Norwegian University of Science and Technology

Zero-Knowledge Proofs: Simultaneously ensuring integrity and privacy

Tjerand Silde @ NDC Security 2026

Introduction

Associate Professor in Cryptology

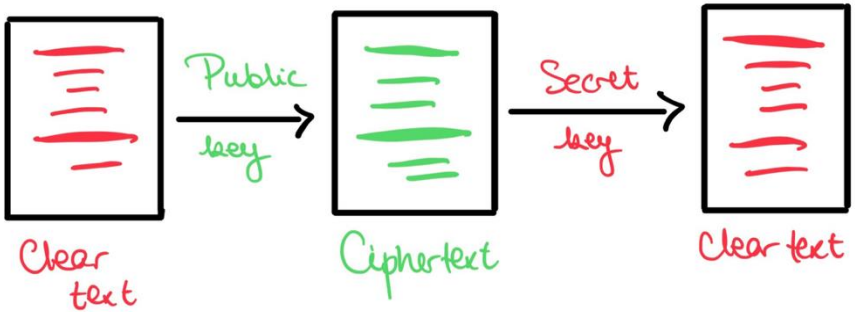
Department of Information Security and Communication Technology at NTNU

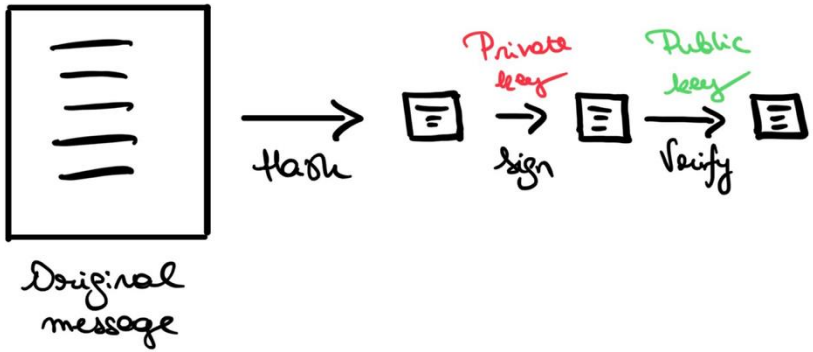
Leading NTNU Applied Cryptology Lab

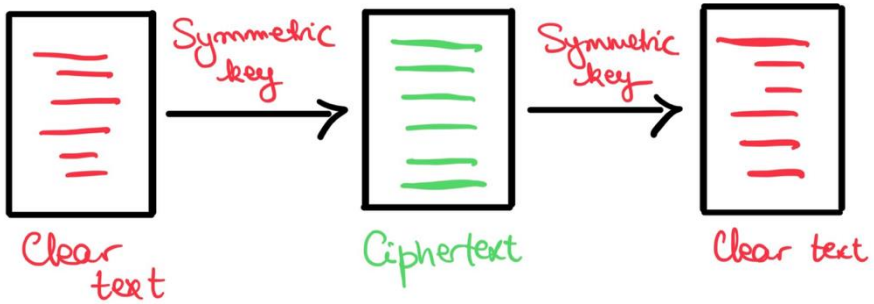
Quantum-safe cryptography, privacy applications & secure implementation

Cryptography Expert @ Pone Biometrics

Cryptography Today







Cryptography Today

Secure messaging:	Signal, WhatsApp, iMessage, …
Secure connections:	TLS, SSH, IPsec, …
Digital authentication:	FIDO, Buypass ID, Bank ID, …
Payments:	PayPal, VISA / Mastercard, Bitcoin, Apple / Google Pay, Vipps, …

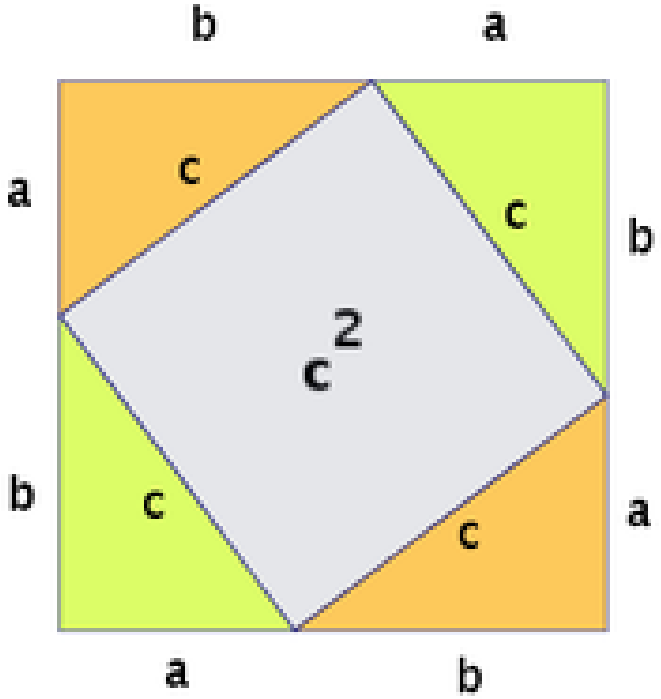
What else is out there?

