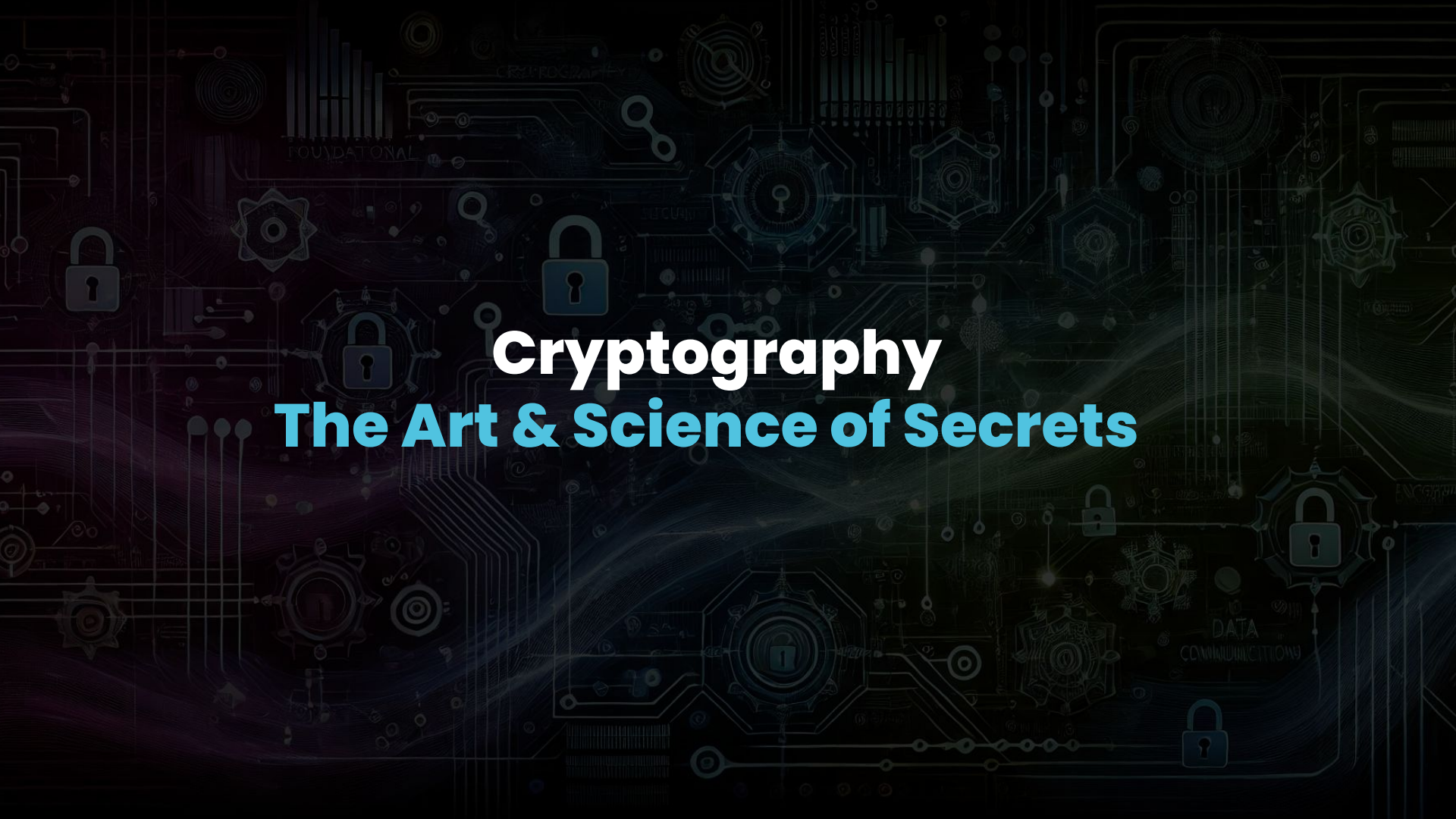
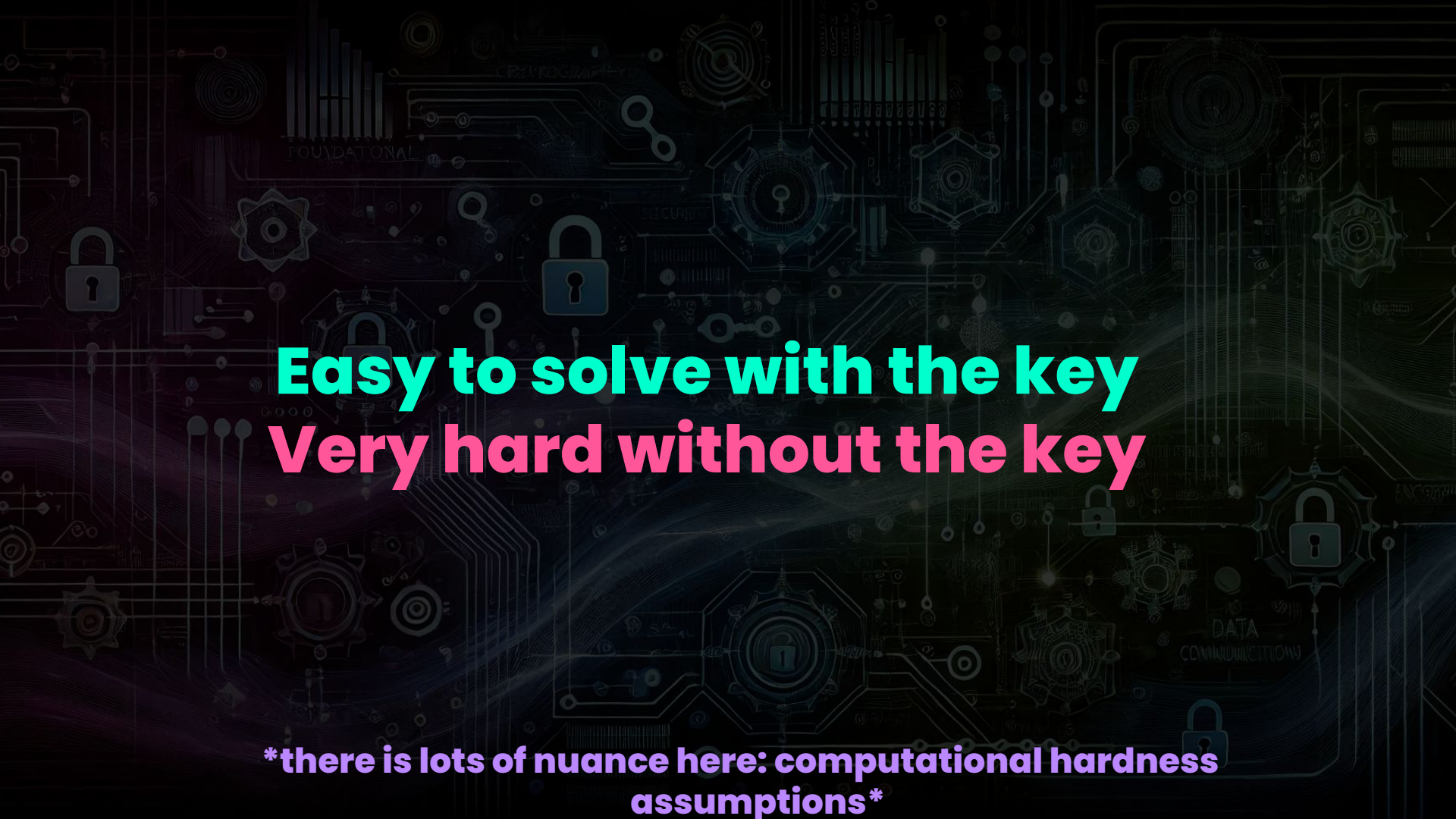
AtHeartEngineer & Ying Tong

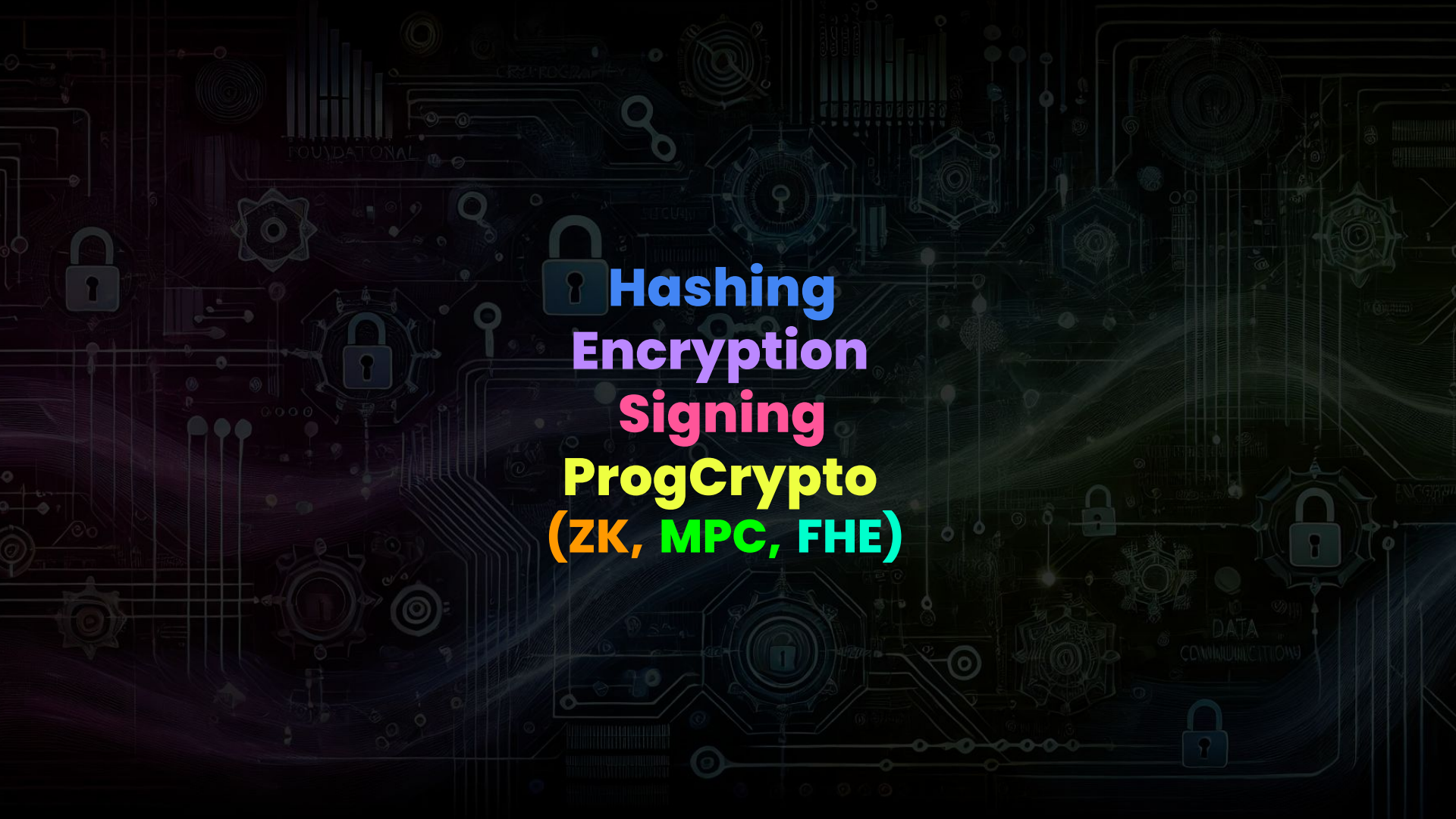
An Introduction to Cryptography


Cryptography
The Art & Science of Secrets

Information Security Triad

- **Confidentiality** : only the intended recipient can read the message
- **Integrity** : the message was not tampered with
- **Authenticity** : the message was sent from the intended sender


Easy to solve with the key
Very hard without the key
there is lots of nuance here: computational hardness assumptions

