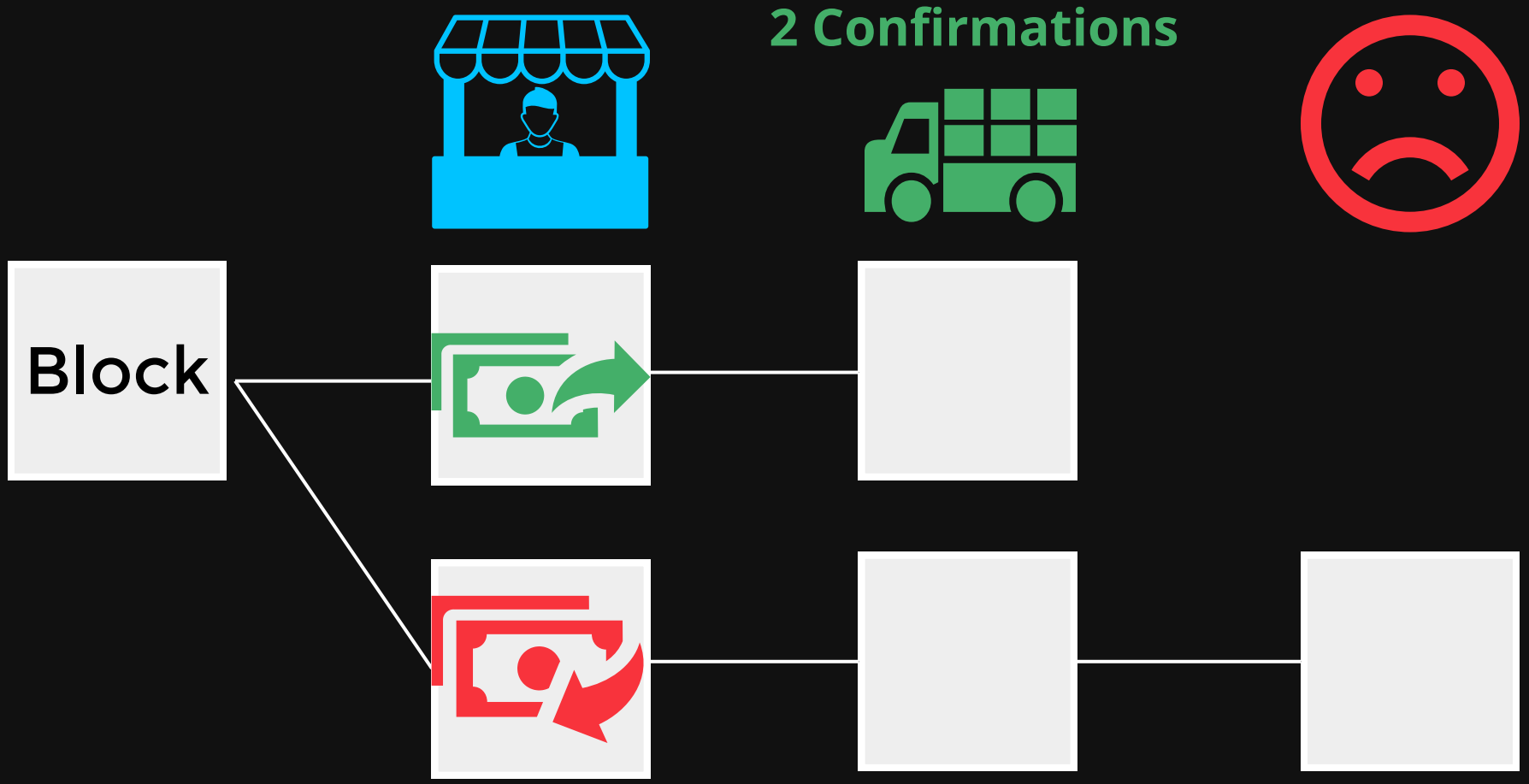

2 Confirmations
Block


2 Confirmations
Block


2 Confirmations
Block


2 Confirmations
Block

Double Spending



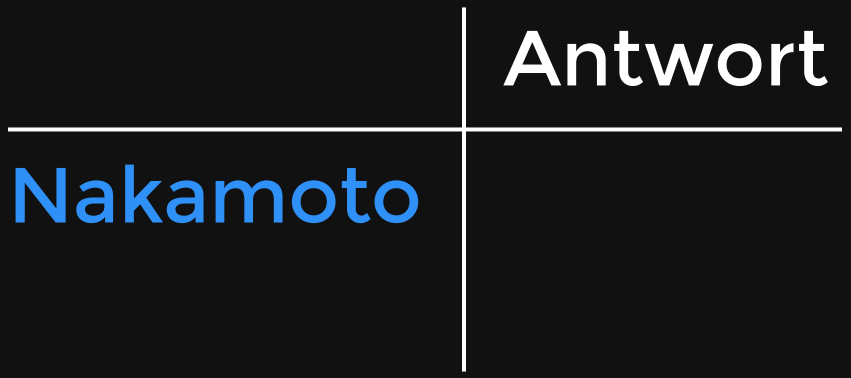
How many confirmations should the receiver of a transaction wait for?

