IDEAS NCBR

Zero Knowledge Proofs

and Their Applications in Cyber Physical Systems

Shahriar Ebrahimi

IDEAS NCBR

shahriar.ebrahimi@ideas ncbr.pl

sh.ebrahimi92@gmail.com

20th Nov. 2023

Presentation Flow

Background	• Fundamental concepts, Different proving systems
Case-study	• Tornado-cash, Proof of Reserve, Wormholes
Practical ZKP in IoT	• Privacy, Scalability, Integrity
zRA	• Transparent Remote Attestation based on zkSNARKs, NDSS `24
Challenges & Considerations	• Practical ZKP in IoT
Future Directions	• Advancements in ZKP, Role of energy and efficiency

Take away

- Some AI-generated anime-style images[1] ☺
- Concept of proving systems
  - ZK point of view
    - Secure verification of a problem
    - not the solution itself
- VC ≠ ZKP
- Challenges and possible research areas
  - Intersection of ZKP/VC and CPS/IoT
- Some great starting points for ZKP and VC

[1]All of the anime images are generated by Bing Image Creator

Proving Systems

Definition:

- Cryptographic protocols,
  that enable one party (the prover),
  to prove to another party (the verifier),
  that a given statement is true.
- [*without revealing any information about the statement itself*].

Principles:

- **Completeness:**
  honest prover can convince all verifiers.
- **Soundness:**
  a dishonest prover cannot convince any verifier.
- Optional [***Zero-knowledge***]:
  the verifier learns nothing beyond the validity of the statement.

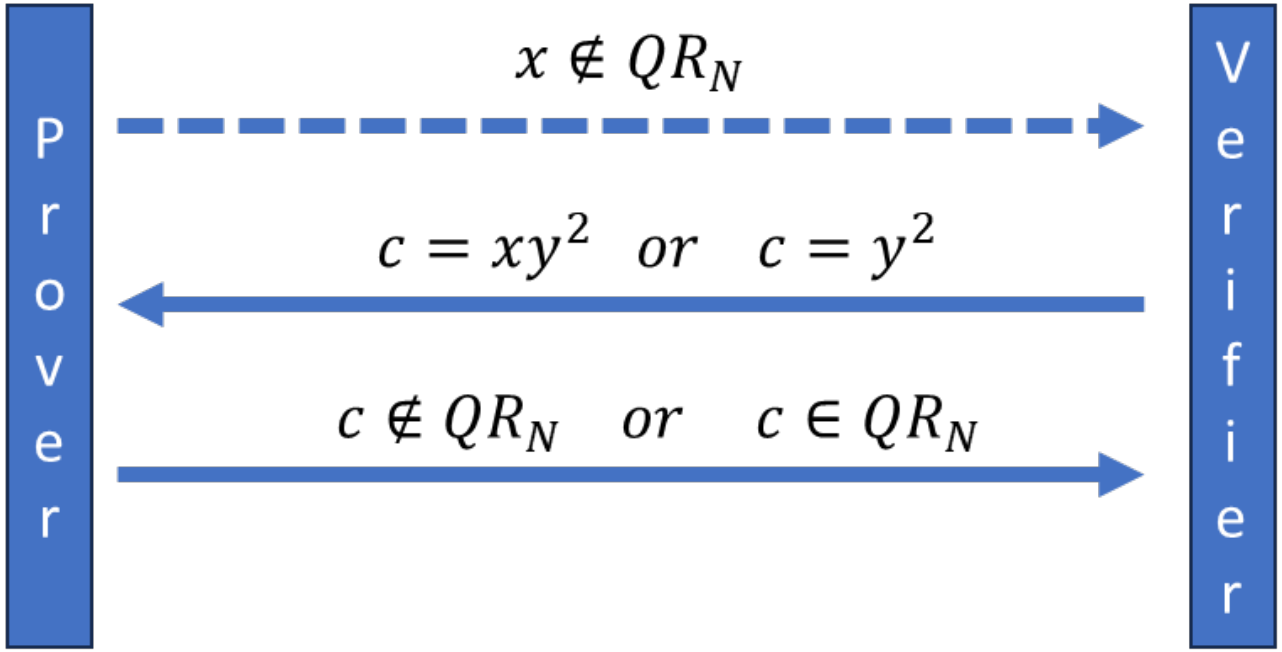
Types of Proofs

- Interactive Proofs
  - Series of interactions between the prover and the verifier
    - Challenge --> Response
  - Each round increases the confidence in the validity of the proof
  - Dishonest Prover's success probability = $^{1}/_{c^{n}}$, for n rounds.