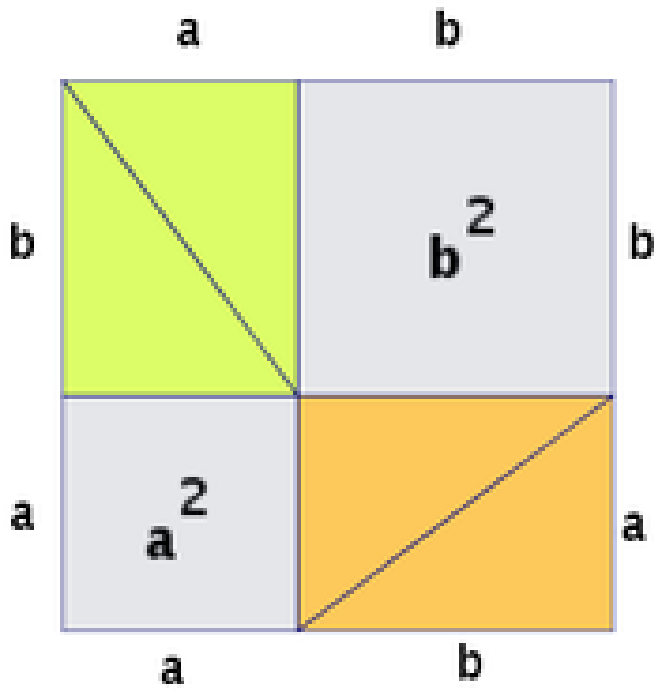
Privacy vs. Integrity

- Users want to hide sensitive data (identity, vote, transaction, health status, model weights…)
- Systems must ensure the hidden data is valid, well-formed, and not malicious, before consumed
- We want to achieve data-sharing without breaking the privacy or the integrity of the data involved