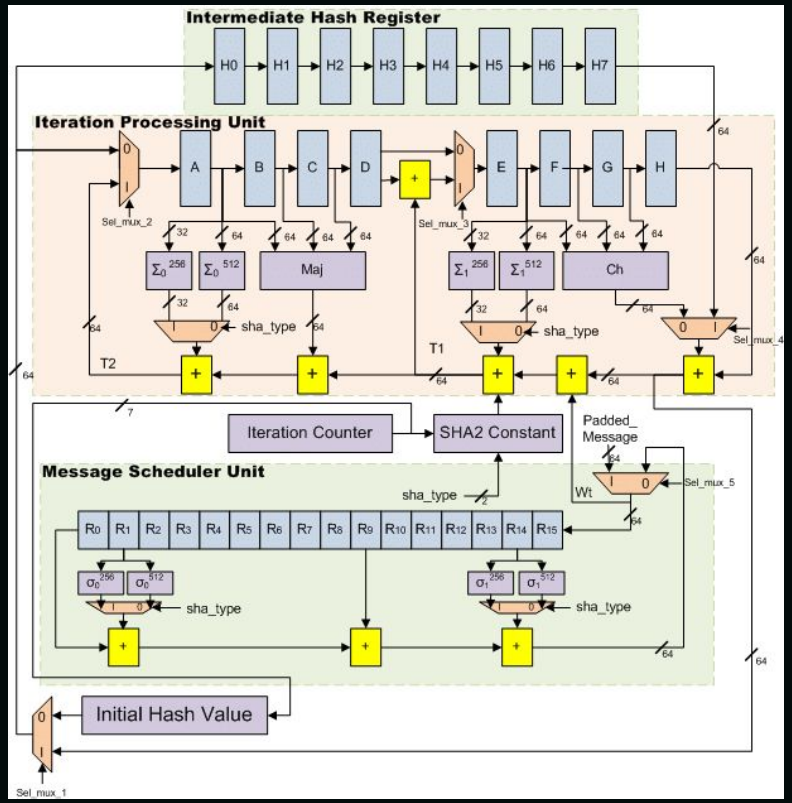

Hashing
Encryption
Signing
ProgCrypto
(ZK, MPC, FHE)


ONE WAY

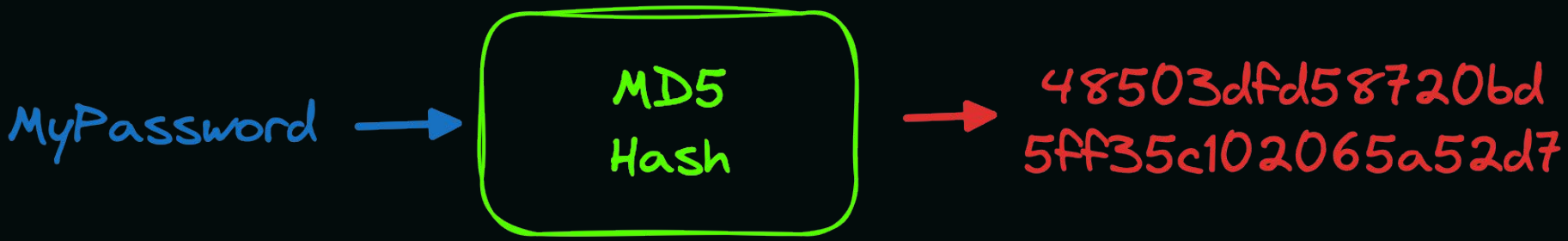

เดินรถทางเดียว

Hashing ≈ Fingerprint

Hashing


Intermediate Hash Register
H0
H1
H2
H3
H4
H5
H6
H7
Iteration Processing Unit
A
B
C
D
E
F
G
H
Sel_mux_2
Sel_mux_3
Maj
Ch
sha_type
T1
T2
Sel_mux_4
Iteration Counter
SHA2 Constant
Padded_ Message
Message Scheduler Unit
Wt
Sel_mux_5
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
R14
R15
Initial Hash Value
Sel_mux_1

Hashing ≈ Fingerprint

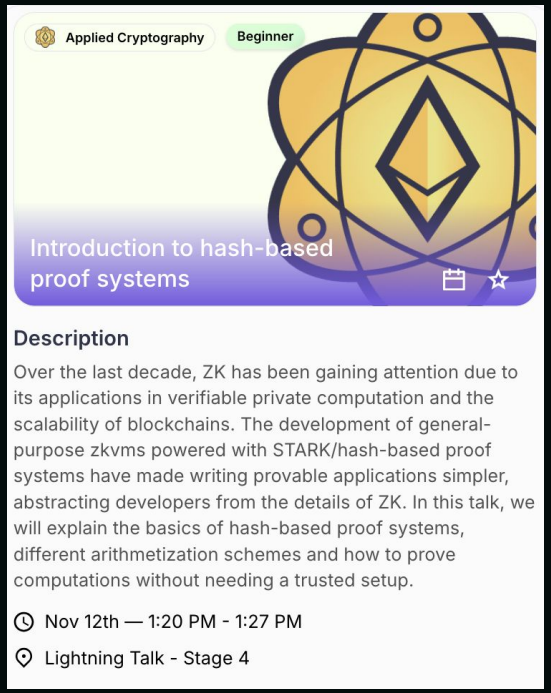


Hashing

- One Way Functions
  - *Blending fruits into a smoothie—you can't get the fruits back.*
- Unique input → unique output
- Fixed Length
- Examples: MD5, SHA2, BLAKE3, Keccak

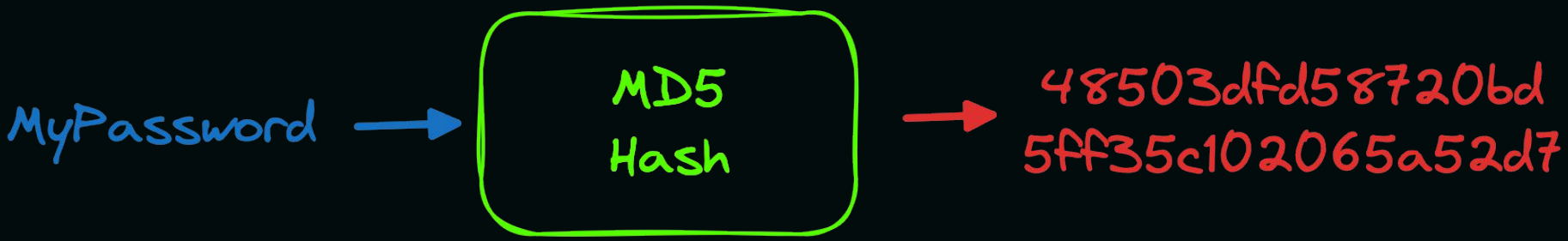
`Hi`	`3639efcd08abb273b1619e82e78c29a7df02c1051b1820e99fc395dcaa3326b8`
`Hell`	`226ff6eccefcf9d992c104354c9d42b32b02979055028620835ad7a3525ba240`
`Help`	`b79cac926e0b2e347e72cc91d5174037c9e17ae7733fd7bdb570f71b10cd7bfc`
`Hello`	`185f8db32271fe25f561a6fc938b2e264306ec304eda518007d1764826381969`

Hashing @ devcon vii



Passwords

Username	Password		Password Hash
Mulder	MyPassword		48503dfd58720bd5ff35c102065a52d7
Scully	MyPassword		48503dfd58720bd5ff35c102065a52d7



Salted Passwords

Username	Password	Salt	Salted Password Hash
Mulder	MyPassword		48503dfd58720bd5ff35c102065a52d7
Scully	MyPassword		48503dfd58720bd5ff35c102065a52d7
Adam	MyPassword	4c9d42b32b029790550	
Jamie	MyPassword	b2e347b029e72cc91d5	

Salted Passwords

Username	Password	Salt	Salted Password Hash
Mulder	MyPassword		48503dfd58720bd5ff35c102065a52d7
Scully	MyPassword		48503dfd58720bd5ff35c102065a52d7
Adam	MyPassword	4c9d42b32b029790550	a4736f4a5db98d242b6d66dc4357b54a
Jamie	MyPassword	b2e347b029e72cc91d5	9bed1fe4d0afb63171394404663cc0b5

bcrypt (hashing) with per-user salts

- **Facebook**
- **Twitter**
- **Google**
- **Dropbox**
- **LinkedIn**
- **Netflix**
- etc

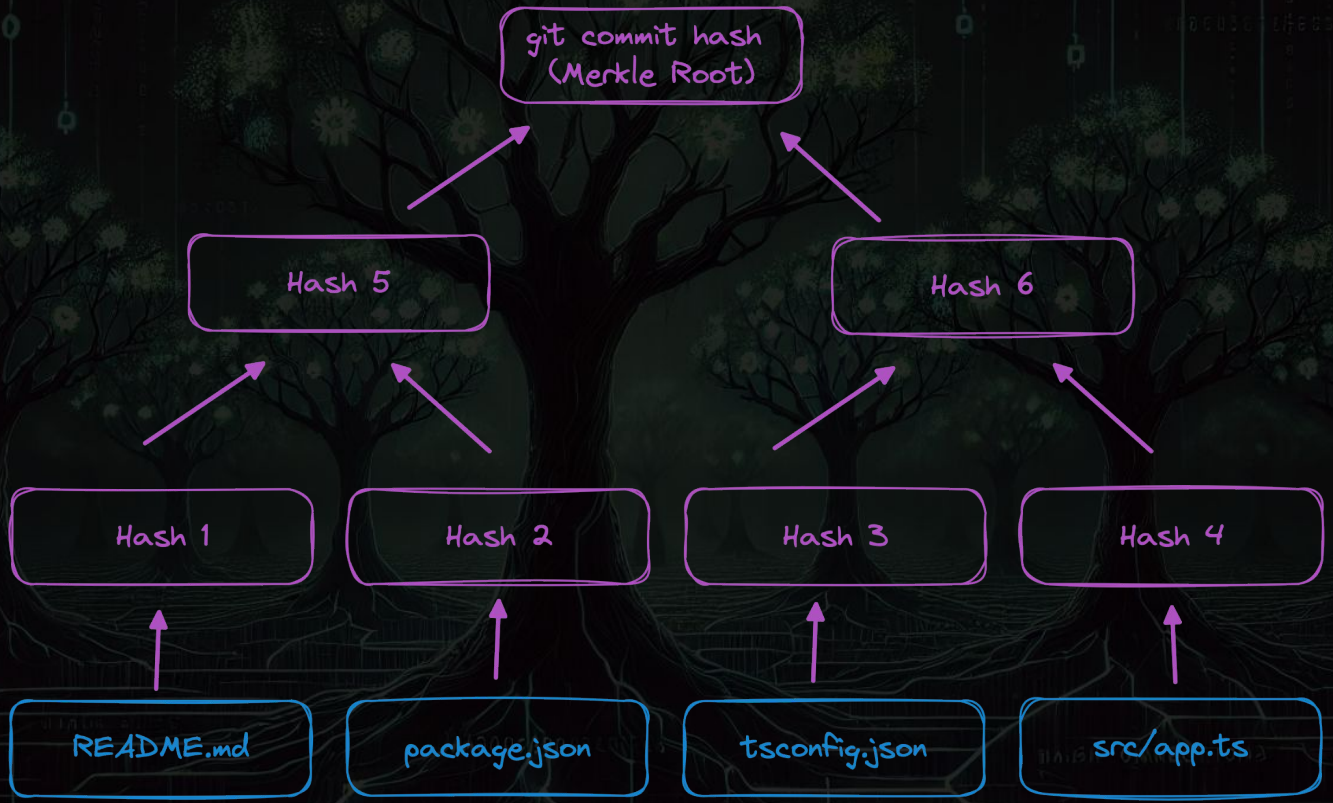
Cryptographic hash function

Preimage resistance: Given $H(x)$, it's hard to find x

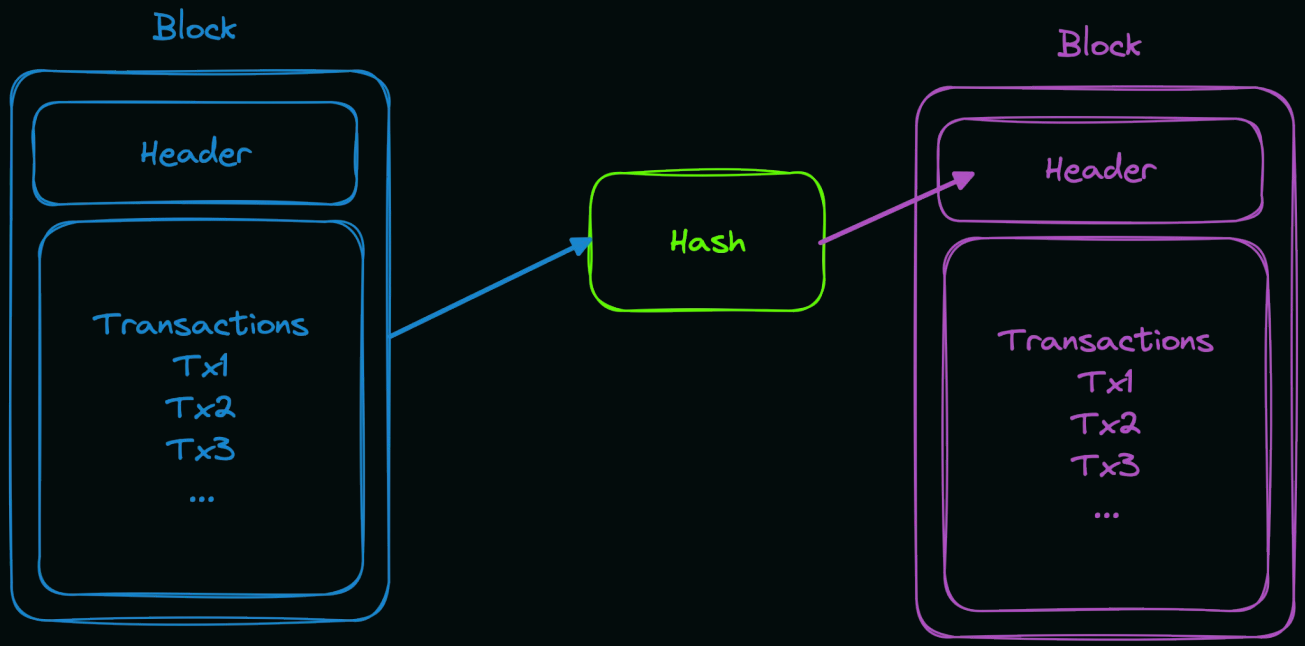
Second preimage resistance: Given m_1, it's hard to find another m_1 such that $H(m_1)=H(m_2)$