How many confirmations should the receiver of a transaction wait for?

With 99% probability of success and
20% attacking hashrate

How many confirmations should the receiver of a transaction wait for?

With 99% probability of success and
20% attacking hashrate



How many confirmations should the receiver of a transaction wait for?

With 99% probability of success and
20% attacking hashrate

	Antwort
Nakamoto	7

How many confirmations should the receiver of a transaction wait for?

With 99% probability of success and
20% attacking hashrate

	Antwort
Nakamoto	7
Rosenfeld	8

How many confirmations should the receiver of a transaction wait for?

With 99% probability of success and
20% attacking hashrate

	Antwort
Nakamoto	7
Rosenfeld	8

The authors differ in their model of the attacker

Without understanding the model, the result is worthless.

Without understanding the model, the result is worthless.

For example, both models ignore whether attacking is a rational strategy in the first place.

What is Cryptography?

What is Cryptography?

the scientific study of techniques for securing digital information, transactions, and distributed computations.

From: Introduction to Modern Cryptography, J. Katz, Y. Lindell

What is Cryptography?

the scientific study of techniques for securing digital information, transactions, and distributed computations.

From: Introduction to Modern Cryptography, J. Katz, Y. Lindell

Definitions, models, assumptions and precise security proofs play a central role

What are Signatures?

What are Signatures?

In Bitcoin, digital signatures ensure that only the coin's owner can spend it.

What are Signatures?

In Bitcoin, digital signatures ensure that only the coin's owner can spend it.

private Key

public Key

What are Signatures?

In Bitcoin, digital signatures ensure that only the coin's owner can spend it.

