Programmable Cryptography and Smart Contract

Shouki Tsuda

Researcher / Developer

What is the Programmable Cryptography?

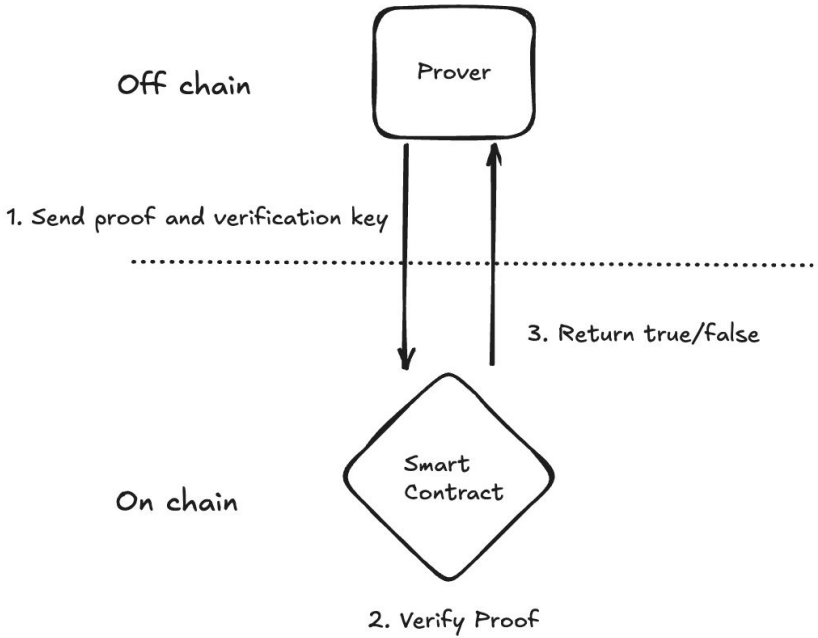
An emerging paradigm in cryptography that allows cryptographic systems to be designed, customized, and deployed in a modular like a software systems.

- Zero-knowledge proof
- Secure Computing
  - FHE
  - MPC
- ...etc

This section show how to run smart contracts based on off-chain execution results

Section 1

ZKP × Smart Contract



It is easy to interact with the smart contract

Solidity Verifier enables on-chain verification.

Another smart contract can be executed based on the verification results.

ZKP as a component of Verifiable Computation

ZKP + Secure Computing

= Verifiable Computation

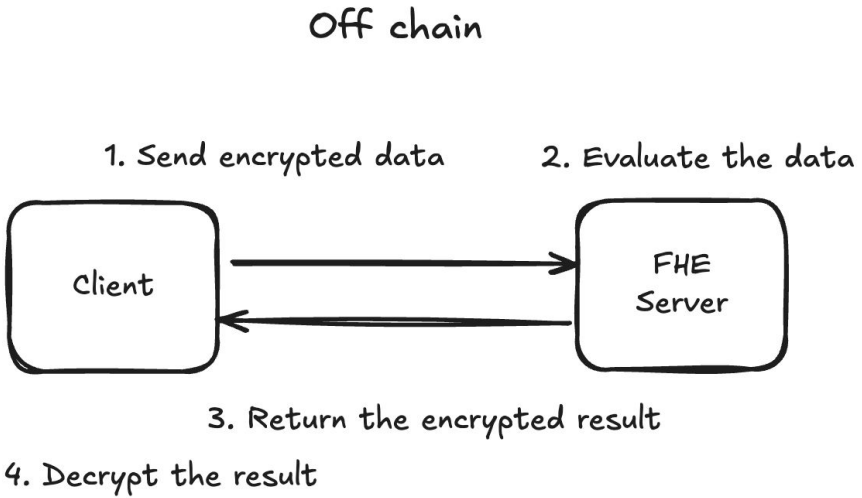
Although not expressive in itself, it can add verifiability to Secure Computing.