- Non-Interactive Proofs
  - Usually are publicly verifiable.
  - Require more complex prover [and verifier].
  - Examples
    - SNARKs
    - STARKs

Interactive Proof: Quadratic Non-Residuosity Example

- Problem: $QR_N = \{\forall a \in \mathbb{Z}_N | a \equiv b^2 \ mod \ N\}$
- Claim: I know x, such that $x \notin QR_N$ (or $x \in \overline{QR_N}$)

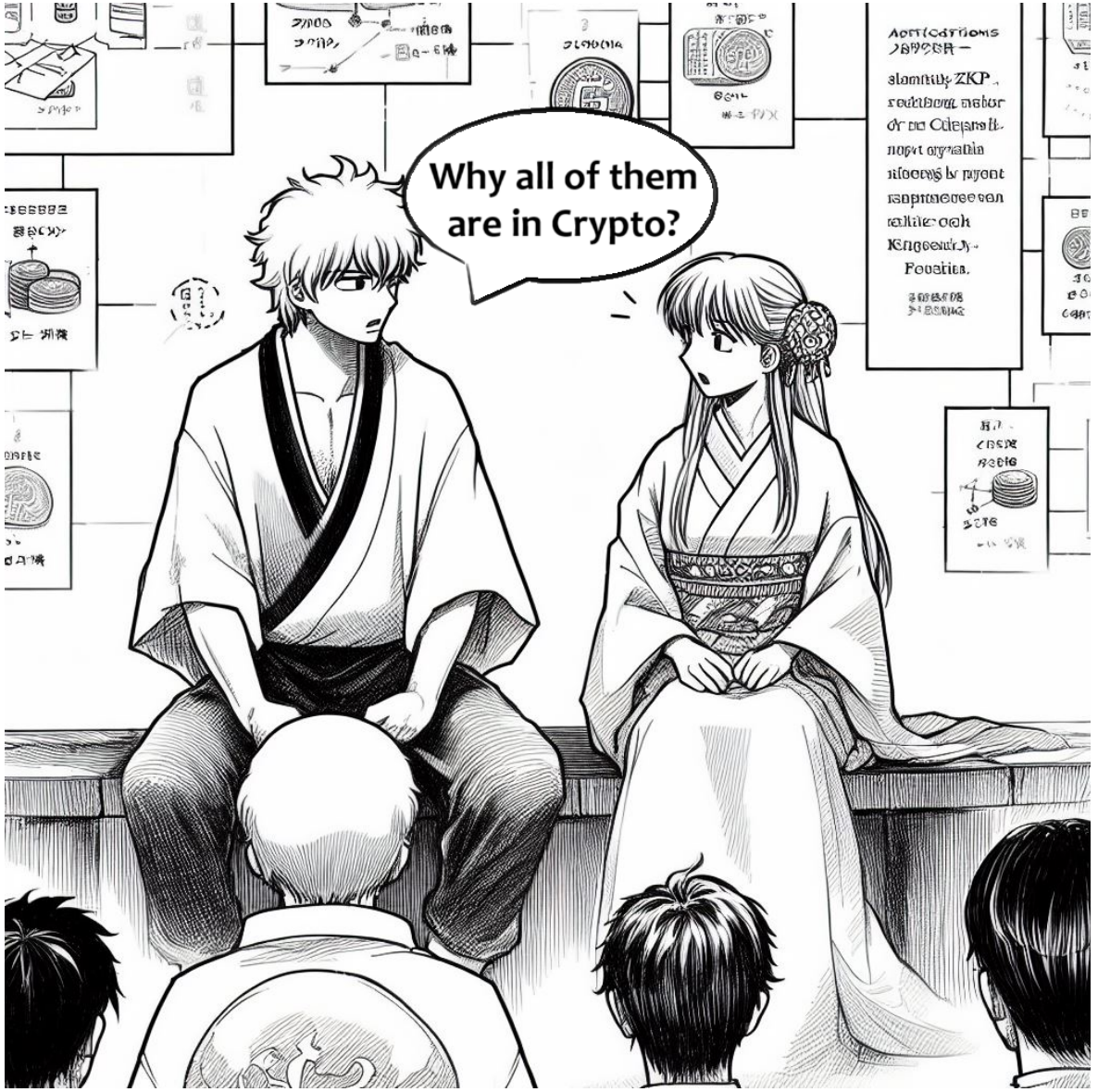


Non-Interactive Proof: Hash-based Commitments Example

1) Implement the hash algorithm in a VC framework
   - VC: Verifiable Computation
   - SNARKs, STARKs, Bulletproofs, etc.
2) Commit to the hash result