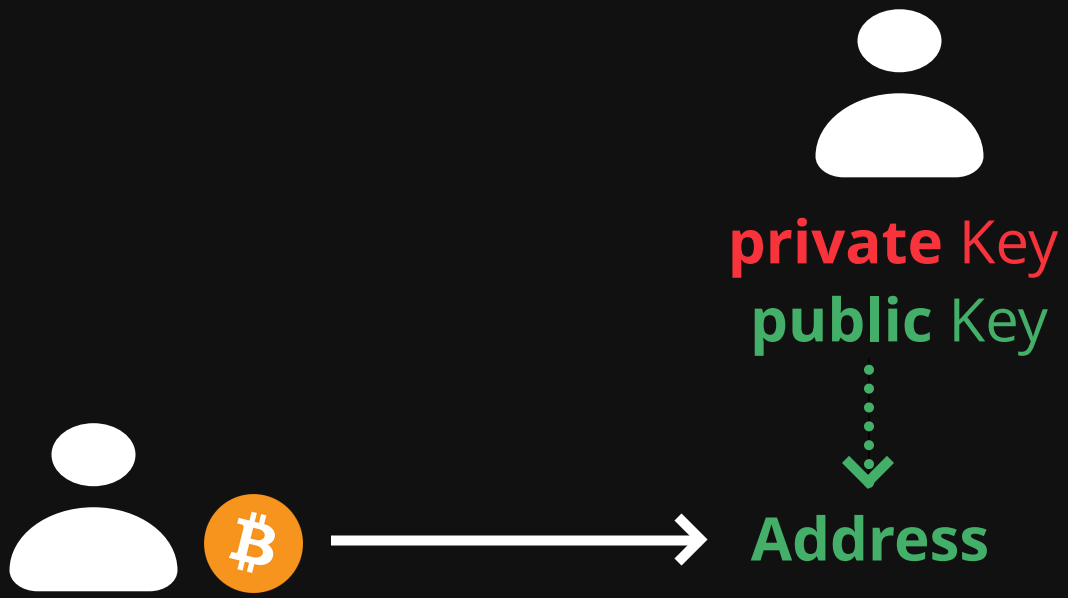
private Key

public Key

Address

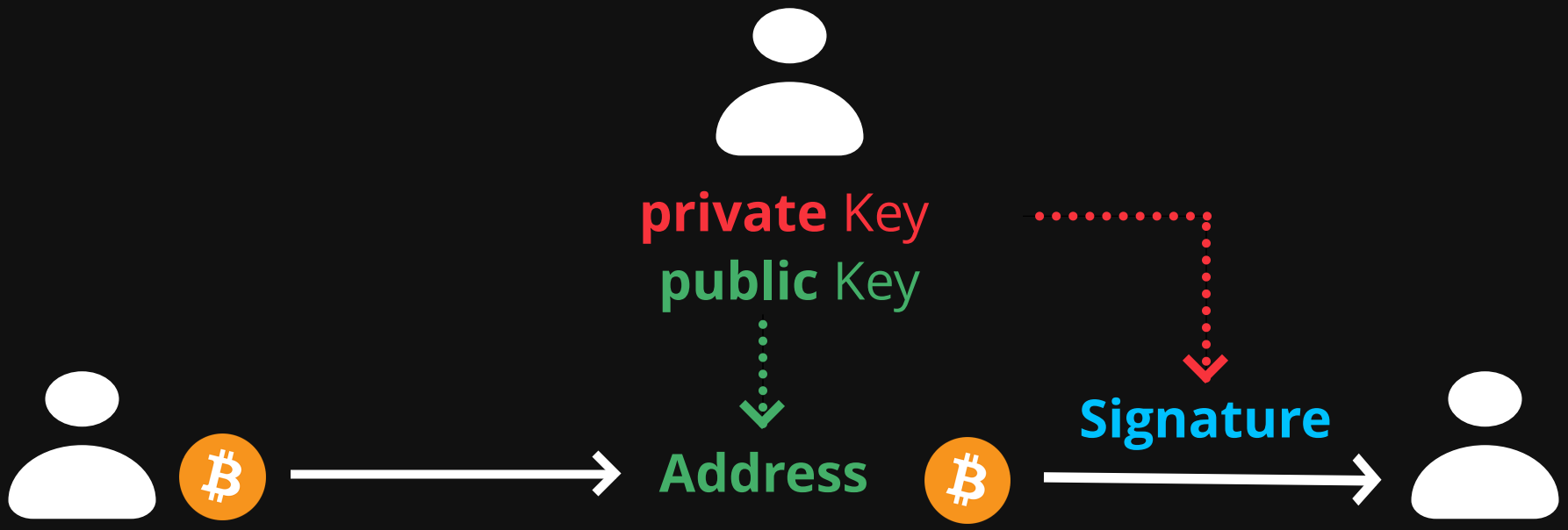
What are Signatures?

In Bitcoin, digital signatures ensure that only the coin's owner can spend it.



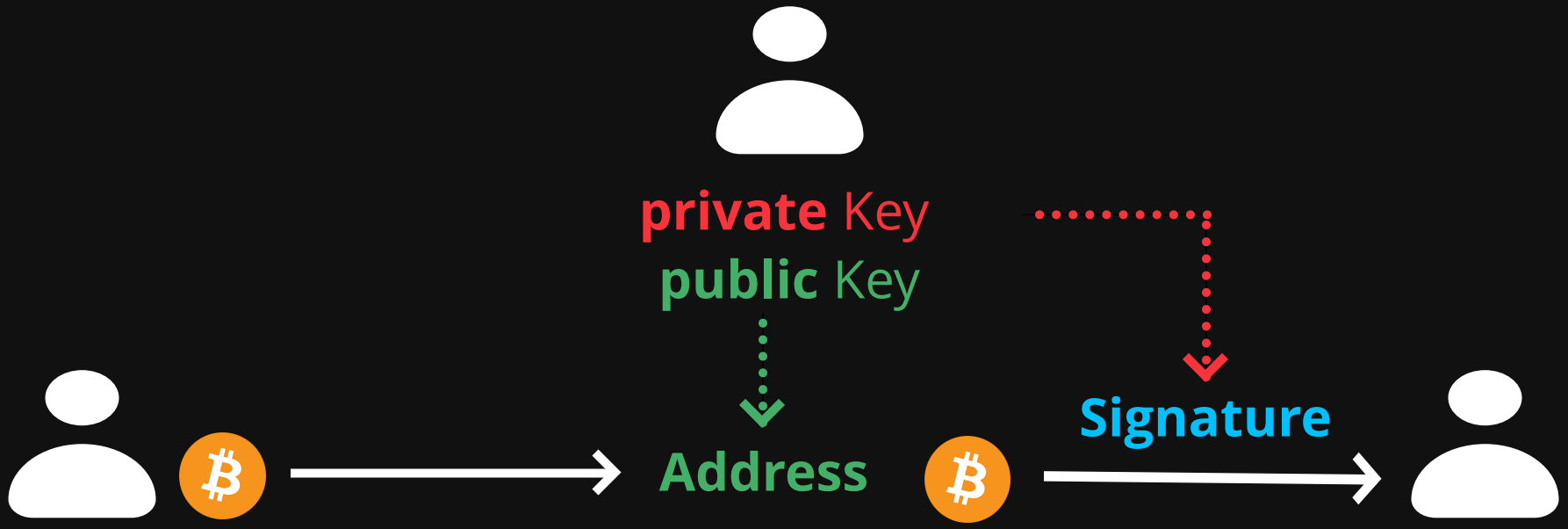
What are Signatures?