Collision resistance: It's hard to find any two distinct messages m_1, m_2 such that $H(m_1)=H(m_2)$

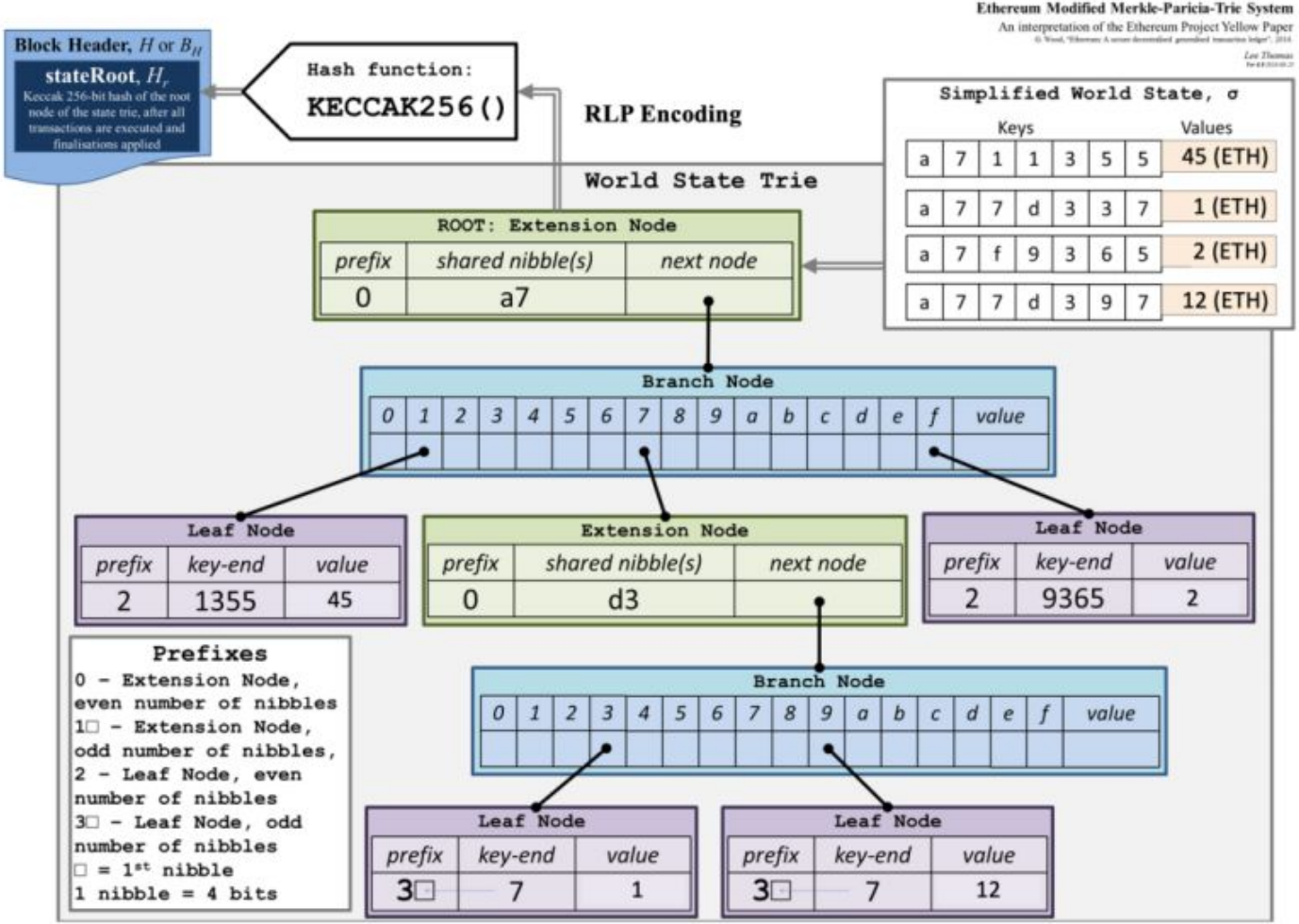
Merkle Trees



Basic Blockchain

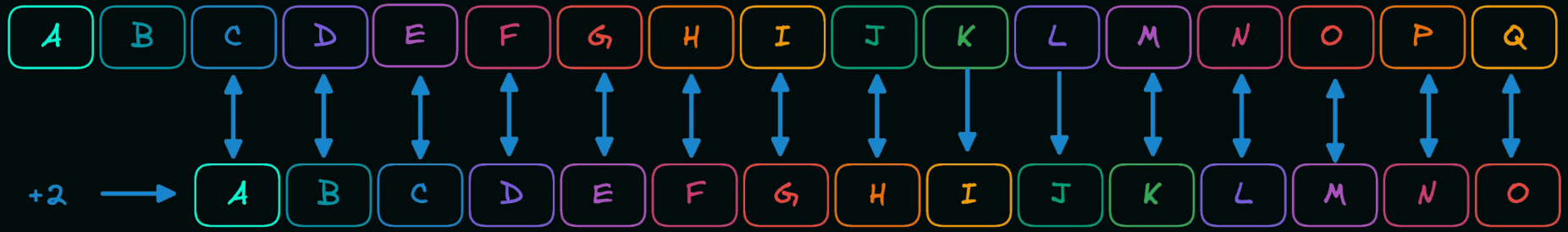


3Blue1Brown
How does bitcoin work?

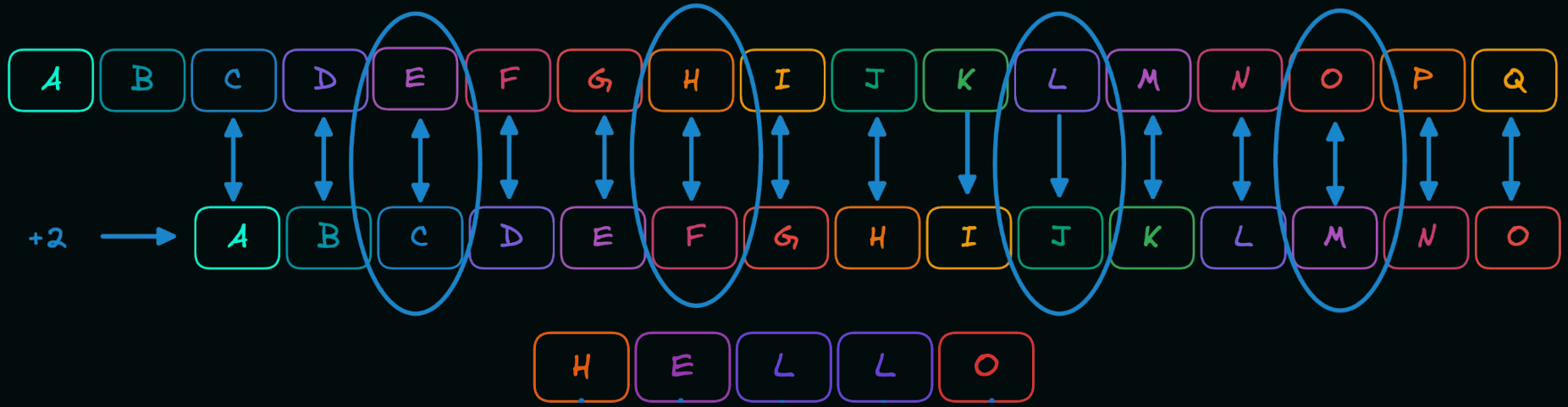

Ethereum Modified Merkle-Paricia-Trie System
An interpretation of the Ethereum Project Yellow Paper
Lee Thomas
Block Header, H or B_H
stateRoot, H_r
Keccak 256-bit hash of the root node of the state trie, after all transactions are executed and finalisations applied
Hash function:
KECCAK256()
RLP Encoding
Simplified World State, σ
Keys
Values
a 7 1 1 3 5 5 45 (ETH)
a 7 7 d 3 3 7 1 (ETH)
a 7 f 9 3 6 5 2 (ETH)
a 7 7 d 3 9 7 12 (ETH)
World State Trie
ROOT: Extension Node
prefix
shared nibble(s)
next node
0
a7
Branch Node
0 1 2 3 4 5 6 7 8 9 a b c d e f value
Leaf Node
prefix
key-end
value
2
1355
45
Extension Node
prefix
shared nibble(s)
next node
0
d3
Leaf Node
prefix
key-end
value
2
9365
2
Prefixes
0 - Extension Node, even number of nibbles
1□ - Extension Node, odd number of nibbles,
2 - Leaf Node, even number of nibbles
3□ - Leaf Node, odd number of nibbles
□ = 1st nibble
1 nibble = 4 bits
Branch Node
0 1 2 3 4 5 6 7 8 9 a b c d e f value
Leaf Node
prefix
key-end
value
3□
7
1
Leaf Node
prefix
key-end
value
3□
7
12