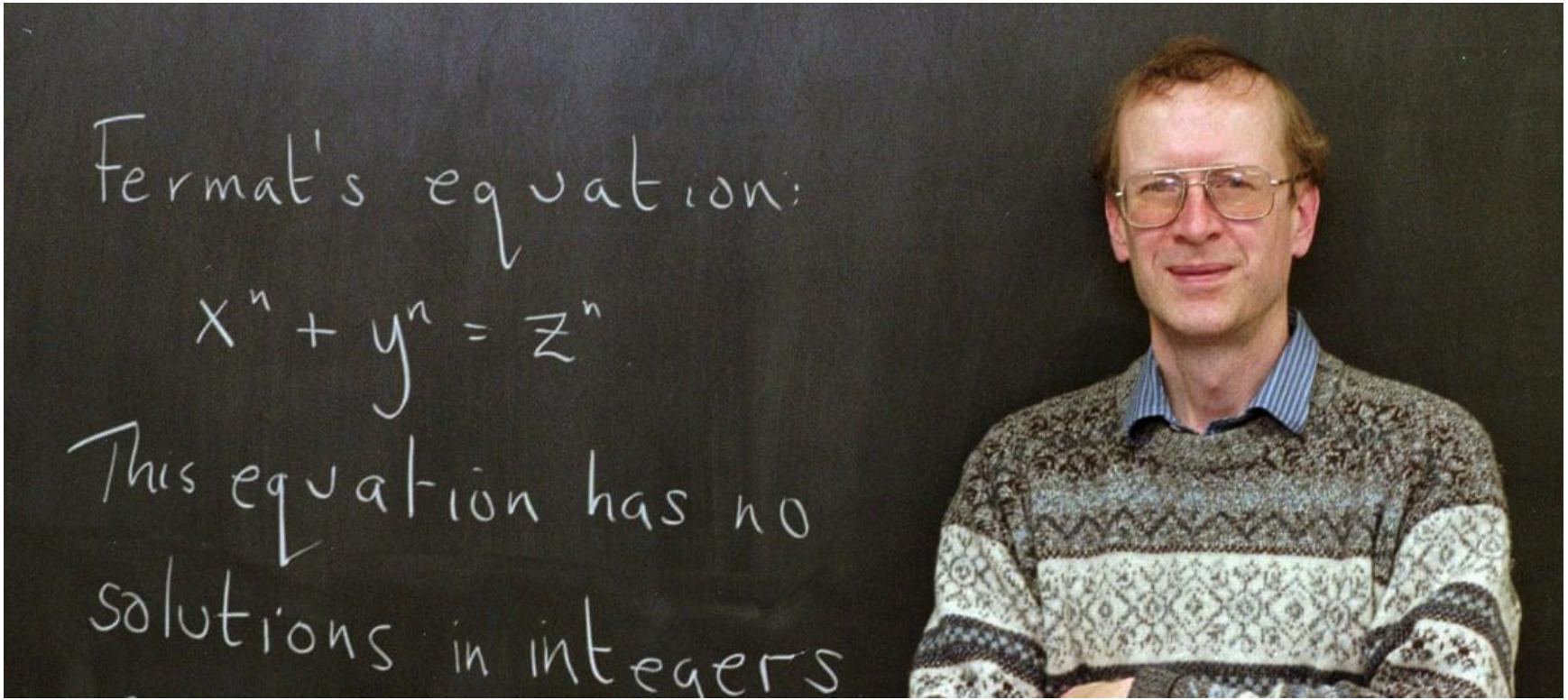
ZERO-KNOWLEDGE PROOFS

Mathematical Proofs

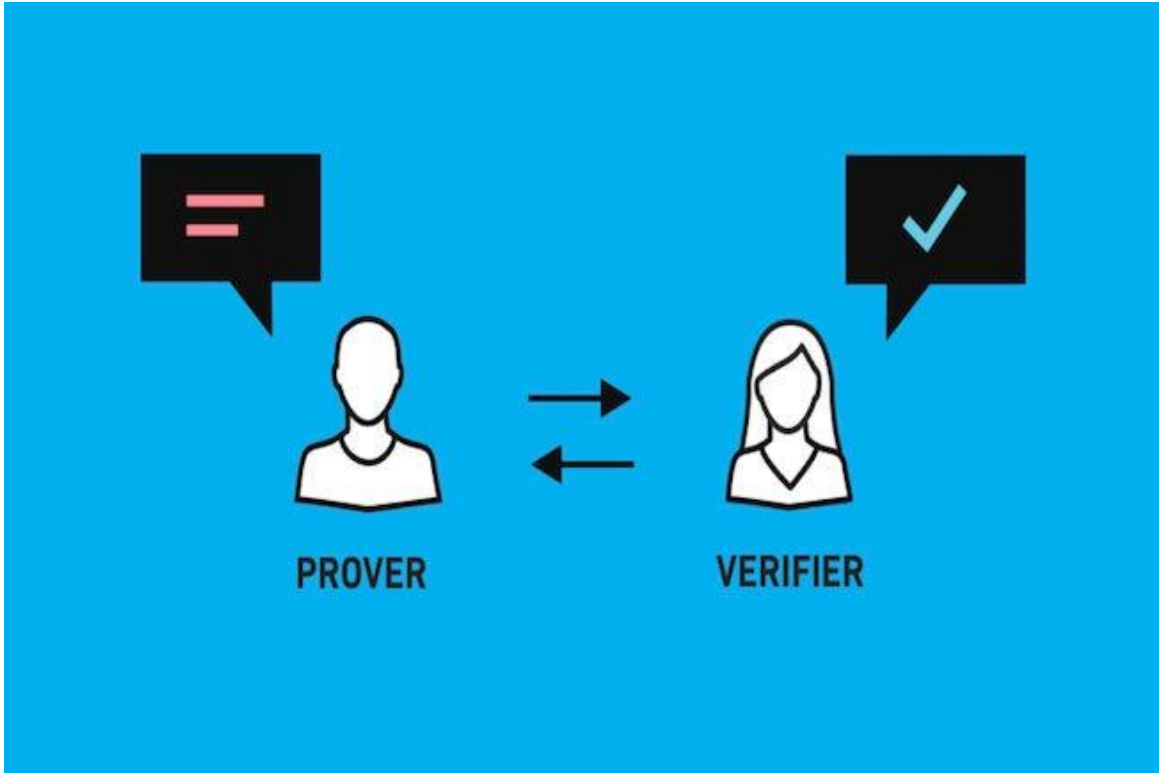




Mathematical Proofs



Zero-Knowledge Proofs



Zero-Knowledge Proofs

The prover publishes a statement and keeps a secret witness.