In Bitcoin, digital signatures ensure that only the coin's owner can spend it.



What are Signatures?

In Bitcoin, digital signatures ensure that only the coin's owner can spend it.



They consist of three parts...

What are Signatures?

Definition:

What are Signatures?

Definition:

- `KeyGen`
  - Output: key pair consisting of private and public key

What are Signatures?

Definition:

- `KeyGen`
  - Output: key pair consisting of private and public key
- `Sign`
  - Input: private key and message
  - Output: signature

What are Signatures?

Definition:

- `KeyGen`
  - Output: key pair consisting of private and public key
- `Sign`
  - Input: private key and message
  - Output: signature
- `Verify`
  - Input: public key, message, signature
  - Output: true or false

What are Signatures?

Definition:

- `KeyGen`
  - Output: key pair consisting of private and public key
- `Sign`
  - Input: private key and message
  - Output: signature
- `Verify`
  - Input: public key, message, signature
  - Output: true or false

Forgery: a valid signature for the public key of someone else

When are Signature Schemes secure?

To prove:

When are Signature Schemes secure?

To prove:

- It is extremely difficult to forge a signature

When are Signature Schemes secure?

To prove:

- ~~It is extremely difficult to forge a signature~~

When are Signature Schemes secure?

To prove:

- ~~It is extremely difficult to forge a signature~~
- It takes a very, very long time to forge a signature

When are Signature Schemes secure?

To prove:

- ~~It is extremely difficult to forge a signature~~
- ~~It takes a very, very long time to forge a signature~~

When are Signature Schemes secure?

To prove:

- ~~It is extremely difficult to forge a signature~~
- ~~It takes a very, very long time to forge a signature~~
- Assumption: there is a "problem X", that probably takes very, very long to solve

When are Signature Schemes secure?

To prove:

- ~~It is extremely difficult to forge a signature~~
- ~~It takes a very, very long time to forge a signature~~
- Assumption: there is a "problem X", that probably takes very, very long to solve
  - Forging a signature is at least as hard as solving problem X

When are Signature Schemes secure?

To prove:

- ~~It is extremely difficult to forge a signature~~
- ~~It takes a very, very long time to forge a signature~~
- Assumption: there is a "problem X", that probably takes very, very long to solve
  - Forging a signature is at least as hard as solving problem X
  - Or the other way around: if problem X is hard, then the signature scheme is secure

The Security Proof

The Security Proof

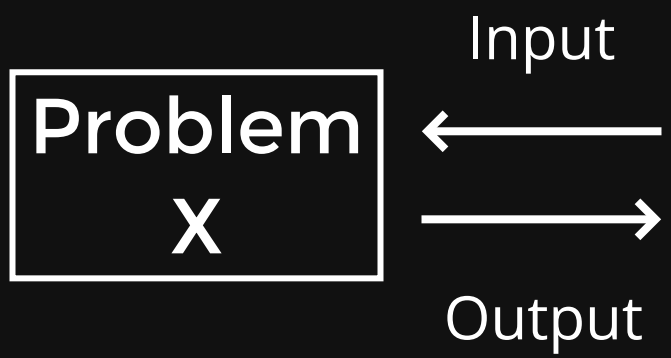
Problem X

The Security Proof

Problem X

(Discrete logarithm on elliptic curve secp256k1)