Encryption

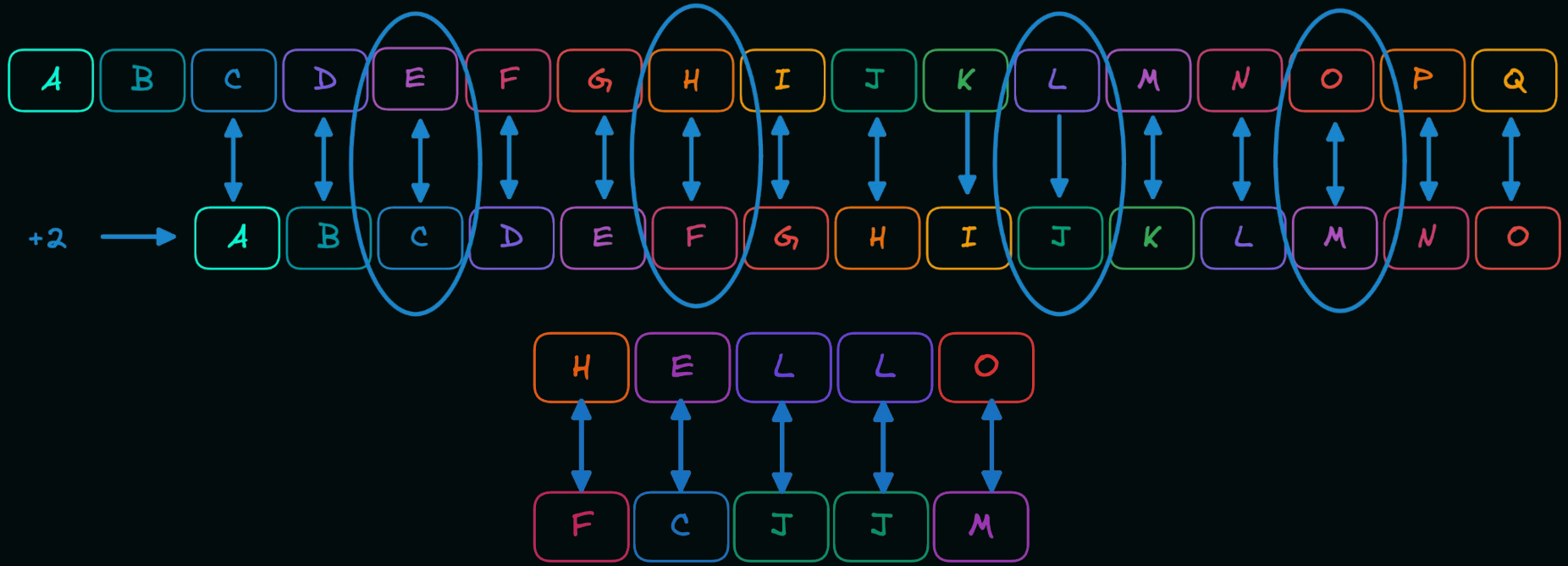
Cesar Cipher Example



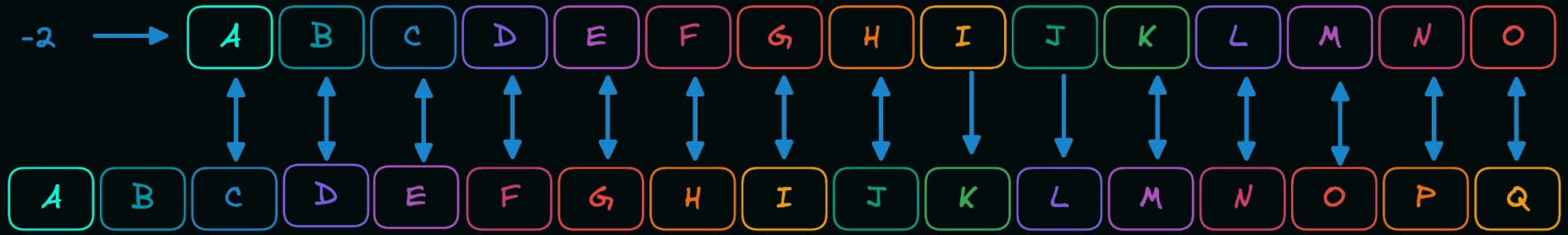
Cesar Cipher Example

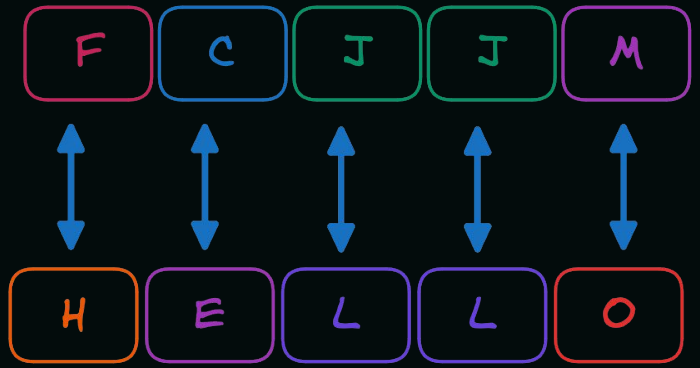


Cesar Cipher Example



Cesar Cipher Example



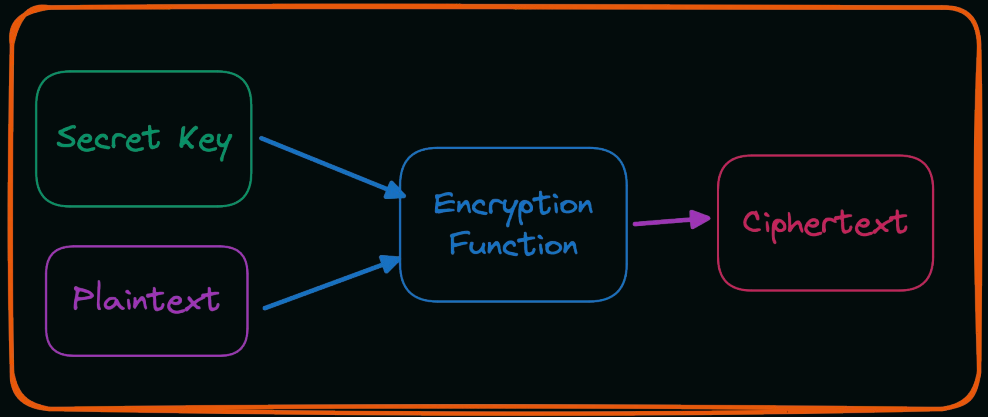


$\log_2(26-1) \approx 4.64$ bits

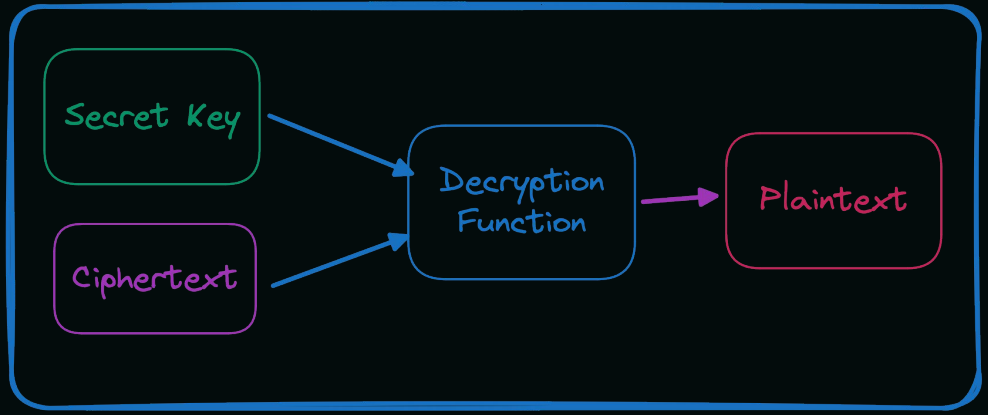
Symmetric Encryption

- All parties have the same “key” (think password)
- The same key is used to encrypt and decrypt
- No Authentication
- ***Fast***
- Data at rest (your laptop and phone) / TLS
- Example: AES

Encrypt



Decrypt

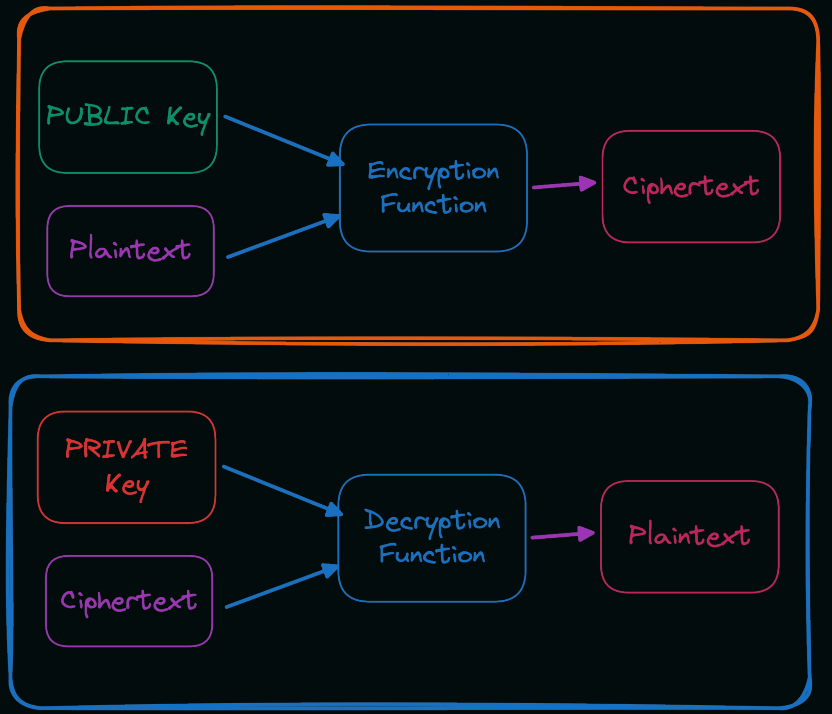


Asymmetric Encryption

- Pick a secret *priv_key*
- *pub_key* = derive_pubkey(*priv_key*)

- *c* = Enc(*pub_key*, *msg*)
- *msg* = Dec(*priv_key*, *c*)

- Examples: RSA

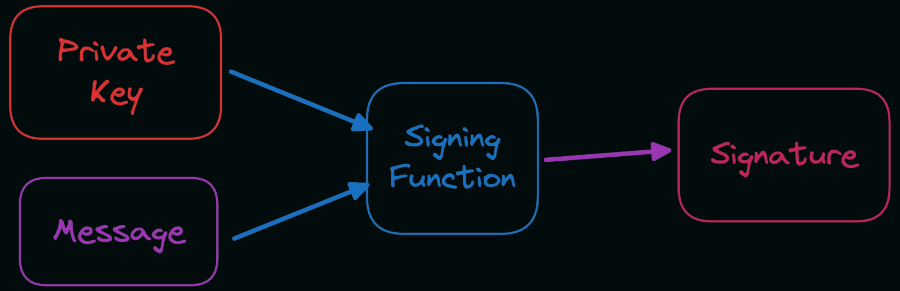


Signing ≈ Watermark

- Pick a secret *priv_key*
- *pub_key* = derive_pubkey(*priv_key*)

- *sig* = Sign(*priv_key*, *msg*)
- Verify(*sig*, *pub_key*, *msg*)

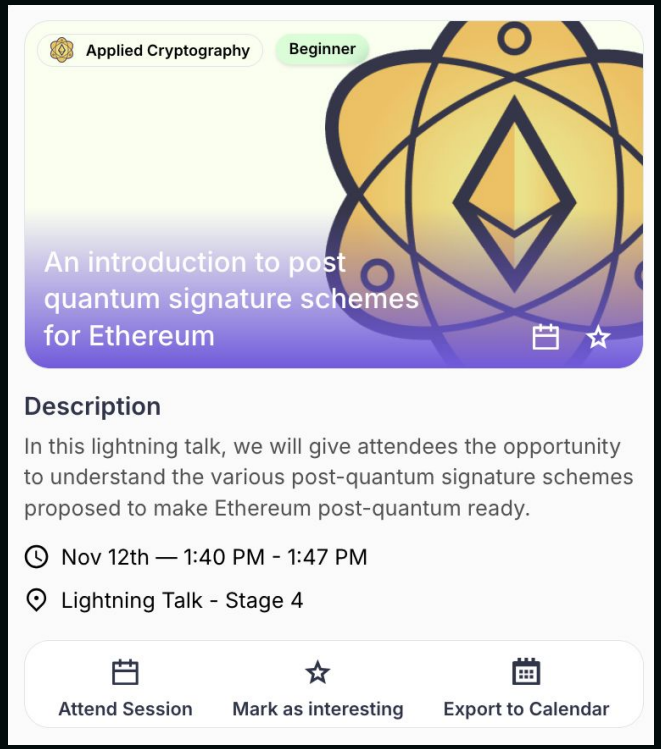
- Examples: RSA, BLS, (EC)DSA



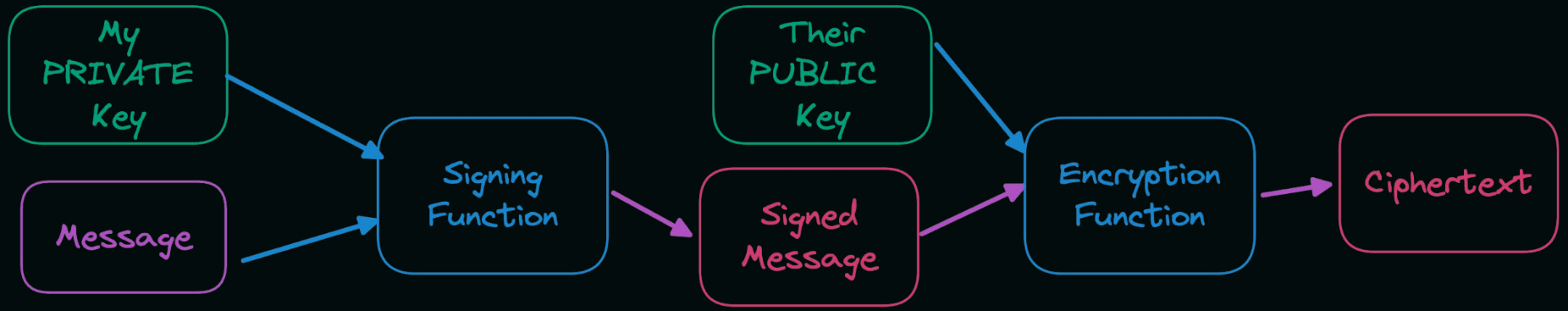
Signing ≈ Watermark

- Unique signature using a private key and the message.
- Ensures authenticity
- Verifiable with the public key
- Signature can't be used to find the private key or original message