- **Correctness**: the protocol works with the secret ✅
- **Soundness**: one cannot cheat without the secret ❌
- **Zero-knowledge**: the protocol does not leak the secret 🤫

Goldwasser, Micali, and Rackoff (1985)

The Knowledge Complexity of Interactive Proof-Systems

(Extended Abstract)

Shafi Goldwasser
MIT

Silvio Micali
MIT

Charles Rackoff
University of Toronto

NTNU | Norwegian University of Science and Technology

Zero-Knowledge vs. Mathematical Proofs

- ZKPs allows for interaction between parties, while mathematical proofs are written down once and for all
- ZKPs allows for a (cryptographically) small cheating prob., while mathematical proofs are perfectly correct
- ZKPs might be much smaller in size than the witness itself
- (ZKPs can also be non-interactive by using hash-functions)

APPLICATIONS FROM ZKP

Quantum-Safe Signatures

FIPS 204

Federal Information Processing Standards Publication

Module-Lattice-Based Digital Signature Standard

Category: Computer Security **Subcategory: Cryptography**

Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8900

Signatures from ZKP

Private information: $\mathbf{s}_1 \in [\beta]^m, \mathbf{s}_2 \in [\beta]^n$
Public information: $\mathbf{A} \in \mathcal{R}_{q,f}^{n \times m}, \mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 \in \mathcal{R}_{q,f}^n$

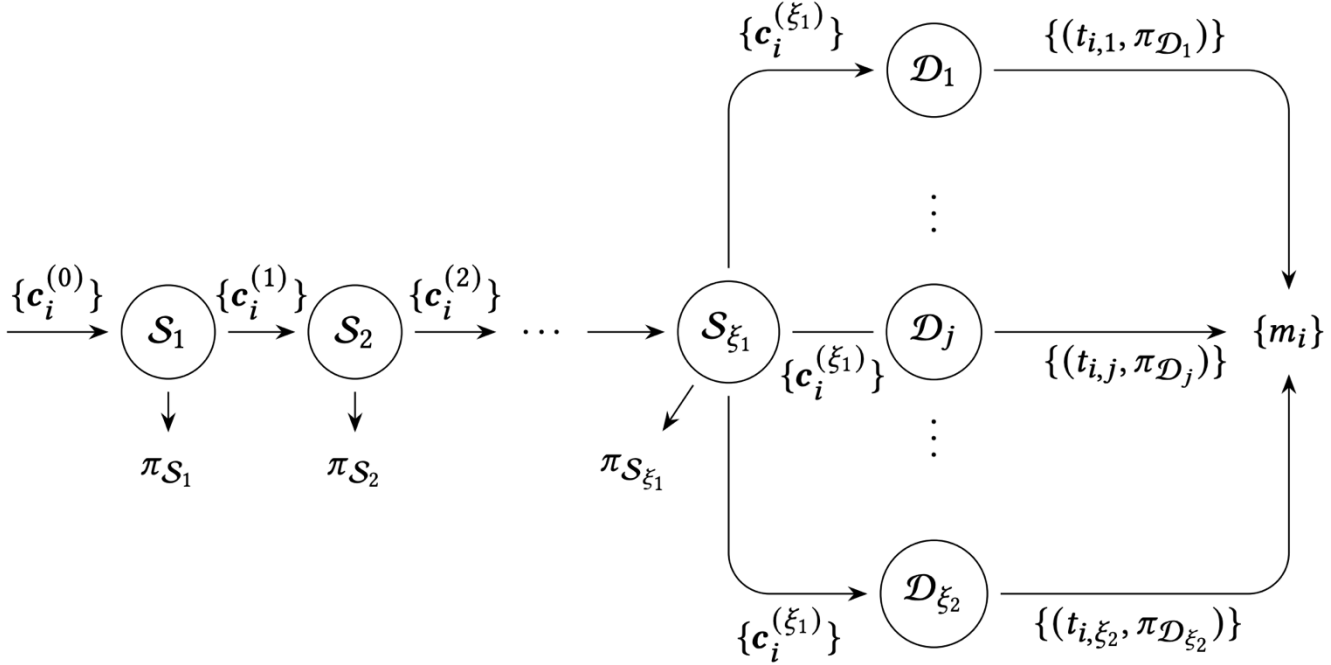
Prover		Verifier
$\mathbf{y}_1 \leftarrow [\gamma + \bar{\beta}]^m$ $\mathbf{y}_2 \leftarrow [\gamma + \bar{\beta}]^n,$ $\mathbf{w} := \mathbf{A}\mathbf{y}_1 + \mathbf{y}_2$		
	$\xrightarrow{\mathbf{w}}$	
		$c \leftarrow \mathcal{C}$
	$\xleftarrow{c}$	
$\mathbf{z}_1 := c\mathbf{s}_1 + \mathbf{y}_1$ $\mathbf{z}_2 := c\mathbf{s}_2 + \mathbf{y}_2$ if $\mathbf{z}_1 \notin [\bar{\beta}]^m$ or $\mathbf{z}_2 \notin [\bar{\beta}]^n$ then $(\mathbf{z}_1, \mathbf{z}_2) := \perp$		
	$\xrightarrow{(\mathbf{z}_1, \mathbf{z}_2)}$	
		Accept iff $\mathbf{z}_1 \in [\bar{\beta}]^m$ and $\mathbf{z}_2 \in [\bar{\beta}]^n$ and $\mathbf{A}\mathbf{z}_1 + \mathbf{z}_2 - c\mathbf{t} = \mathbf{w}$