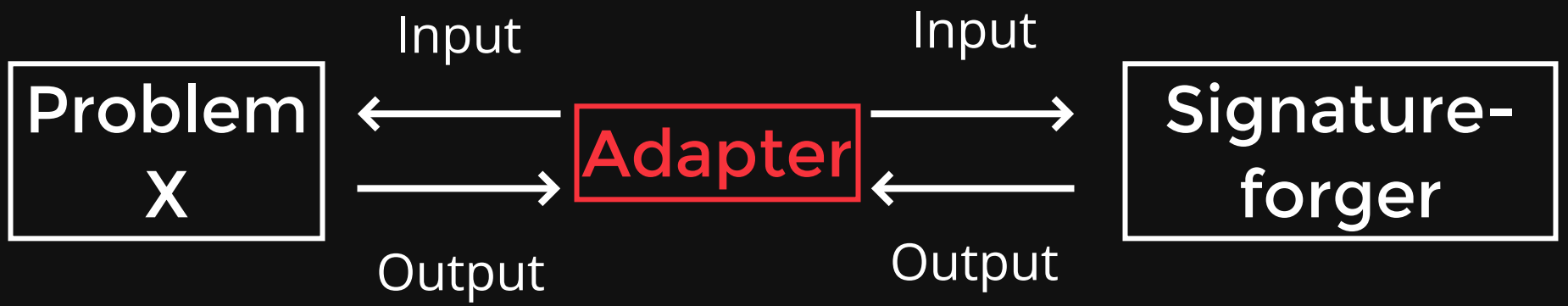
The Security Proof



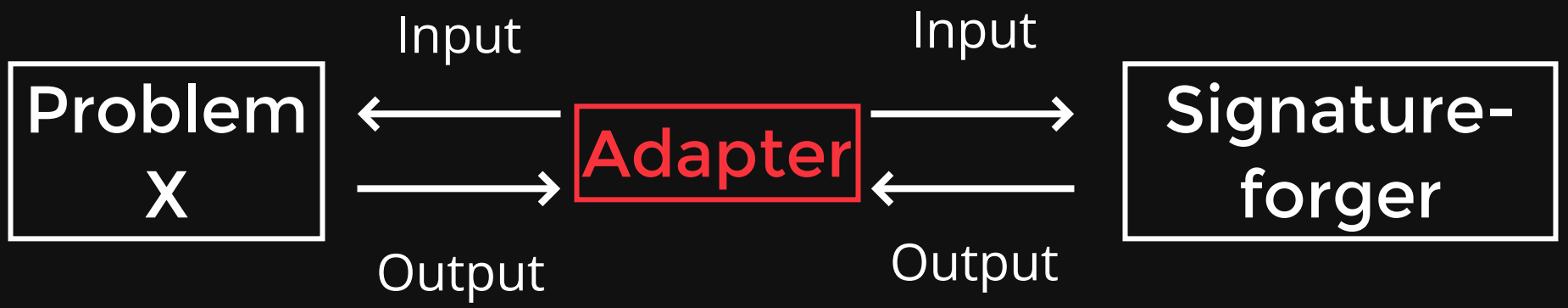
The Security Proof

The Security Proof

The Security Proof



The Security Proof



If there is a forger, we can solve problem X

Time to catch our breath...

No proof, no problem?

No proof, no problem?

- Example:

No proof, no problem?

- Example:
  - **Problem X** (Dlog on secp256k1) is considered hard, because it is not known how to solve it efficiently

No proof, no problem?

- Example:
  - **Problem X** (Dlog on secp256k1) is considered hard, because it is not known how to solve it efficiently
  - Hash function **SHA-256** does not have a complete security proof

No proof, no problem?

- Example:
  - **Problem X** (Dlog on secp256k1) is considered hard, because it is not known how to solve it efficiently
  - Hash function **SHA-256** does not have a complete security proof
  - The **ECDSA** signature scheme has a security proof, but in an unusual model

No proof, no problem?

- Example:
  - **Problem X** (Dlog on secp256k1) is considered hard, because it is not known how to solve it efficiently
  - Hash function **SHA-256** does not have a complete security proof
  - The **ECDSA** signature scheme has a security proof, but in an unusual model
- But, new schemes without proof: should be sceptical!

No proof, no problem?

- Example:
  - **Problem X** (Dlog on secp256k1) is considered hard, because it is not known how to solve it efficiently
  - Hash function **SHA-256** does not have a complete security proof
  - The **ECDSA** signature scheme has a security proof, but in an unusual model
- But, new schemes without proof: should be sceptical!
  - First versions o**f half-aggregation**, **discreet-log contracts**, **pay-to-contract** outright insecure

No proof, no problem?