Section 2

FHE × Smart Contract

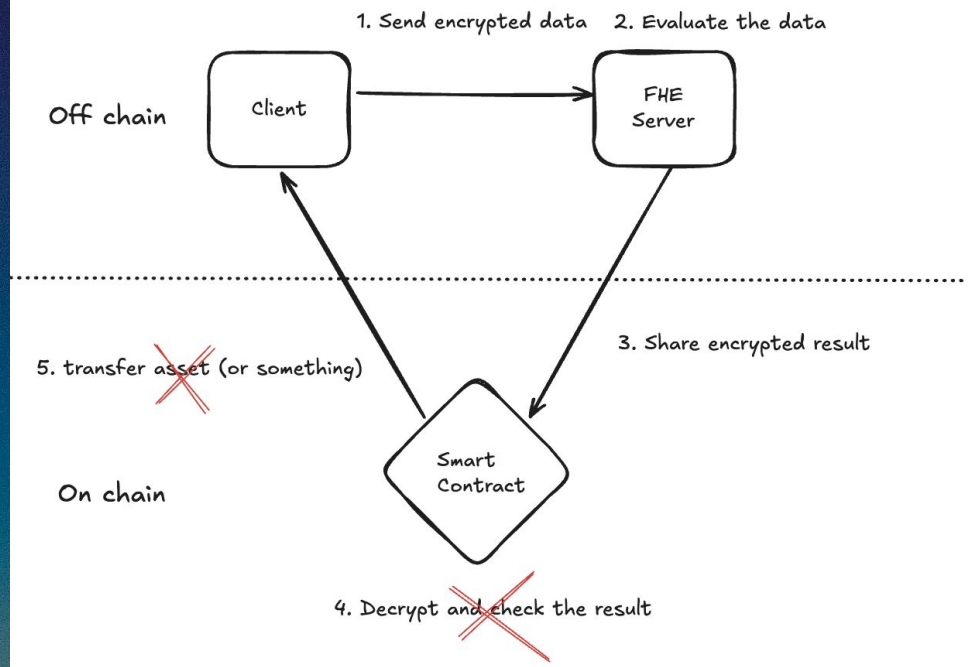
Basic principles of FHE

Data encrypted by the client can be evaluated in encrypted form on the server side.



Can a smart contract be executed based on the encrypted results?

Encrypted execution results cannot be decrypted by third parties other than the client



Solutions

Verifiable FHE

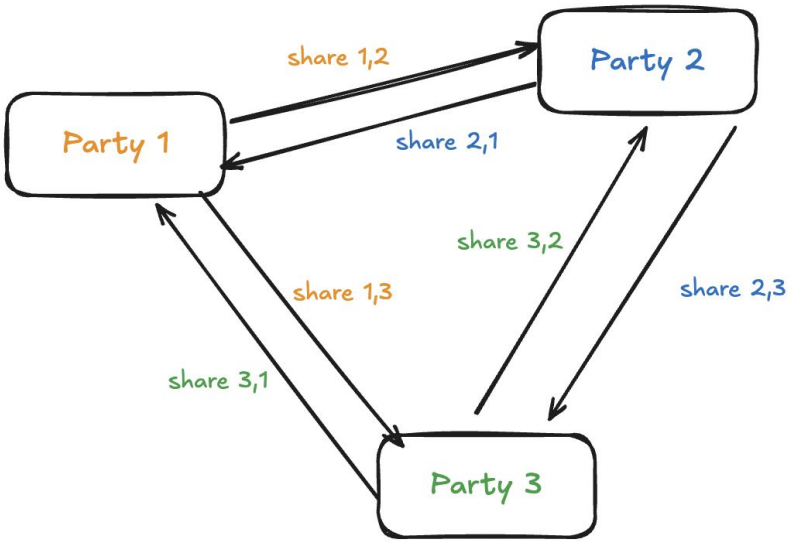
- ZKP × FHE
- ZKP enables on-chain verification of FHE execution
- Proof time is added to even the heaviest FHE execution
- [VKH23] [TW24]..etc

Multi-key FHE

- MPC × FHE
- The decryption key can be split and managed by a third party
- Dealer required
- [LTV12] [MW16]...etc

Section 3

MPC × Smart Contract



1. Split own confidential data into pieces
2. Share them with other participants
3. Evaluate some functions mutually
4. Share of evaluation results are passed to other parties
5. Each with their own final results

Basic principles of MPC

Some participants evaluate together and keep their data secret.