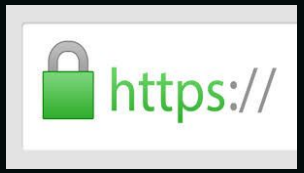
Signature schemes @ devcon vii

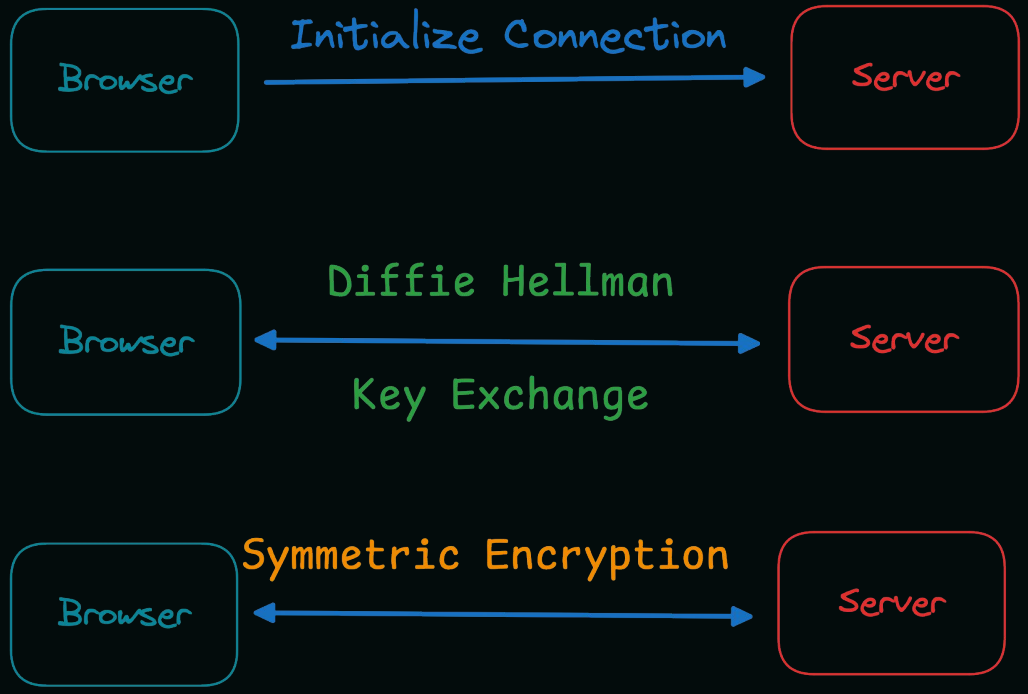


Signing & Encrypting a Message



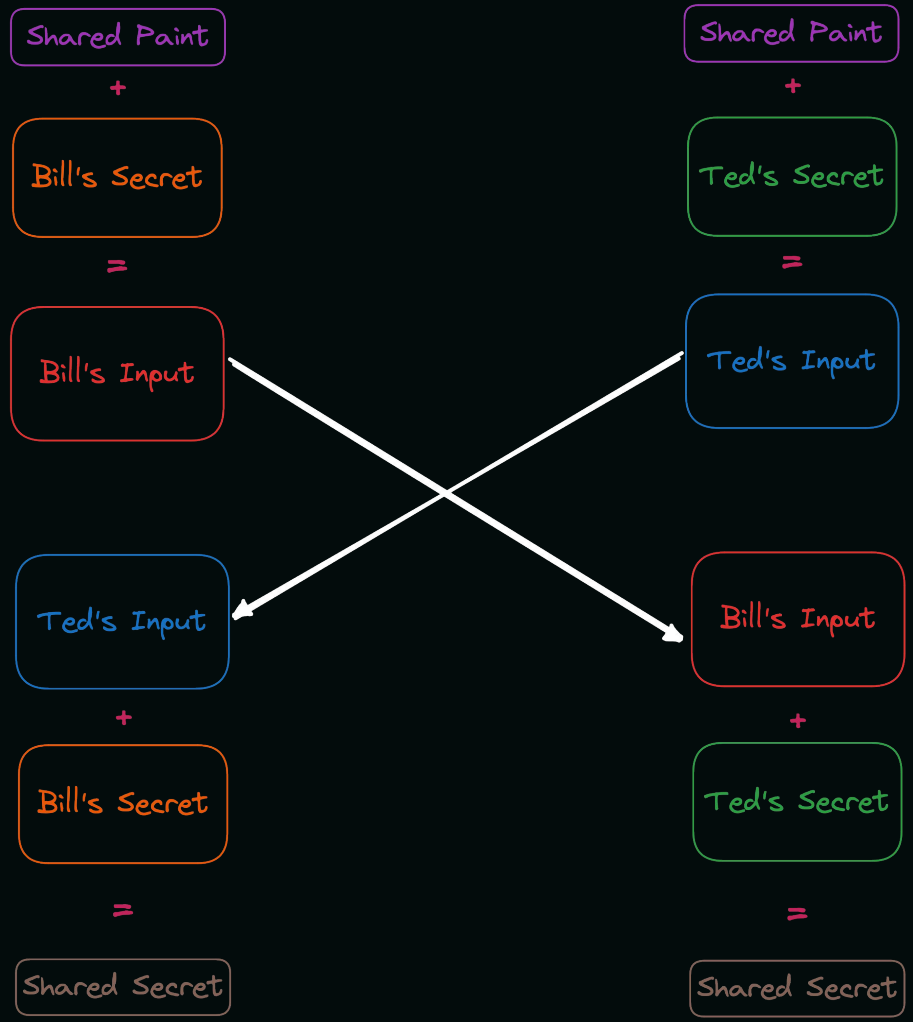
e.g. HTTPS


https://


Browser
Initialize Connection
Server
Browser
Diffie Hellman
Key Exchange
Server
Browser
Symmetric Encryption
Server

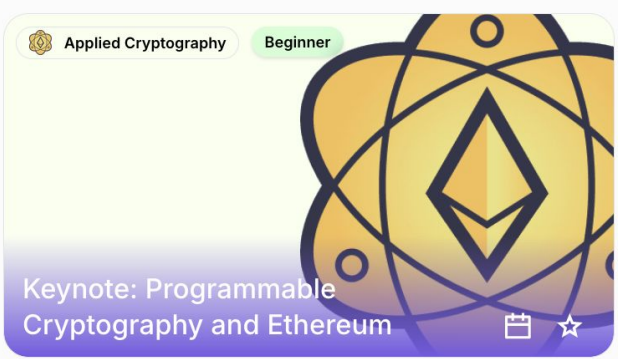
Diffie Hellman

- Way of exchanging info to make a shared symmetric key (secret) without anyone being able to spy
- Used in TLS, Signal




“Programmable Cryptography”:
ZK, (S)MPC, FHE

Programmable cryptography @ devcon vii



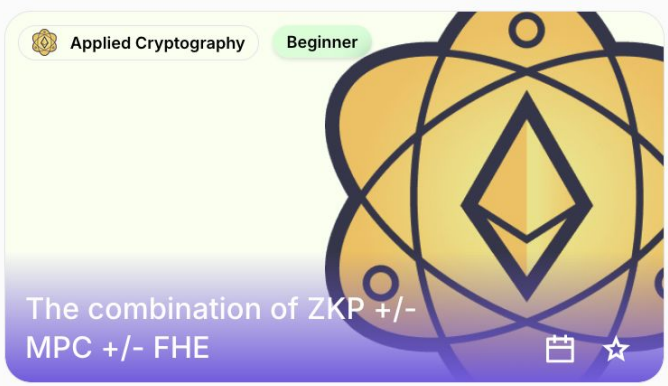
Description

Programmable Cryptography is a "second generation" of cryptographic primitives - primitives that allow arbitrary programs to be executed "inside of" or "on top of" cryptographic objects. Programmable cryptography provides three key affordances that complement and amplify the affordances of Ethereum--verifiability, confidentiality, and non-interactivity. We'll discuss how these technologies can reshape the Internet over the next 50 years.

STARTS IN 32 MINUTES

Nov 12th — 3:00 PM - 3:25 PM

Talk - Main Stage



Description

This talk will provide you with the necessary intuition to understand when you should use ZKP, MPC or FHE, or any combination of them.

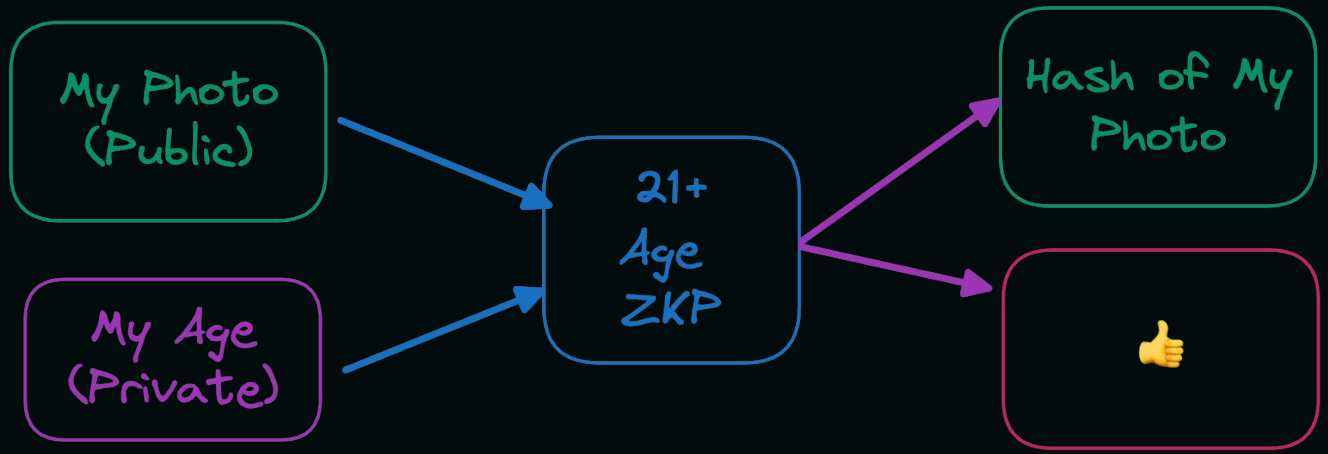
Nov 12th — 12:40 PM - 12:47 PM

Lightning Talk - Stage 4

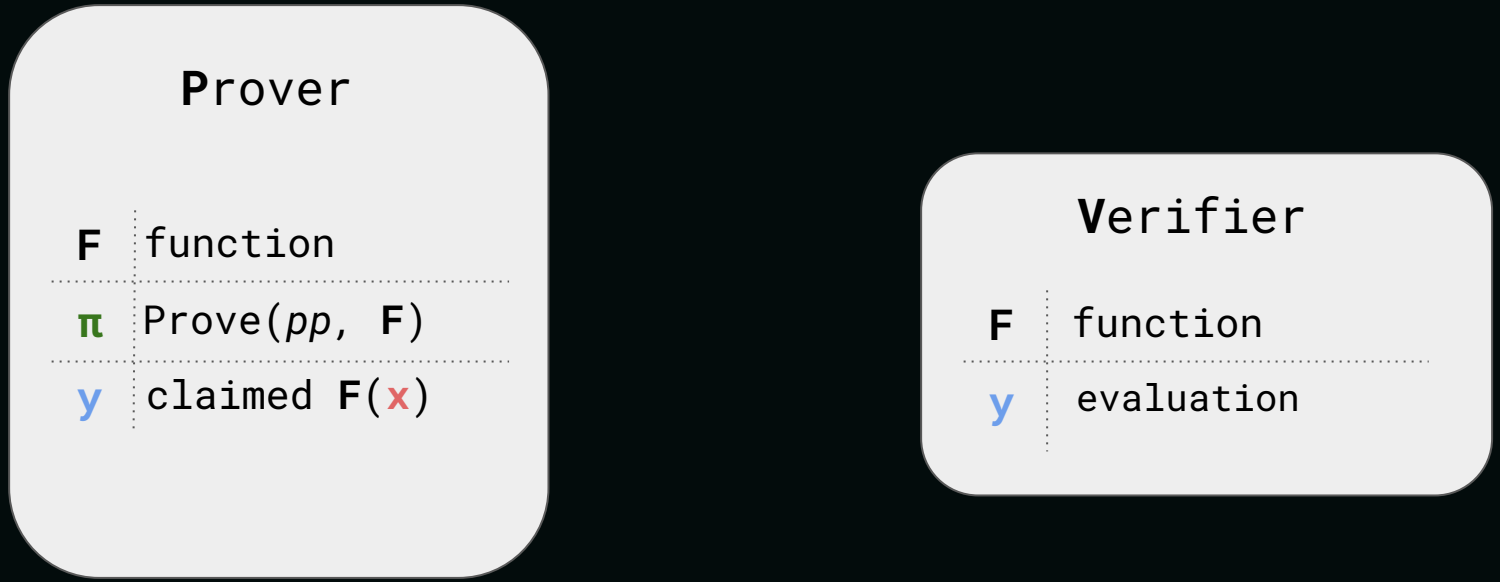
Attend Session | Mark as interesting | Export to Calendar