3) Generate witness ω: execute VC program with proper inputs
4) Generate proof π: using the witness ω
5) Send the proof π to the verifier [*or broadcast to network*]
6) Verify the proof π

Case-study

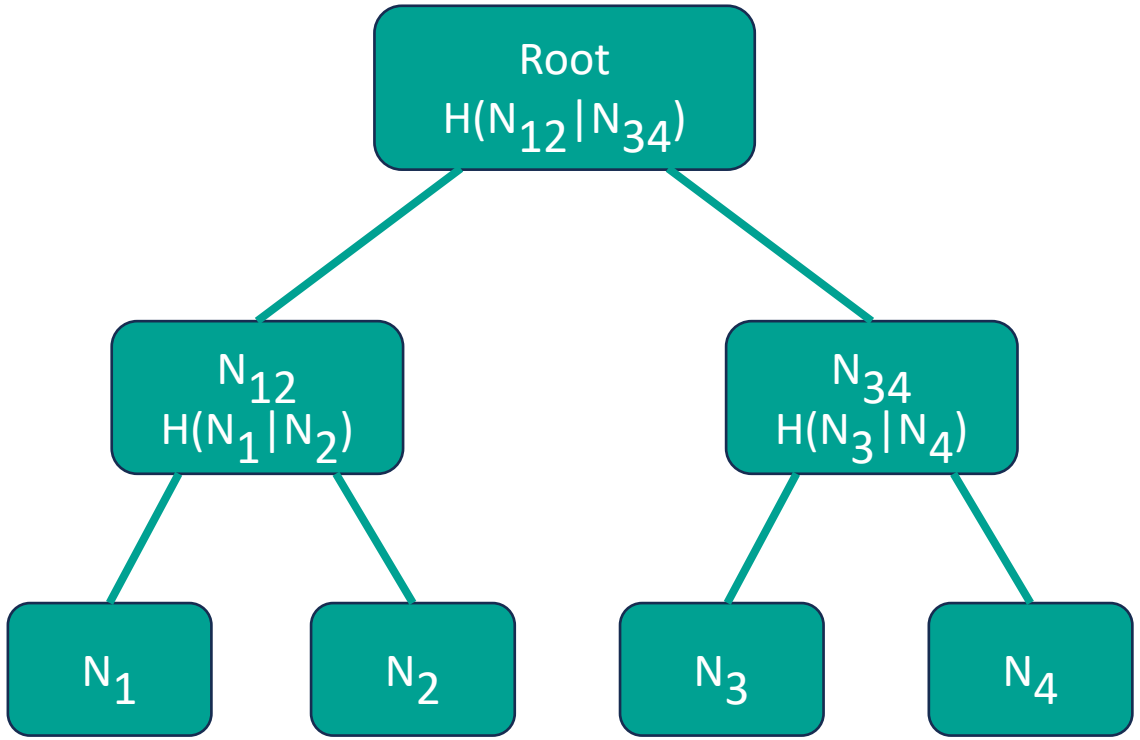
- Tornado-cash[1]
  - Privacy-preserving transactions
    - A commitment scheme
    - proofs of inclusion in a Merkle tree
- Proof of Reserve
  - Prove owning some currency on-chain
  - Without revealing the exact amount or owned addresses
- Wormholes[2] (i.e. proof of burn)
  - Proofs of burning certain amounts of a currency
  - Mintable on the same chain [*or other chains --> trustless bridge*]