Electronic Voting

Electronic Voting

- Prove that all ciphertexts contains valid votes
- Prove that encrypted votes are shuffled correctly
- Prove that encrypted votes are decrypted correctly

Electronic Voting

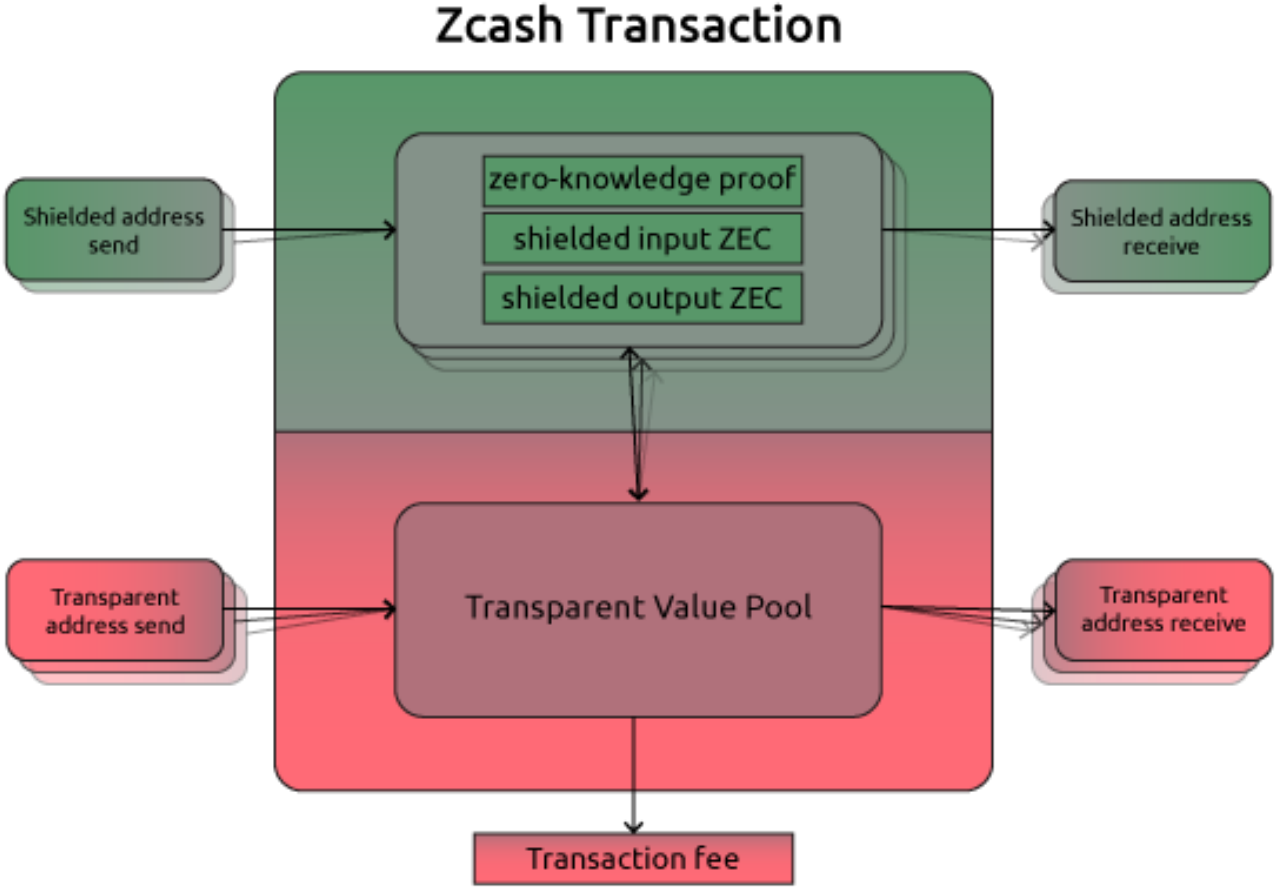


Anonymous Transactions

Anonymous Transactions

- Encrypt a transaction (sender, receiver, amount)
- Prove that the unknown sender has the amount available
- Prove that the funds are not already spent