Zero Knowledge

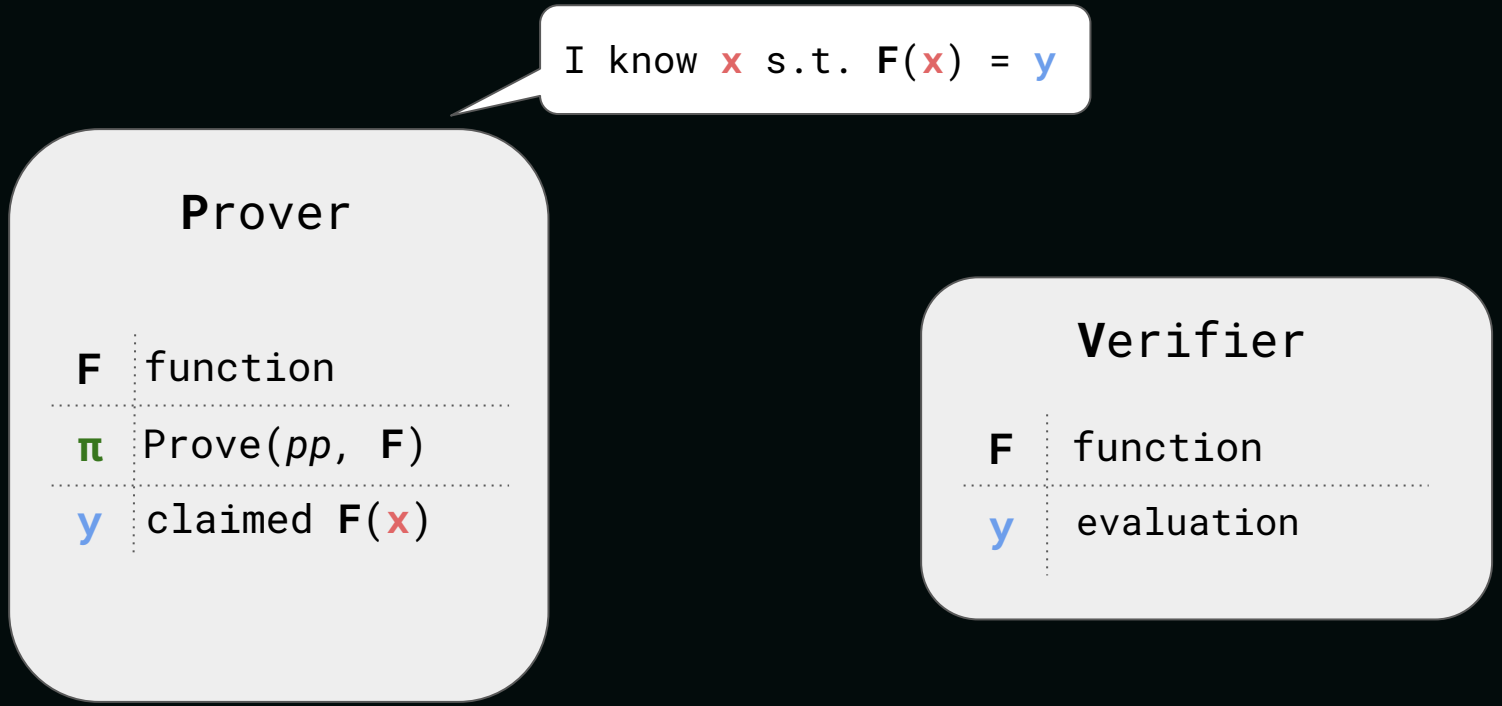
- A “prover” can prove statements to a “verifier”
- The verifier is convinced with high probability that the prover knows a satisfying secret
- The verifier learns nothing about the prover’s secret beyond the relation