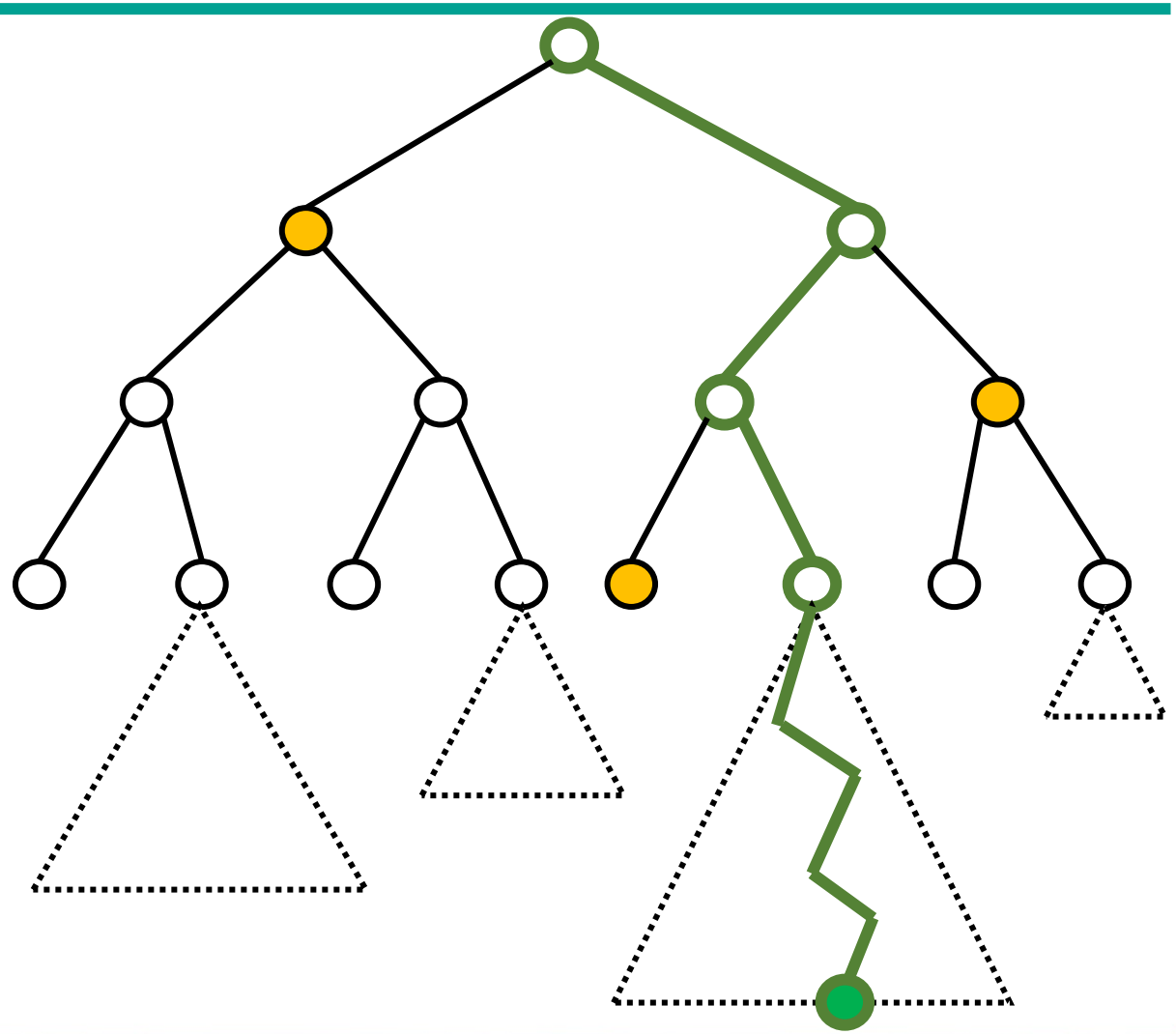


[1]Pertsev, A. et. al. (2019). Tornado Cash Privacy Solution Version 1.4.