- Example:
  - **Problem X** (Dlog on secp256k1) is considered hard, because it is not known how to solve it efficiently
  - Hash function **SHA-256** does not have a complete security proof
  - The **ECDSA** signature scheme has a security proof, but in an unusual model
- But, new schemes without proof: should be sceptical!
  - First versions o**f half-aggregation**, **discreet-log contracts**, **pay-to-contract** outright insecure
  - **BIP 32** (HD-Wallets) more complicated than necessary

Back to the real world

Back to the real world

Paper

Back to the real world

Paper

Bsp.: "Let $P \in \mathbb{G}$"

Back to the real world

Paper → Implementation

Bsp.: "Let $P \in \mathbb{G}$"

```
unsigned char
P[33];
```

Back to the real world

	Paper	Specification	Implementation
Bsp.:	"Let $P \in \mathbb{G}$"	"Let P be an array of 33 bytes"	`unsigned char P[33];`

Back to the real world

	Paper	Specification	Implementation
Bsp.:	"Let $P \in \mathbb{G}$"	"Let P be an array of 33 bytes"	`unsigned char P[33];`

Specification / Bitcoin Improvement Proposal (BIP):

Back to the real world

	Paper	Specification	Implementation
Bsp.:	"Let $P \in \mathbb{G}$"	"Let P be an array of 33 bytes"	`unsigned char P[33];`

Specification / Bitcoin Improvement Proposal (BIP):

- Fram mathematical objects to bits & bytes

Back to the real world

	Paper	Specification	Implementation
Bsp.:	"Let $P \in \mathbb{G}$"	"Let P be an array of 33 bytes"	`unsigned char P[33];`

Specification / Bitcoin Improvement Proposal (BIP):

- Fram mathematical objects to bits & bytes
- Goal: allows for compatible implementations

Back to the real world

	Paper	Specification	Implementation
Bsp.:	"Let $P \in \mathbb{G}$"	"Let P be an array of 33 bytes"	`unsigned char P[33];`

Specification / Bitcoin Improvement Proposal (BIP):

- Fram mathematical objects to bits & bytes
- Goal: allows for compatible implementations
- Specification of specifications: BIP 2

Back to the real world

	Paper	Specification	Implementation
Bsp.:	"Let $P \in \mathbb{G}$"	"Let P be an array of 33 bytes"	`unsigned char P[33];`

Specification / Bitcoin Improvement Proposal (BIP):

- Fram mathematical objects to bits & bytes
- Goal: allows for compatible implementations
- Specification of specifications: BIP 2
- Unclear specifications lead to vulnerabilities in implementations

Back to the real world

	Paper	Specification	Implementation
Bsp.:	"Let $P \in \mathbb{G}$"	"Let P be an array of 33 bytes"	`unsigned char P[33];`

Specification / Bitcoin Improvement Proposal (BIP):

- Fram mathematical objects to bits & bytes
- Goal: allows for compatible implementations
- Specification of specifications: BIP 2
- Unclear specifications lead to vulnerabilities in implementations
- In the future ideally: formal specifications, that allow provably correct implementations

Back to the real world

	Paper	Specification	Implementation
Bsp.:	"Let $P \in \mathbb{G}$"	"Let P be an array of 33 bytes"	`unsigned char P[33];`

Back to the real world

Bsp.:

Paper	Specification	Implementation
"Let $P \in \mathbb{G}$"	"Let P be an array of 33 bytes"	`unsigned char P[33];`

Implementation:

Back to the real world

	Paper	Specification	Implementation
Bsp.:	"Let $P \in \mathbb{G}$"	"Let P be an array of 33 bytes"	`unsigned char P[33];`

Implementation:

- Should obviously be correct

Back to the real world

	Paper	Specification	Implementation
Bsp.:	"Let $P \in \mathbb{G}$"	"Let P be an array of 33 bytes"	`unsigned char P[33];`

Implementation:

- Should obviously be correct
- Free of "side-channels", e.g., correlation of private key and computation time

Back to the real world

Paper → Specification → Implementation

	Paper	Specification	Implementation
Bsp.:	"Let $P \in \mathbb{G}$"	"Let P be an array of 33 bytes"	`unsigned char P[33];`

Implementation:

- Should obviously be correct
- Free of "side-channels", e.g., correlation of private key and computation time

Hello,

We'd like to announce the release of version 0.3.2 of libsecp256k1:

https://github.com/bitcoin-core/secp256k1/releases/tag/v0.3.2

This is a bugfix release after 0.3.1. The impetus for this release is the discovery that GCC 13 became smart enough to optimize out a specific timing side-channel protection mechanism in the ECDH code that could leave applications vulnerable to a side-channel attack. This has been fixed in 0.3.2 [1].