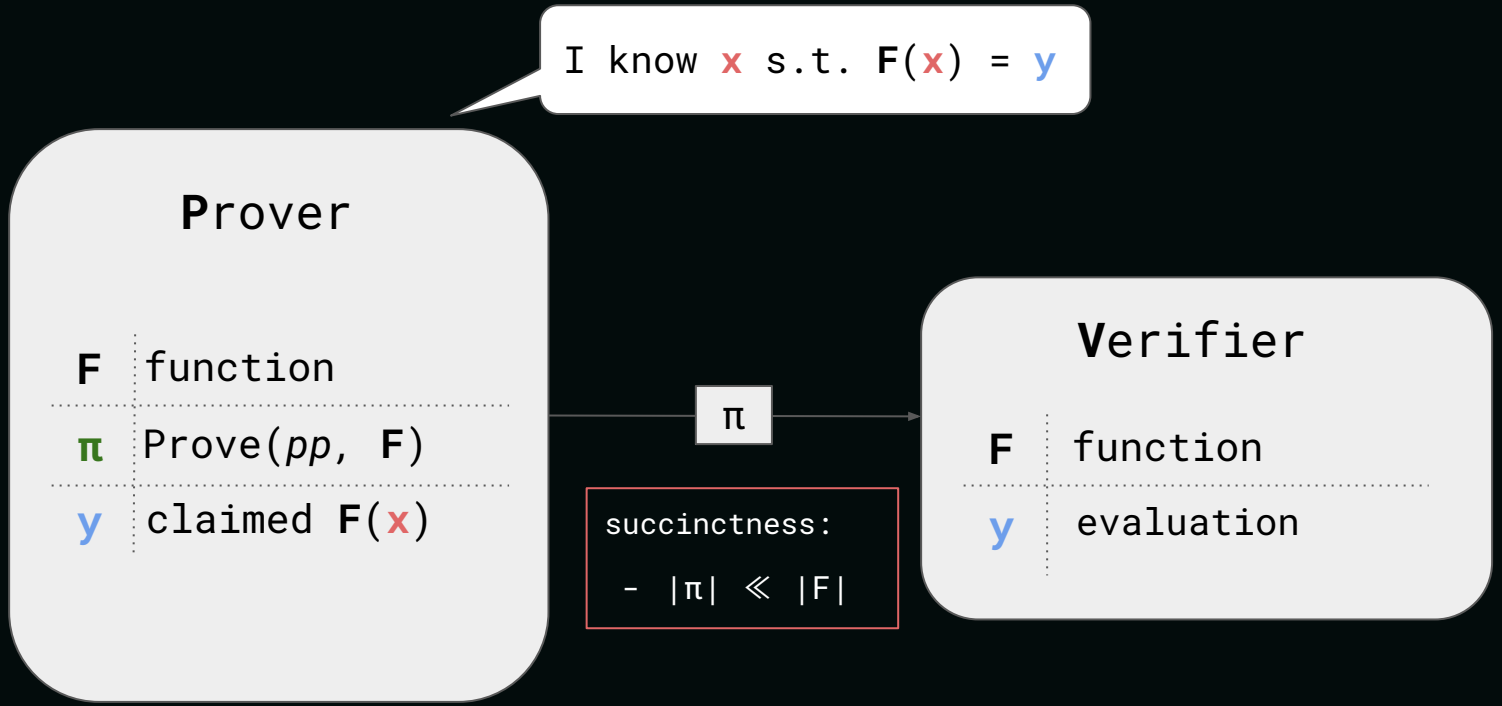
Zero Knowledge



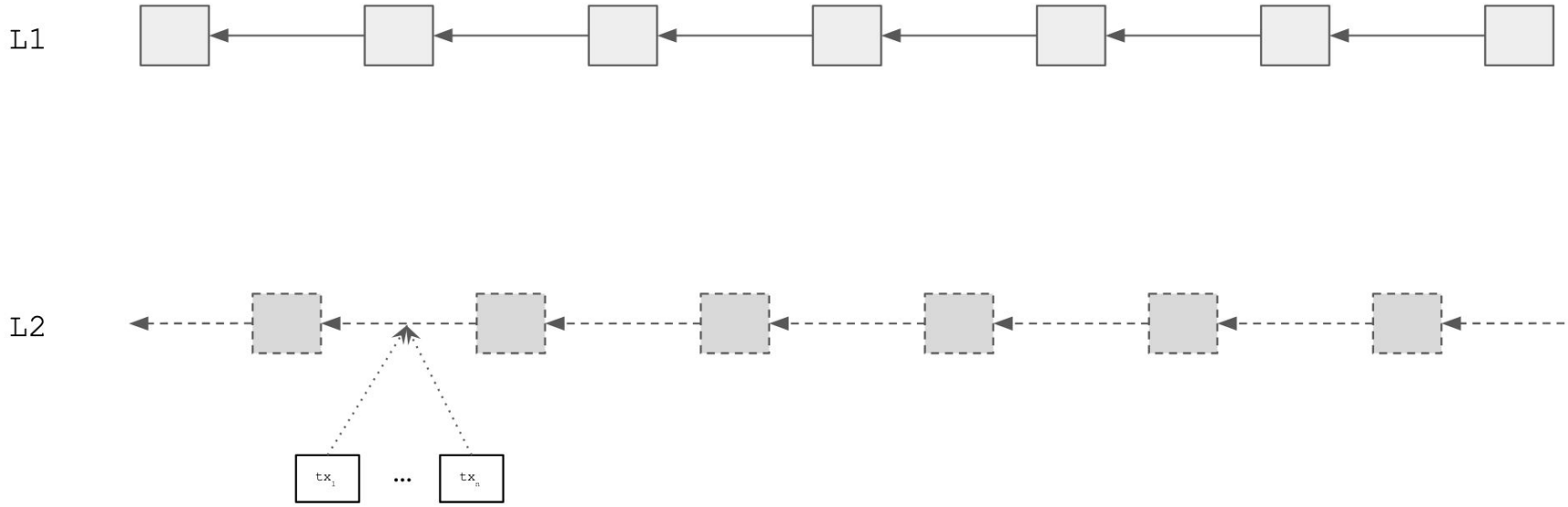
Zero Knowledge



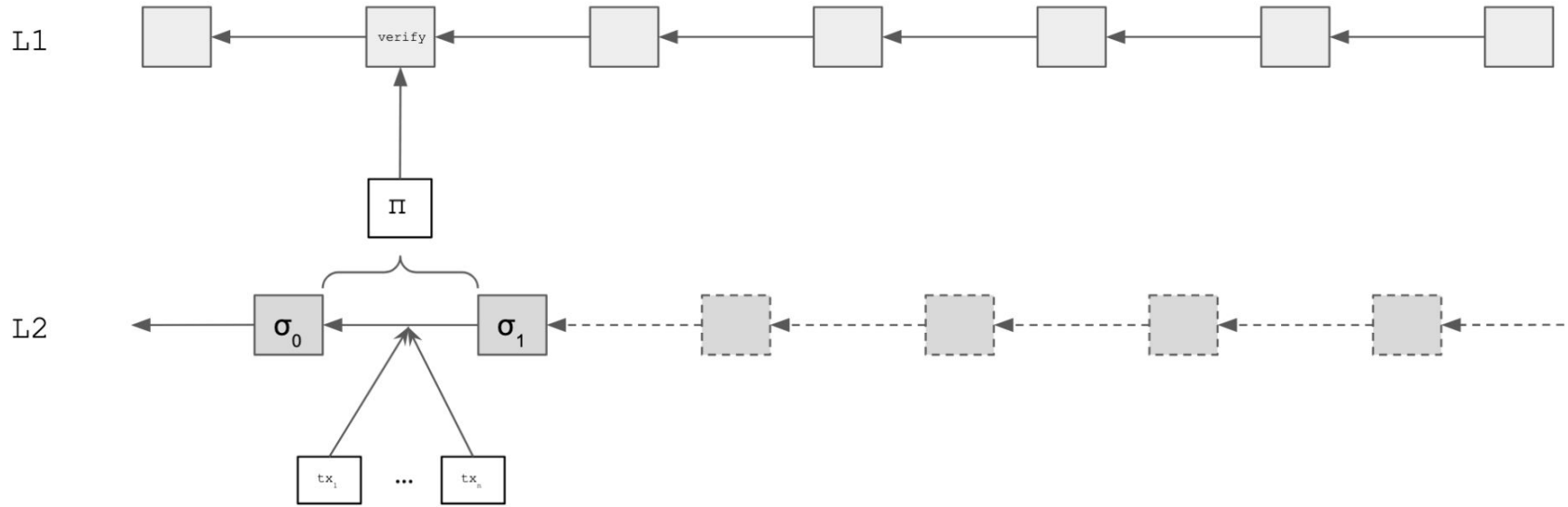
Zero Knowledge



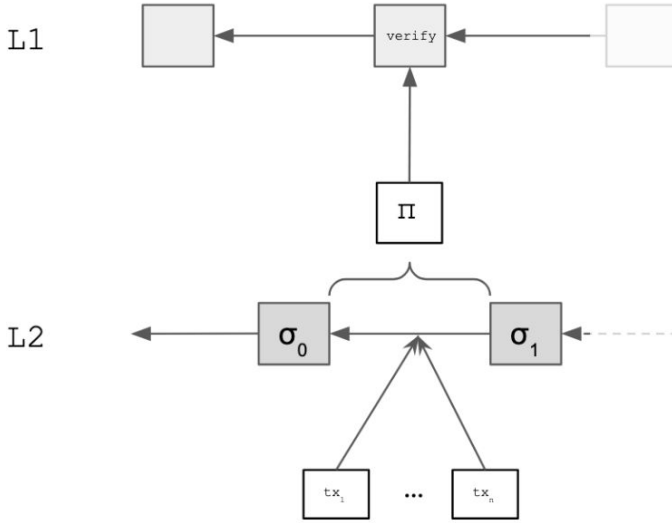
Use-case: SNARKs for succinct rollups



Use-case: SNARKs for succinct rollups



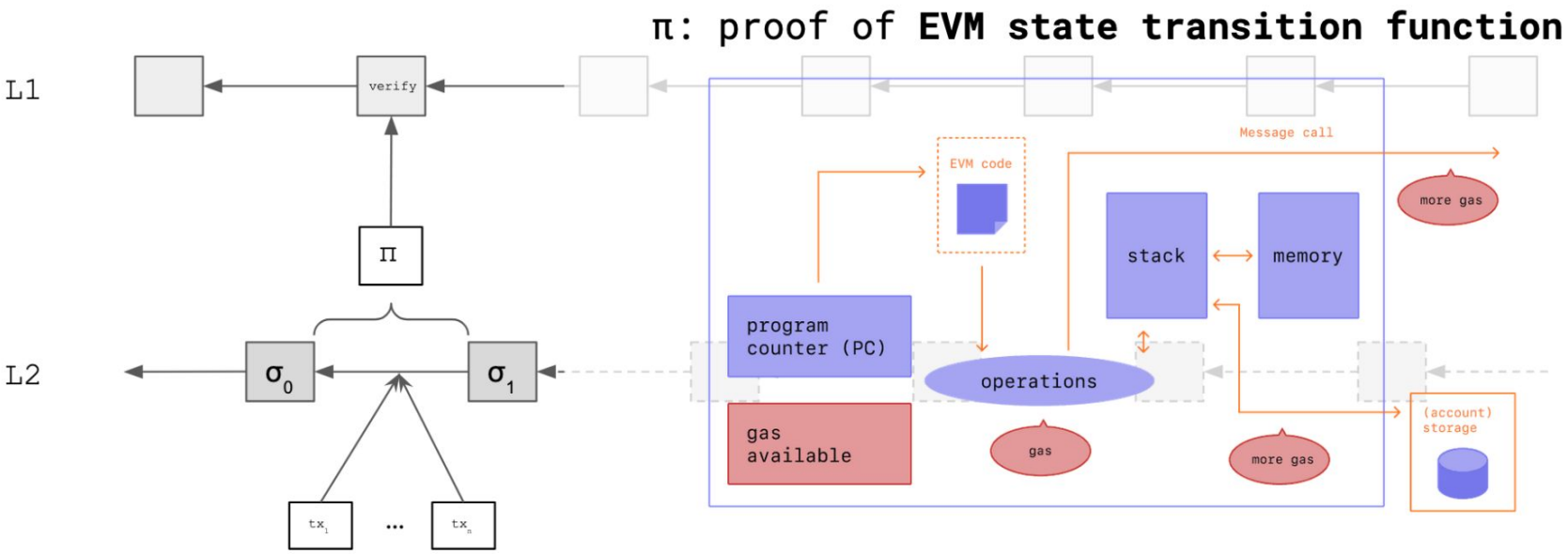
Use-case: SNARKs for succinct rollups



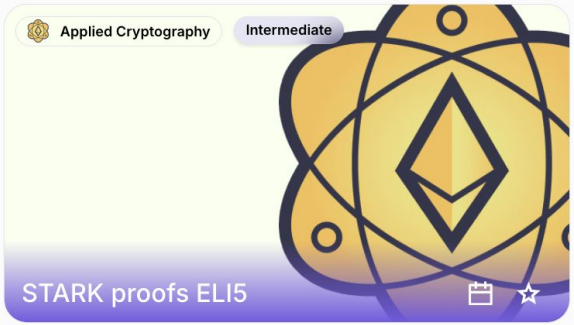
π: proof of **zkRollup** spend statement, e.g.

- tx is signed by owner of funds;
- tx is spending a note that exists;
- tx is not double-spending;
- etc.

Use-case: SNARKs for succinct rollups



zk @ devcon vii

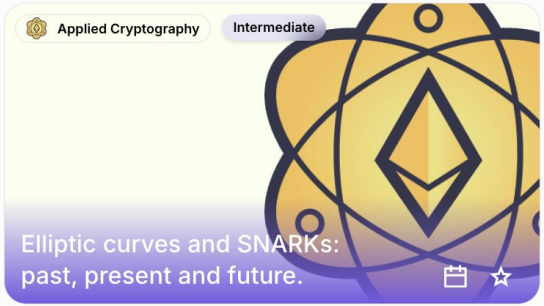


Description

Let's face it, ZK proofs are intimidating. But they don't have to be! ZK proofs are complex not because of the depth math they use, but because of the large number of fields of mathematics they leverage features from. In this talk, we'll break down STARK proofs into simple blocks and colorful analogies so that you get a good high level overview of how they work

Nov 12th — 1:50 PM - 1:57 PM

Lightning Talk - Stage 4



Description

Elliptic curves are used in many proof systems. Some systems (e.g. Bulletproofs) use plain curves (e.g. ed25519). Some (e.g. Groth16, KZG-PLONK) use pairing-friendly curves (e.g. BLS12-381). Some recursive systems require pairing-friendly 2-cycle (e.g. MNT4/6) or 2-chains (e.g. BLS12-377/BW6-761). Some other recursive/folding systems require plain 2-cycle (e.g. Pasta). In this talk we will go through the difference between these curves and why there isn't a silver bullet curve for all scenarios.

Nov 12th — 5:00 PM - 5:25 PM

Talk - Stage 3

Description

Linear codes underlie one of the main approaches in zero-knowledge proofs and arguments, including works like FRI, Ligero, Brakedown and Orion. In this talk, we describe how to extend one of the protocols from Ligero and Brakedown to the regime of batched polynomial commitments, at the cost of a single extra operation in the verifier. Similarly to Redshift, we opt for increased efficiency via the list decoding regime. We also present an optimisation for using the resulting commitment with PIOPs.

Nov 12th — 6:00 PM - 6:25 PM

Talk - Stage 3

applied zk @ devcon vii

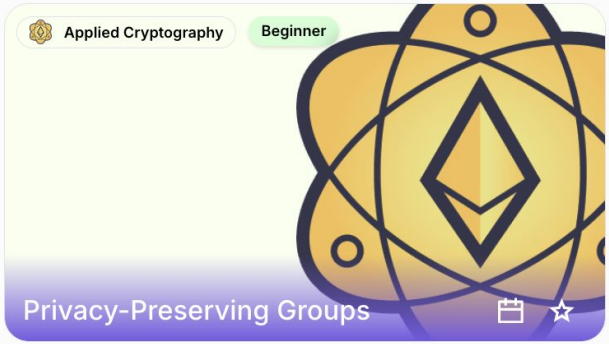
Description

MACI is a protocol that can be used to run private on chain polls. This talk will introduce the protocol, dive into some of the technical aspects. Finally we will talk about the team's plans for the future and how the community can get involved to help improve the project.

Nov 12th — 2:00 PM - 2:07 PM

Lightning Talk - Stage 4

Attend Session | Mark as interesting | Export to Calendar

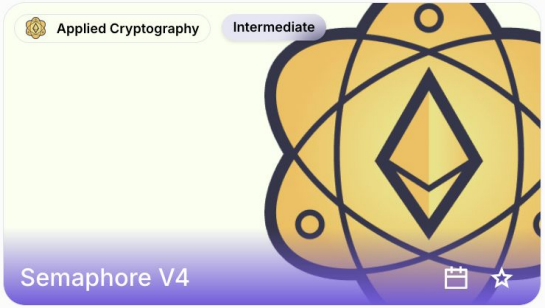


Description

This talk will explore the concept of privacy-preserving groups and the challenges associated with managing them. It will cover different ideas to add anti-sybil mechanisms to enhance group security and trust. The presentation will also highlight real-world projects working on it and provide practical use cases to illustrate their application and impact.

Nov 12th — 2:30 PM - 2:37 PM

Lightning Talk - Stage 4



Description

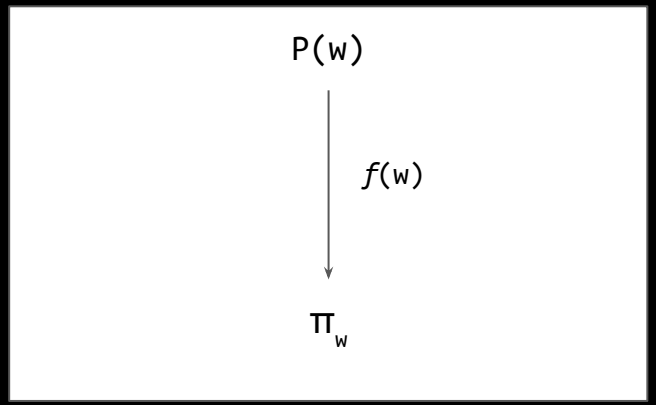
Semaphore is a protocol enabling individuals to prove group membership and send messages (such as votes or endorsements) anonymously. The latest version enhances efficiency and simplifies the use of libraries and contracts. This presentation will cover the new features, project vision, and the importance and challanges of zero-knowledge technologies.