Basically, results are only available to MPC participants

Solutions

Verifiable MPC

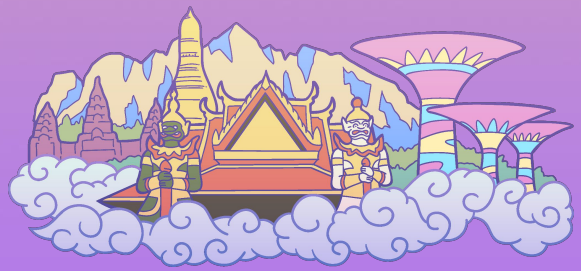
- ZKP × MPC
- Proof of MPC execution
- [Verifiable Encryption from MPC-in-the-Head]

Collaborative SNARKs

- ZKP × MPC
- MPC that generate proof collaboratively
- [Experimenting with Collaborative zk-SNARKs]

Section 4

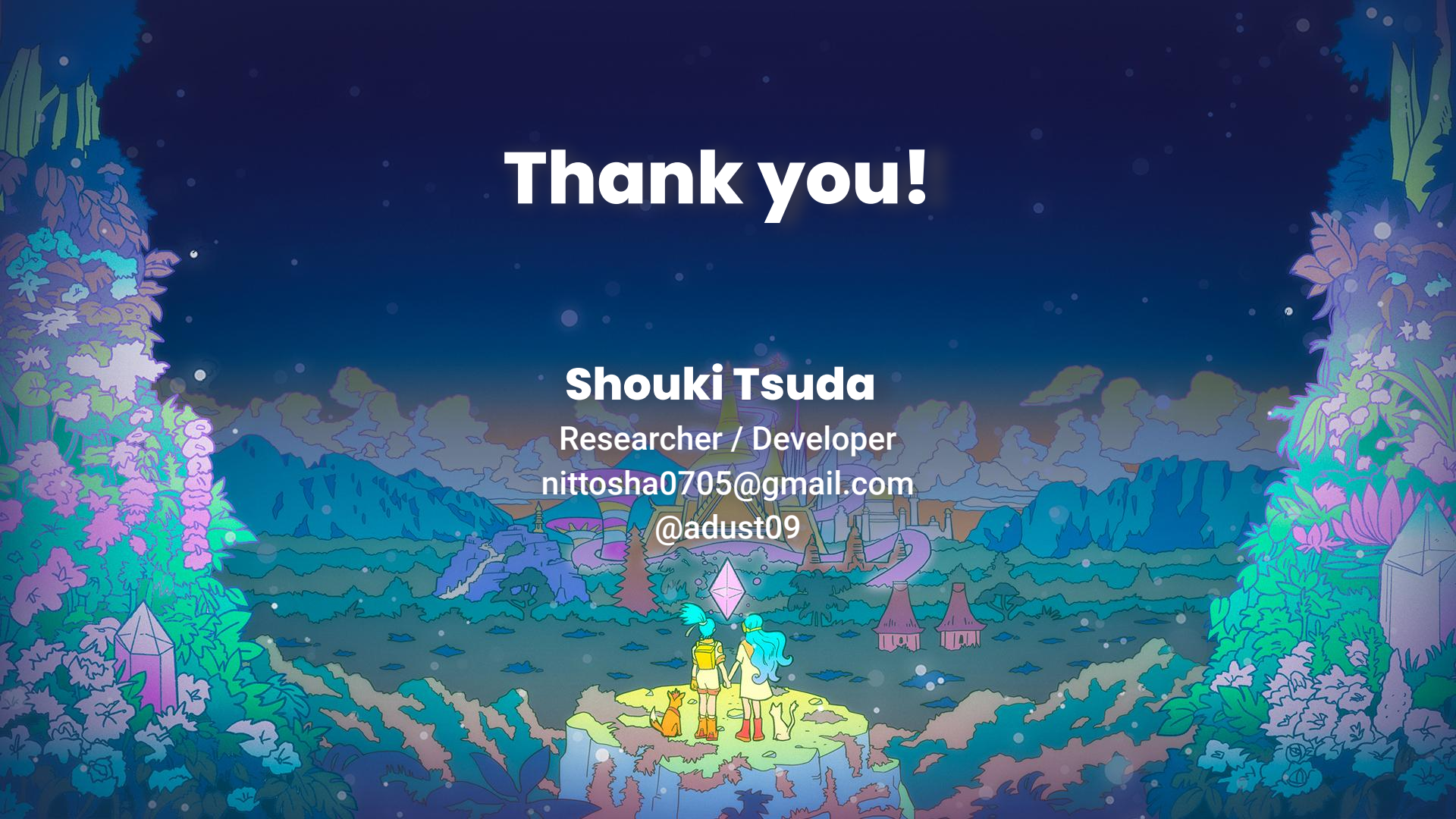
Conclusion



Conclusion

- ZKP has on-chain verifiability but low expressive power
- Secure Computing is expressive but does not have on-chain verifiability
- Basically, Secure Computing cannot execute smart contracts based on execution results, but can be combined with ZKP

Thank you!