What is Bitcoin Cryptography?

on-chain

(Base Layer)

off-chain

("Layer 2")

---

on-chain

(Base Layer)

off-chain

("Layer 2")

---

on-chain

(Base Layer)

validated by every Bitcoin node, consensus

off-chain ("Layer 2")	optional, settlement on base layer
on-chain (Base Layer)	validated by every Bitcoin node, consensus

(Multiparty-)
Payment Channels

Sidechains

Federated
E-Cash

off-chain

("Layer 2")

optional, settlement on base layer

---

on-chain

(Base Layer)

validated by every Bitcoin node,
consensus

Cryptography

Cryptography

Bitcoin Cryptography

Cryptography

Bitcoin Cryptography

Requires consensus from the Bitcoin community, therefore

Cryptography

Bitcoin Cryptography

Requires consensus from the Bitcoin community, therefore

- established assumptions

Cryptography

Bitcoin Cryptography

Requires consensus from the Bitcoin community, therefore

- established assumptions
- efficient

Cryptography

Bitcoin Cryptography

Requires consensus from the Bitcoin community, therefore

- established assumptions
- efficient
- "simple" to analyze and implement

Cryptography

Bitcoin Cryptography

Consensus is fluid

Requires consensus from the Bitcoin community, therefore

- established assumptions
- efficient
- "simple" to analyze and implement

Cryptography

Bitcoin Cryptography

Cryptography

Bitcoin Cryptography

Layer 2 Cryptography

Cryptography, which

Cryptography

Bitcoin Cryptography

Layer 2 Cryptography

Cryptography, which

- builds on Bitcoin's base layer

Cryptography

Bitcoin Cryptography

Layer 2 Cryptography

Cryptography, which

- builds on Bitcoin's base layer
- but would not be practical there

Cryptography

Bitcoin Cryptography

Layer 2 Cryptography

Cryptography, which

- builds on Bitcoin's base layer
- but would not be practical there
- allows better efficiency and privacy

Cryptography

Bitcoin Cryptography

Layer 2 Cryptography

Cryptography, which

- builds on Bitcoin's base layer
- but would not be practical there
- allows better efficiency and privacy
- includes, for example, multisignatures, blind signatures, zero-knowledge proofs

Case study: Post-Quantum Cryptography

Case study: Post-Quantum Cryptography

For quantum computers, problem X is theoretically not difficult. However, in practice, the threat situation is questionable.

Case study: Post-Quantum Cryptography

For quantum computers, problem X is theoretically not difficult. However, in practice, the threat situation is questionable.

Post-Quantum Crypto...

Case study: Post-Quantum Cryptography

For quantum computers, problem X is theoretically not difficult. However, in practice, the threat situation is questionable.

Post-Quantum Crypto...

- ... requires novel assumptions

Case study: Post-Quantum Cryptography

For quantum computers, problem X is theoretically not

CRYPTOGRAPHY

‘Post-Quantum’ Cryptography Scheme Is Cracked on a Laptop

6 |

Two researchers have broken an encryption protocol that many saw as a promising defense against the power of quantum computing.

Case study: Post-Quantum Cryptography

For quantum computers, problem X is theoretically not difficult. However, in practice, the threat situation is questionable.

Post-Quantum Crypto...

- ... requires novel assumptions
- ... has low efficiency

Case study: Post-Quantum Cryptography

For quantum computers, problem X is theoretically not difficult. However, in practice, the threat situation is questionable.

Post-Quantum Crypto...

- ... requires novel assumptions
- ... has low efficiency
- ... often has high complexity

Case study: Post-Quantum Cryptography

For quantum computers, problem X is theoretically not difficult. However, in practice, the threat situation is questionable.

Post-Quantum Crypto...

- ... requires novel assumptions
- ... has low efficiency
- ... often has high complexity

"Some people seem to be unable to rationally consider the possibility that NSA is sabotaging post-quantum cryptography."