STARTS IN 2 MINUTES

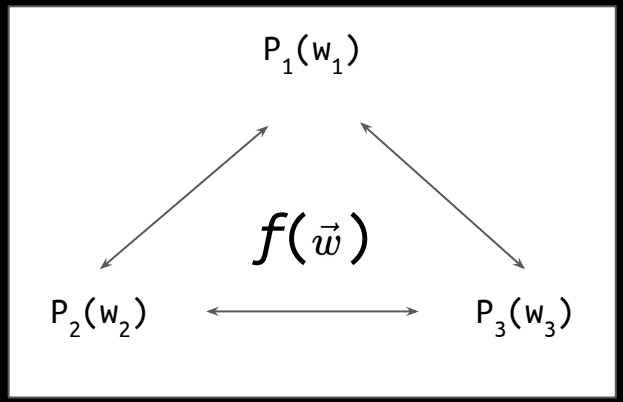
Nov 12th — 2:40 PM - 2:47 PM

Lightning Talk - Stage 4

MPC - Secure Multiparty Computation

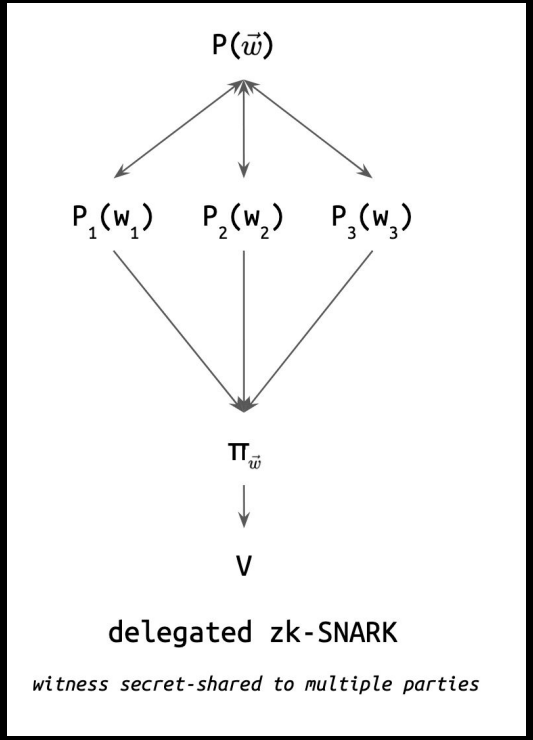


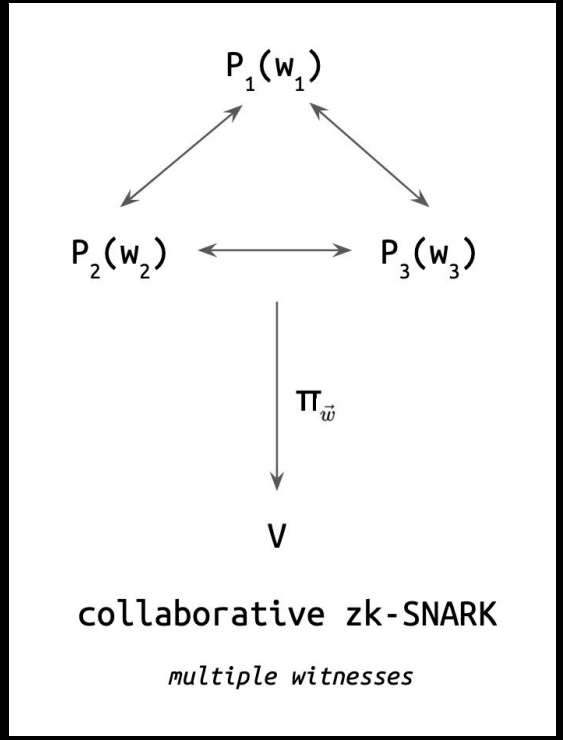
zk-SNARK



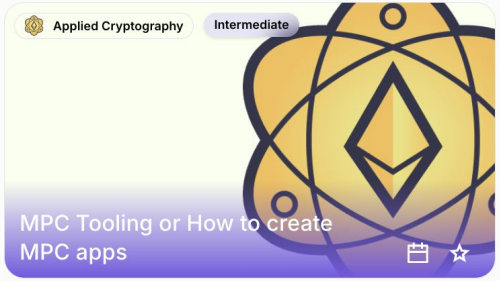
MPC

MPC - Secure Multiparty Computation





MPC @ devcon vii

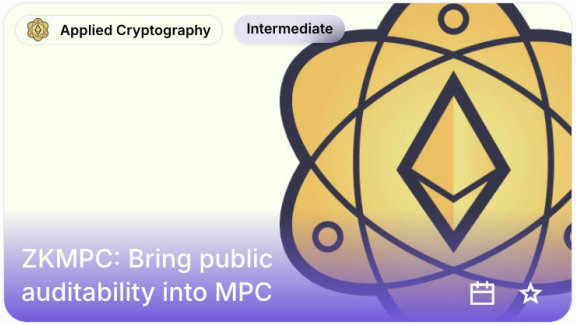


Description

Let's get into the state of the art of MPC development: we'll discuss different MPC schemes, current MPC tooling & how you can create MPC apps today. We'll cover the tech stack from a frontend level (e.g. MPC compilers) to a backend - and of course how we can combine them.

Nov 12th — 12:50 PM - 12:57 PM

Lightning Talk - Stage 4

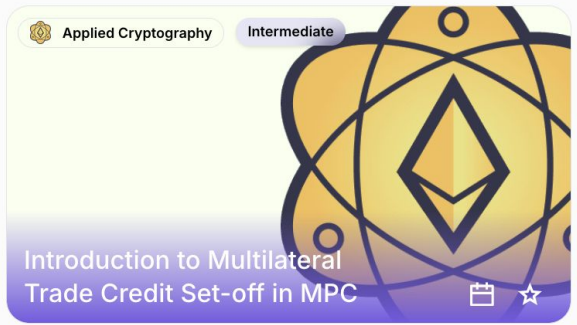


Description

In multi-party computation (MPC), participants collaboratively compute without revealing private inputs. To secure MPC on a blockchain, preventing collusion is essential. We developed a "publicly auditable" version of SPDZ, a widely-used MPC protocol, that enables third-party verification through zero-knowledge proofs (ZKP) collaboratively generated by multiple parties. We will also demonstrate application examples, such as a Game Master-free werewolf game.

Nov 12th — 5:30 PM - 5:55 PM

Talk - Stage 3



Description

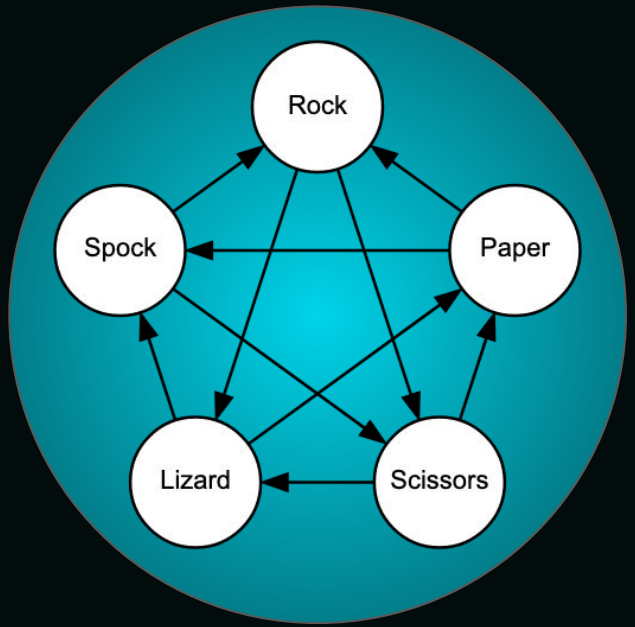
Multilateral Trade Credit Set-off is a process for collecting outstanding invoices from a network of firms and detecting cycles. A cycle is a circular pattern of due payments that connects businesses. Removing a cycle yields liquidity savings for the firms involved. This process is done by a central agency that collects the invoices and performs the netting. Instead, we leverage MPC to perform the set-ff while preserving the privacy of sensitive financial data of the firms

Nov 12th — 1:00 PM - 1:07 PM

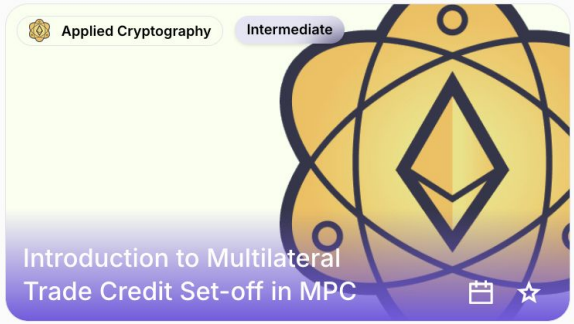
Lightning Talk - Stage 4

MPC Demo

Rock Paper Scissors Lizard Spock



applied MPC @ devcon vii

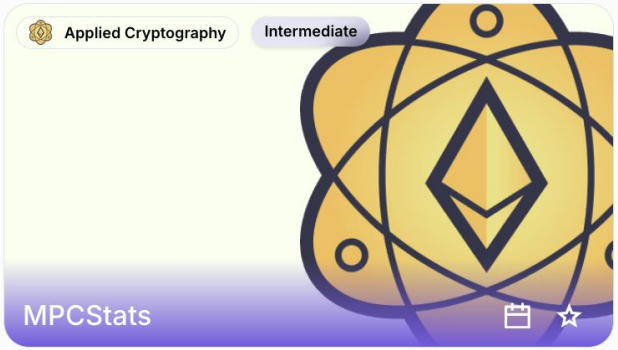


Description

Multilateral Trade Credit Set-off is a process for collecting outstanding invoices from a network of firms and detecting cycles. A cycle is a circular pattern of due payments that connects businesses. Removing a cycle yields liquidity savings for the firms involved. This process is done by a central agency that collects the invoices and performs the netting. Instead, we leverage MPC to perform the set-ff while preserving the privacy of sensitive financial data of the firms

Nov 12th — 1:00 PM - 1:07 PM

Lightning Talk - Stage 4



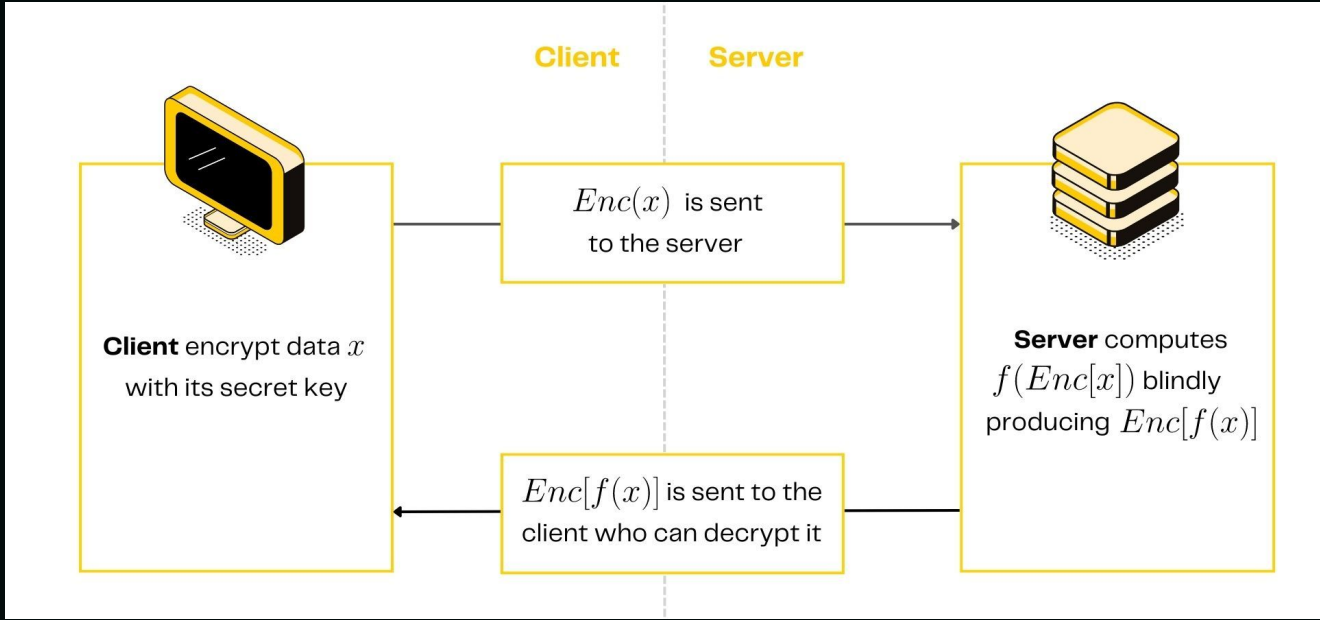
Description

MPCStats is a framework allowing data consumers to query statistical computation from either one or multiple data providers while preserving privacy to those raw data. We support standard statistical operations, including nested and filter ones. Data providers do not leak their data and data consumers can be convinced the computation is done correctly.

Nov 12th — 2:20 PM - 2:27 PM

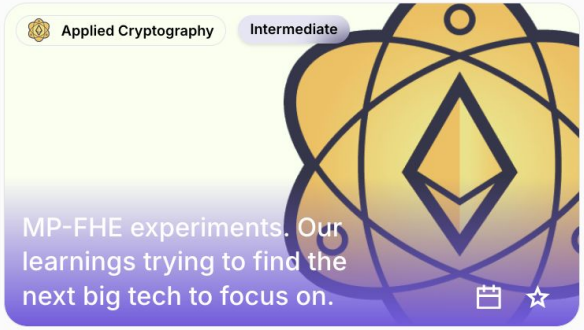
Lightning Talk - Stage 4

FHE – Fully Homomorphic Encryption



source: https://www.zama.ai/post/private-smart-contracts-using-homomorphic-encryption

FHE @ devcon vii



Description

This talk mainly focuses on showcasing the work that some PSE members did while starting to dive into MPC-FHE during Q2 2024. This work is composed by various explorations within the MPC-FHE realm that move towards different directions and goals. From FHE compilers to FFT Bootstrapping GPU optimization proposals, passing by FHE Game demos and many application level implementations, the talk aims to reach beginner-advanced audience on the research/product paths that we have explored so far.

Nov 12th — 1:10 PM - 1:17 PM

Lightning Talk - Stage 4