[2] https://eip7503.org

Merkle tree & Merkle Path





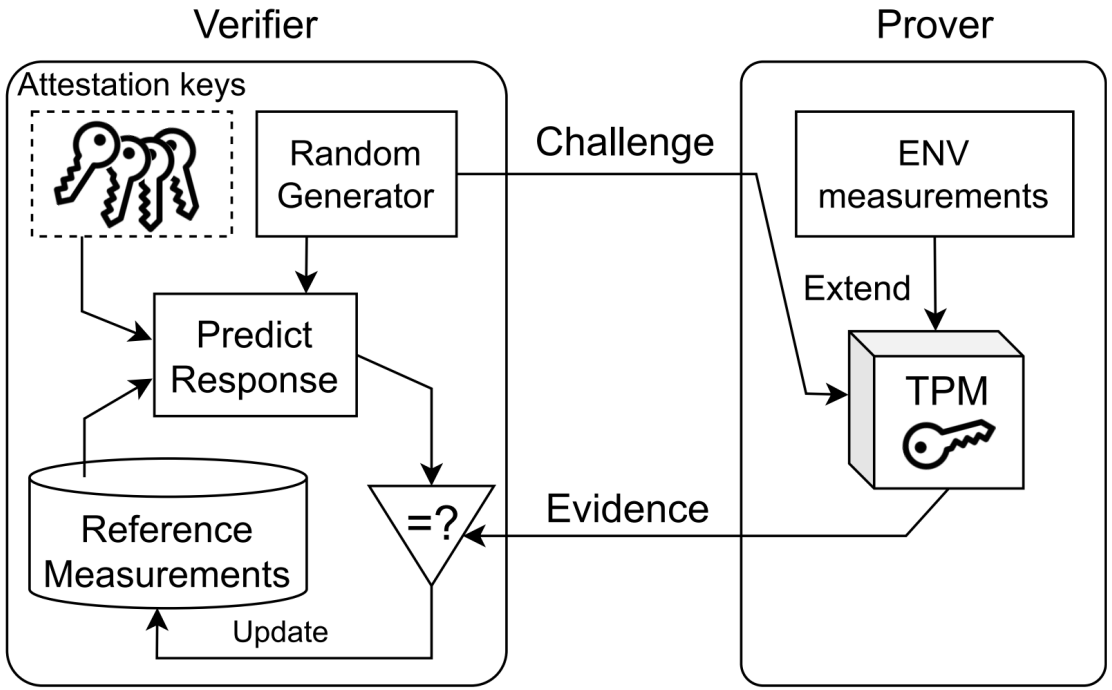
Tornado-cash

- Merkle tree
  - Commitments of deposits
  - Nullifier
    - Prevent double-spend
- Deposit
  - Commit to a pre-image
  - Pedersen hash commitments
- Withdraw
  - Prove: knowledge of a pre-image for a commitment in the tree
    - That the nullifier is not disabled
    - i.e. not spent yet

ZKP for CPS

- Privacy Preservation
  - Identity Protection
  - Less needed Secure Transactions
- Data Integrity Assurance
- Trustless authentication
- Identity Protection
- Reduced Computational Overhead
- Efficient Data Transmission

Remote attestation (RA)