ONE DOES NOT SIMPLY
MAKE BITCOIN QUANTUM-SECURE
imgflip.com

Outlook

Outlook

- Bitcoin Cryptography
  - Further progress on resilience (e.g., security proofs, better specifications, secure code)

Outlook

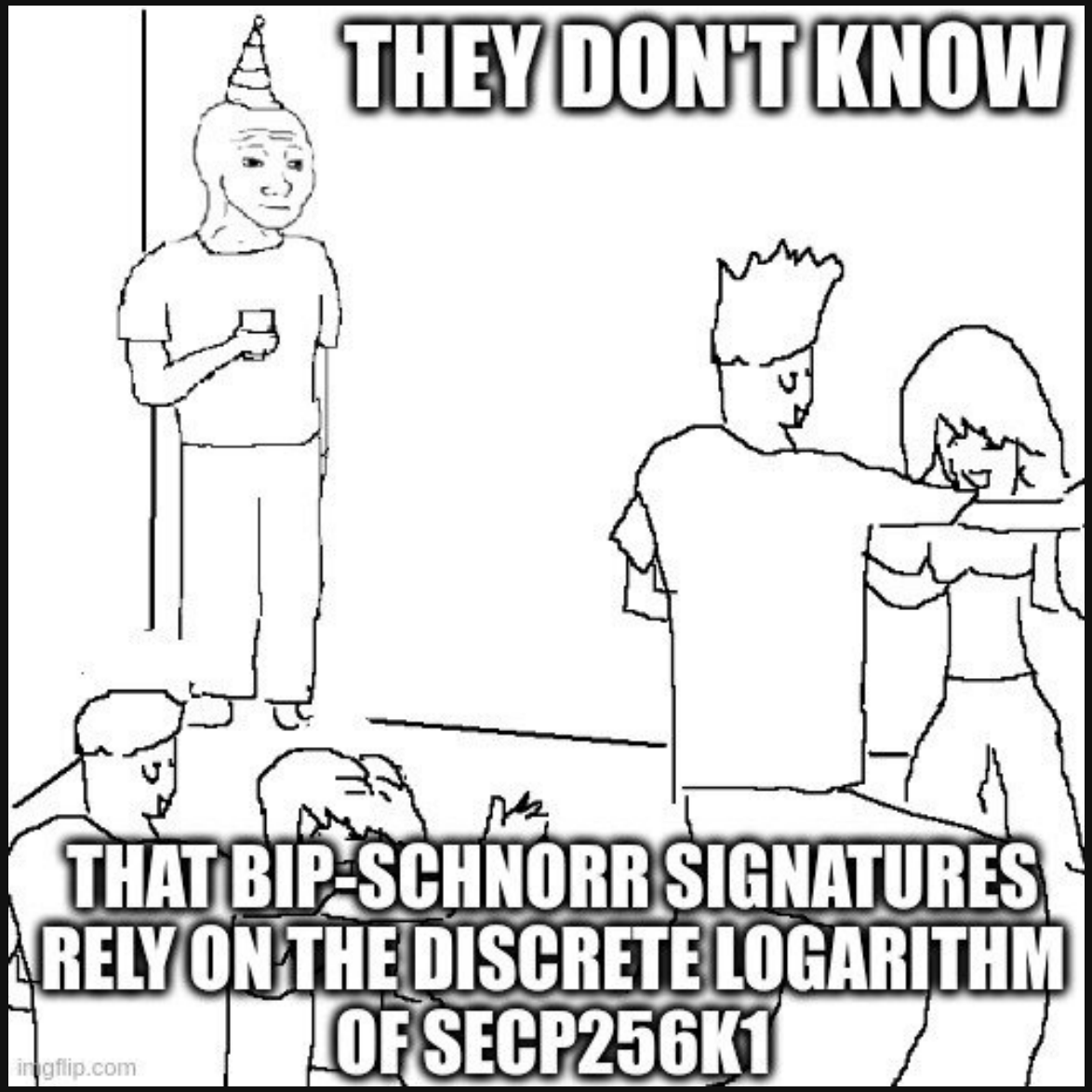
- Bitcoin Cryptography
  - Further progress on resilience (e.g., security proofs, better specifications, secure code)
  - Even with limitations based on current assumptions, significant improvements to the Bitcoin protocol are possible

Outlook

- Bitcoin Cryptography
  - Further progress on resilience (e.g., security proofs, better specifications, secure code)
  - Even with limitations based on current assumptions, significant improvements to the Bitcoin protocol are possible
  - What is considered secure changes over time and, therefore, so does what is consensus-compatible.

Outlook

- Bitcoin Cryptography
  - Further progress on resilience (e.g., security proofs, better specifications, secure code)
  - Even with limitations based on current assumptions, significant improvements to the Bitcoin protocol are possible
  - What is considered secure changes over time and, therefore, so does what is consensus-compatible.
- Layer 2 Cryptography:
  - A vast field with many open questions for theory and practice



Slides: nickler.ninja/slides