Shouki Tsuda

Researcher / Developer

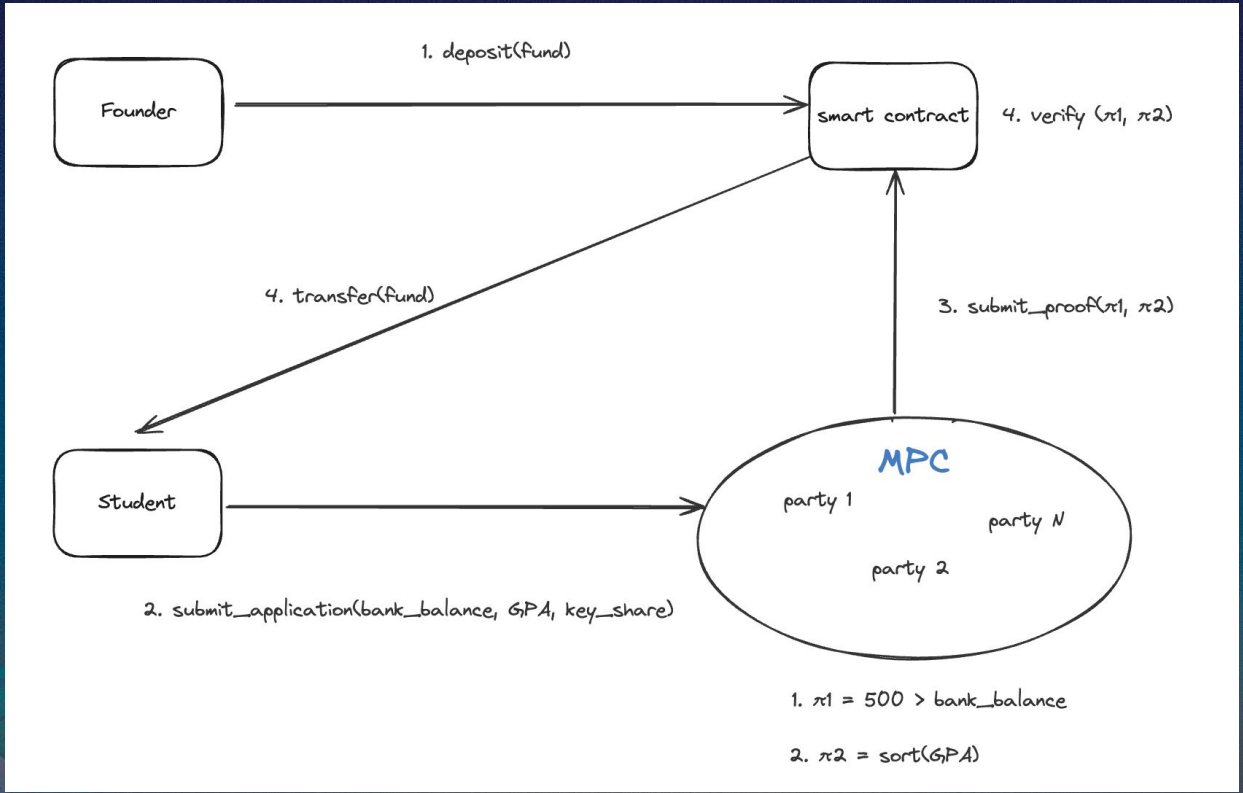
nittosha0705@gmail.com

@adust09

Comparison

Primitive	Expressiveness	On-Chain Verification	Concept
ZKP	Low	Yes	Proving the validity of claims
FHE	High	No	Evaluate on encrypted data
MPC	High	No	Participants collaborate on calculations confidently
Verifiable FHE	High	Yes	Proving computation of FHE
Multi-key FHE	High	No	Splitting description key
Verifiable MPC	High	Yes	Proving computation of MPC
Collaborative SNARK	High	Yes	Participants collaborate on proofs

Example


1. deposit(fund)
Founder
smart contract
4. verify (π1, π2)
4. transfer(fund)
3. submit_proof(π1, π2)
MPC
party 1
party N
party 2
Student
2. submit_application(bank_balance, GPA, key_share)
1. π1 = 500 > bank_balance
2. π2 = sort(GPA)