OwnFace：Own Your Face Before Own Your Data

1 Why

Own Your Face Before Own Your Data

2 How

人脸识别:

$$d(\mathbf{f}^{(0)}, \mathbf{f}^{(1)}) = \sum_{i=1}^{N} (f_i^{(0)})^2 + (f_i^{(1)})^2 - 2 f_i^{(0)} f_i^{(1)} < \epsilon$$

佩德森承诺:

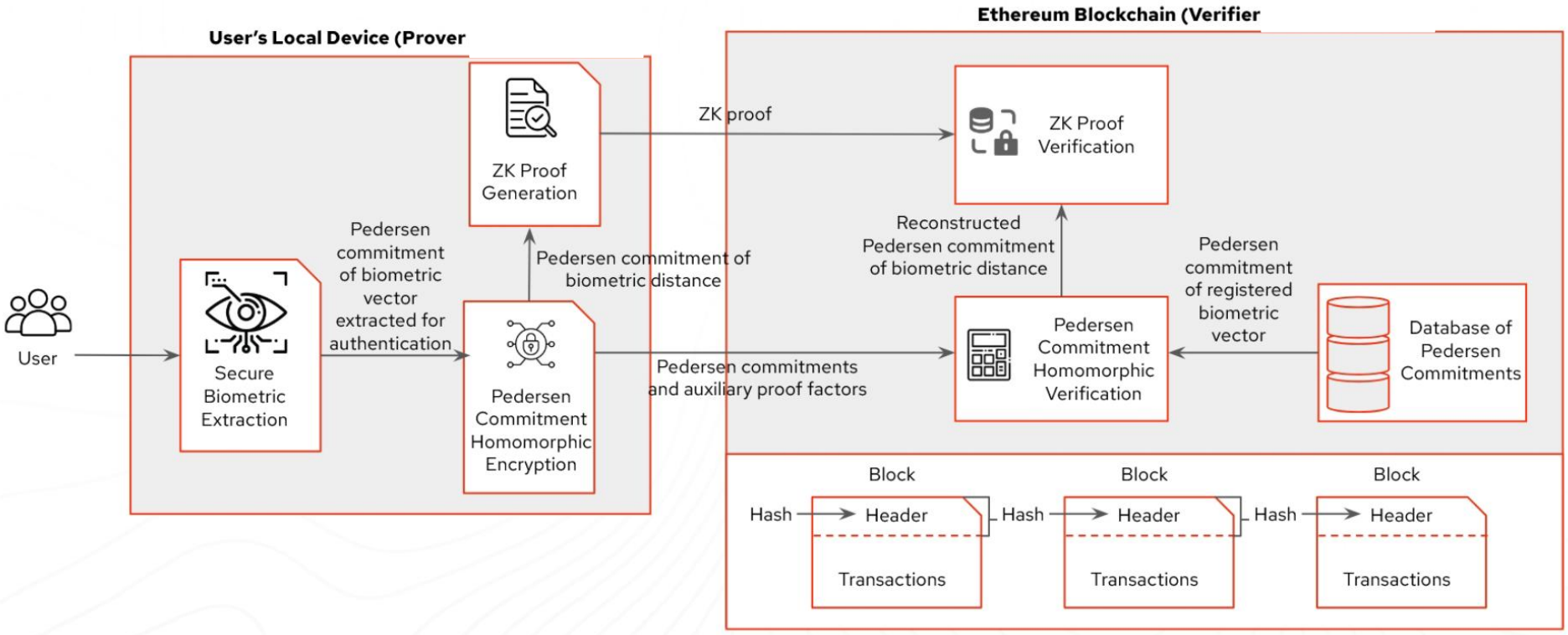
$$c = c_{g,h}(f, r) = g^f h^r \ mod \ p$$

佩德森承诺同态性:

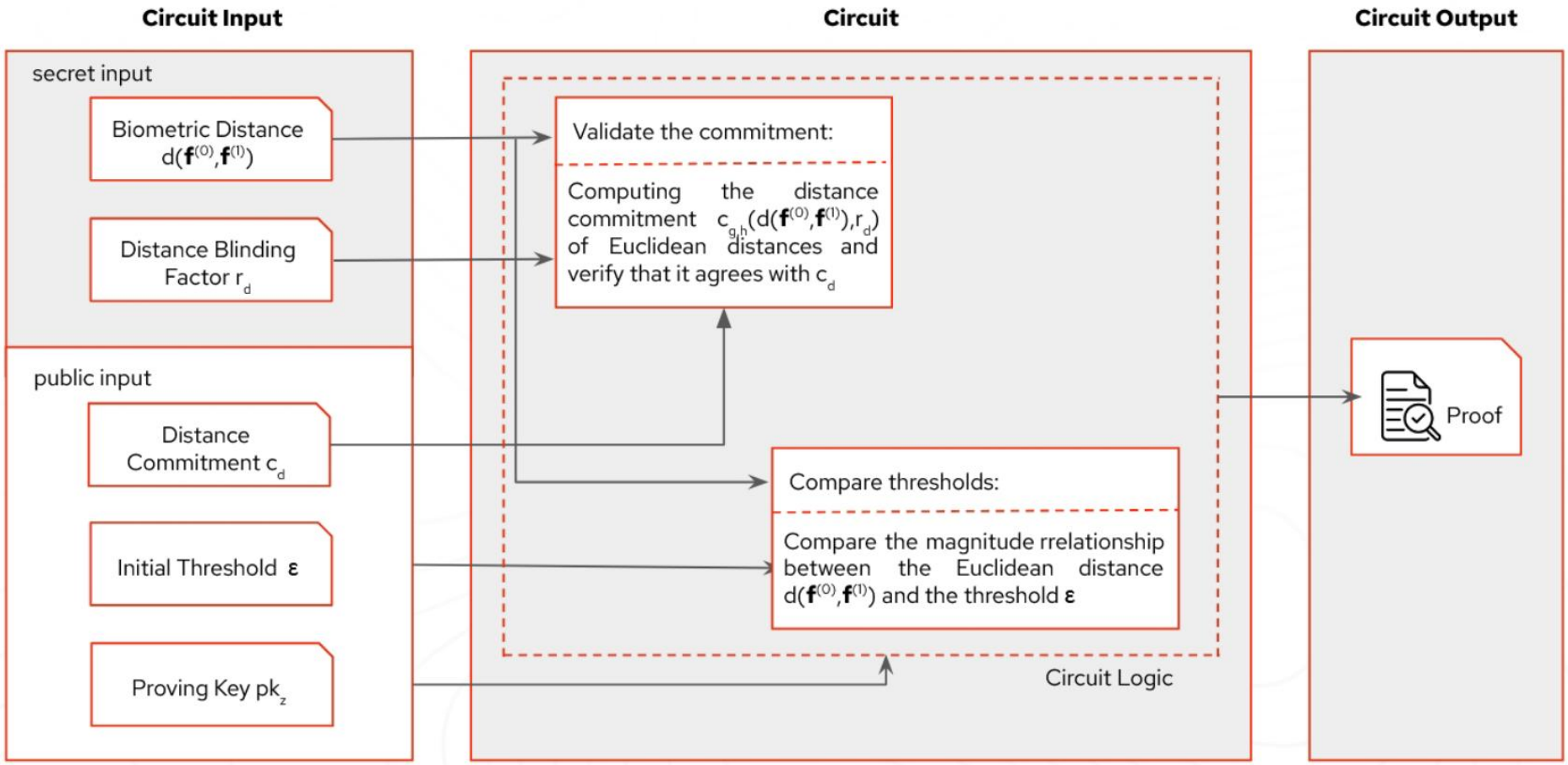
$$\begin{aligned} & c_{g,h}(f^{(0)}, r^{(0)}) \oplus c_{g,h}(f^{(1)}, r^{(1)}) \\ &= c_{g,h}(f^{(0)}, r^{(0)})\, c_{g,h}(f^{(1)}, r^{(1)}) \\ &= g^{f^{(0)}} h^{r^{(0)}} g^{f^{(1)}} h^{r^{(1)}} \ mod \ p \\ &= c_{g,h}(f^{(0)} + f^{(1)}, r^{(0)} + r^{(1)}) \end{aligned}$$

$$\begin{aligned} & c_{g,h}(d(\mathbf{f}^{(0)}, \mathbf{f}^{(1)}), r_d) \\ &= c_{g,h}(\sum_{i=1}^{N} (f_i^{(0)})^2 + (f_i^{(1)})^2 - 2 f_i^{(0)} f_i^{(1)}, r_d) = \\ & \sum_{i=1}^{N} c_{g,h}((f_i^{(0)})^2, r_d) \oplus c_{g,h}((f_i^{(1)})^2, r_d) \odot 2 c_{g,h}(f_i^{(0)} f_i^{(1)}, r_d) \end{aligned}$$

2 How


User's Local Device (Prover
Ethereum Blockchain (Verifier
User
Secure Biometric Extraction
Pedersen commitment of biometric vector extracted for authentication
Pedersen Commitment Homomorphic Encryption
ZK Proof Generation
Pedersen commitment of biometric distance
ZK proof
Pedersen commitments and auxiliary proof factors
ZK Proof Verification
Reconstructed Pedersen commitment of biometric distance
Pedersen Commitment Homomorphic Verification
Pedersen commitment of registered biometric vector
Database of Pedersen Commitments
Block
Hash
Header
Transactions
Block
Hash
Header
Transactions
Block
Hash
Header
Transactions

2 How


Circuit Input
Circuit
Circuit Output
secret input
Biometric Distance $d(\mathbf{f}^{(0)},\mathbf{f}^{(1)})$
Distance Blinding Factor r_d
public input
Distance Commitment c_d
Initial Threshold ε
Proving Key pk_z
Validate the commitment:
Computing the distance commitment $c_{g,h}(d(\mathbf{f}^{(0)},\mathbf{f}^{(1)}),r_d)$ of Euclidean distances and verify that it agrees with c_d
Compare thresholds:
Compare the magnitude rrelationship between the Euclidean distance $d(\mathbf{f}^{(0)},\mathbf{f}^{(1)})$ and the threshold ε
Circuit Logic
Proof

3 What

4 Future

模块 Module	技术栈 Stack	职责 Responsibility
`frontend`	Next.js 15, React 19, RainbowKit, wagmi, viem, Tailwind	注册/认证 UI、摄像头采集、钱包交互、交易状态与证明展示
`backend`	Node.js 20+, Express, TypeScript, snarkjs, circomlibjs, zod	嵌入量化、Poseidon 哈希、Pedersen 承诺、Groth16 生成与验签、REST API
`contract`	Solidity 0.8.24, Hardhat, TypeScript scripts	`OwnFaceRegistry`（承诺存储与认证记录）、`Groth16Verifier`（距离电路验证）
circuits	Circom 2, snarkjs	`distance.circom`、`.wasm`、`.zkey`、`verification_key.json`

本地客户端

可信后端（TEE。。。）

5 Finally

十年之后，儿子问你什么是 **Web3**

谢谢各位聆听！