Anonymous Transactions

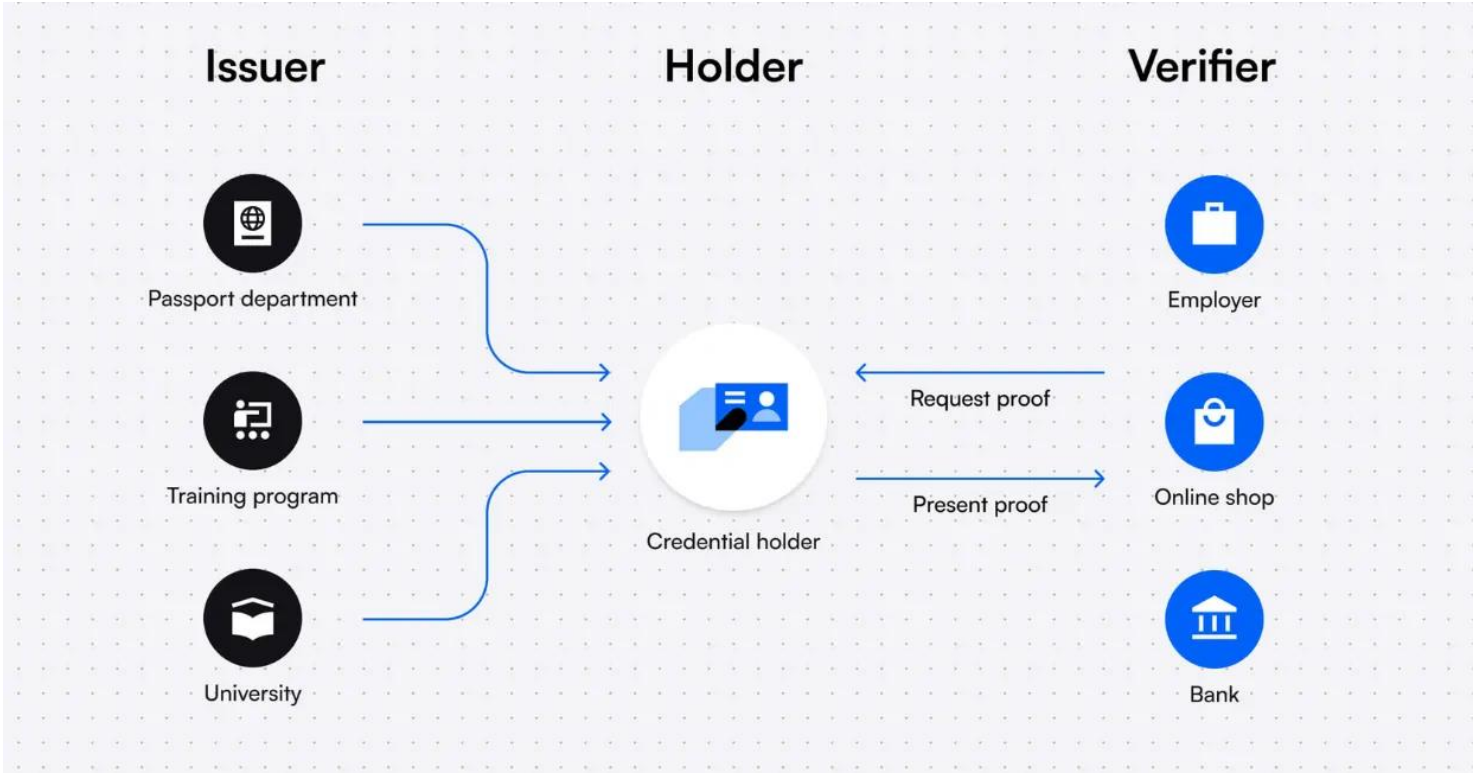


Anonymous Credentials

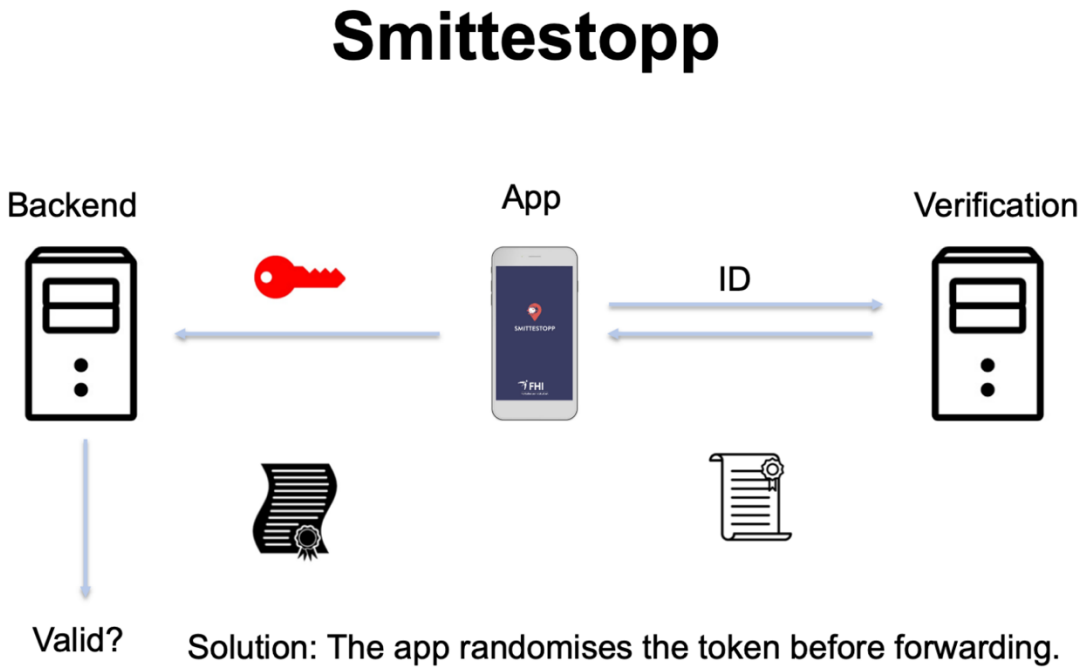
EU Digital Identity

Wallet

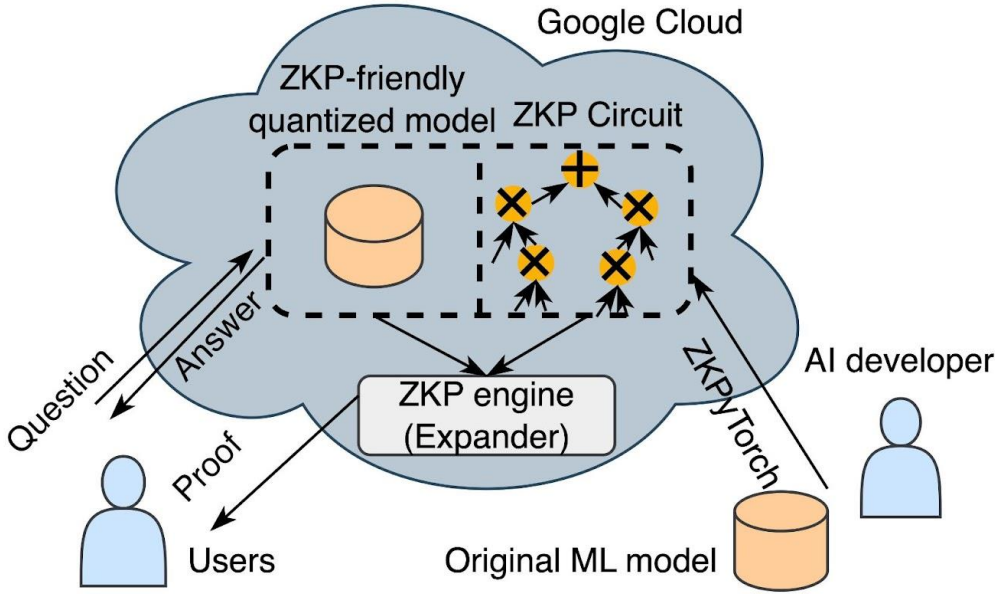
Anonymous Credentials



Anonymous Credentials



Verifiable Machine Learning



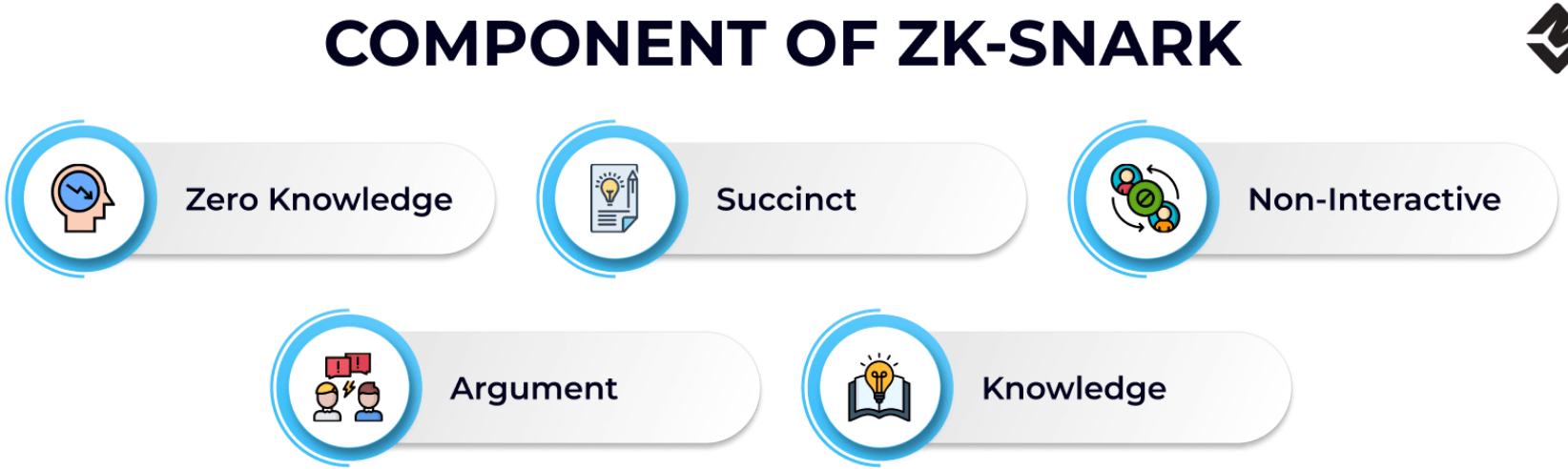
https://blog.polyhedra.network/zkpytorch

Verifiable Machine Learning

- Prove that a machine learning model was trained on a specific set of (potentially encrypted) data

- Prove that a machine learning model was evaluated on a specific set of (potentially encrypted) data

Succinct Arguments



Succinct Arguments

- Proofs are potentially much smaller than the secret itself (even logarithmic or constant size)
- Verification can be much faster than re-computation
- Puts a larger burden on the prover (time, memory)

Standardization Efforts

- NIST currently has a standardization process for ZKPs: https://csrc.nist.gov/projects/pec/zkproof
- Industry Consortia: ZKProof.org community, Ethereum Foundation, Hyperledger, W3C Verifiable Credentials
- There are many constructions available for interesting applications, but there few standards yet…

Open-Source Libraries (for edu)

It is not trivial to get started, but here are some projects:

- BabySNARK: https://github.com/initc3/babySNARK
- miniSTARK: https://github.com/andrewmilson/ministark
- Arkworks: https://github.com/arkworks-rs
- ZoKrates: https://zokrates.github.io

Privacy-Enhancing Cryptography

- Zero-Knowledge Proofs is a leading area within PEC.
- Fully Homomorphic Encryption (FHE) allows us to compute arbitrary functions on encrypted data, e.g., via outsourcing.
- Multi-Party Computation (MPC) allows for mutually distrusting parties to compute a function on party-private input data.
- ZKPs are often used to ensure that FHE and MPC is correct.

Thanks! Questions?

tjerand.silde@ntnu.no
https://tjerandsilde.no