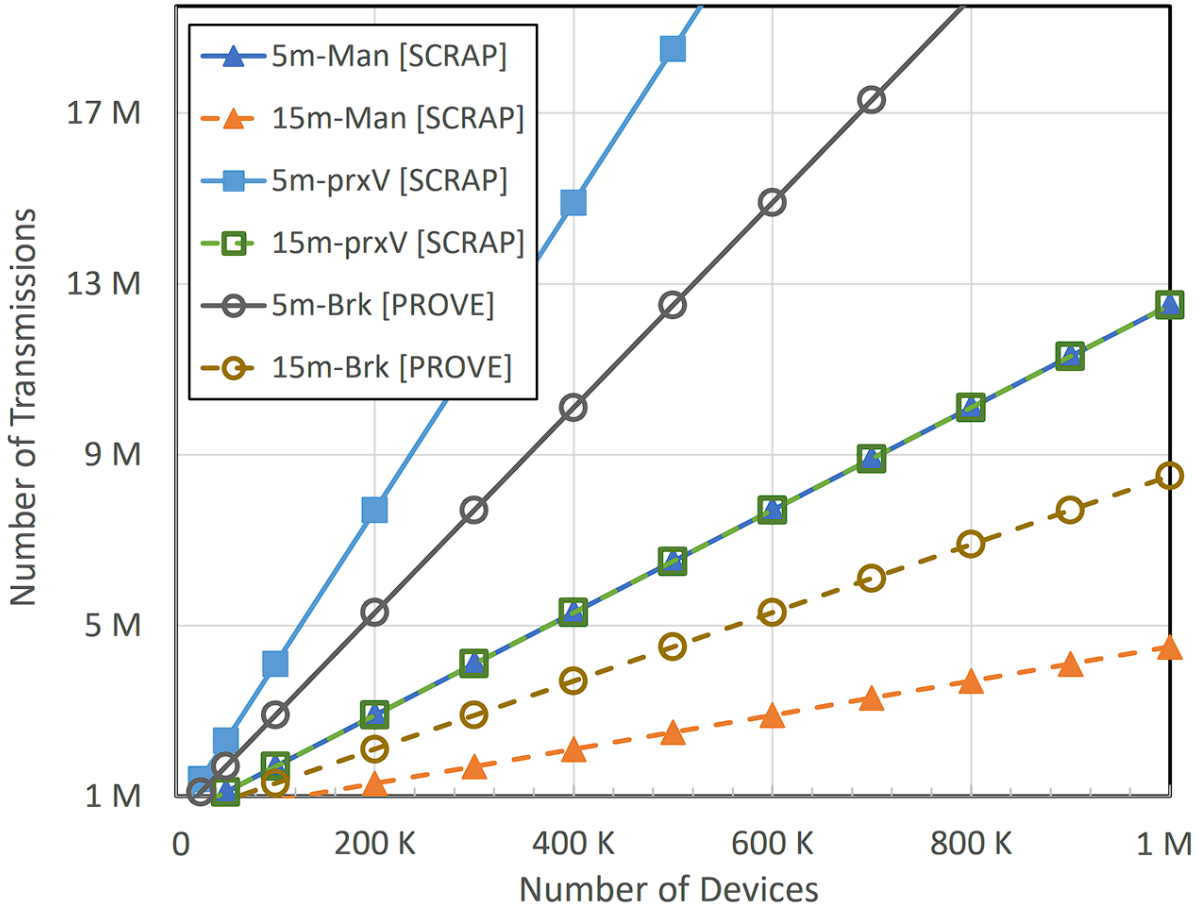


What's wrong with State-of-the-art RA?

- Verifier needs privileged access
- Device must authenticate the verifier
- Everyone must trust the verifier
- Scalability
  - Bounded by the trusted authorities
  - How many trusted parties can we have?
- Availability
  - Availability of the verifier
  - the connection between verifier and prover
- Denial of Service (DoS) attack surface

Scalability? But it is just one tiny message

- Number of requests per hour transmitted by trusted parties
- Scenario
  - attest every 5 or 15 minutes
- [PROVE]
  - Broker takes around 8 seconds to handle 1000 requests
  - 150 attestations per second
  - Theoretical upper bound
    - 100,000 devices



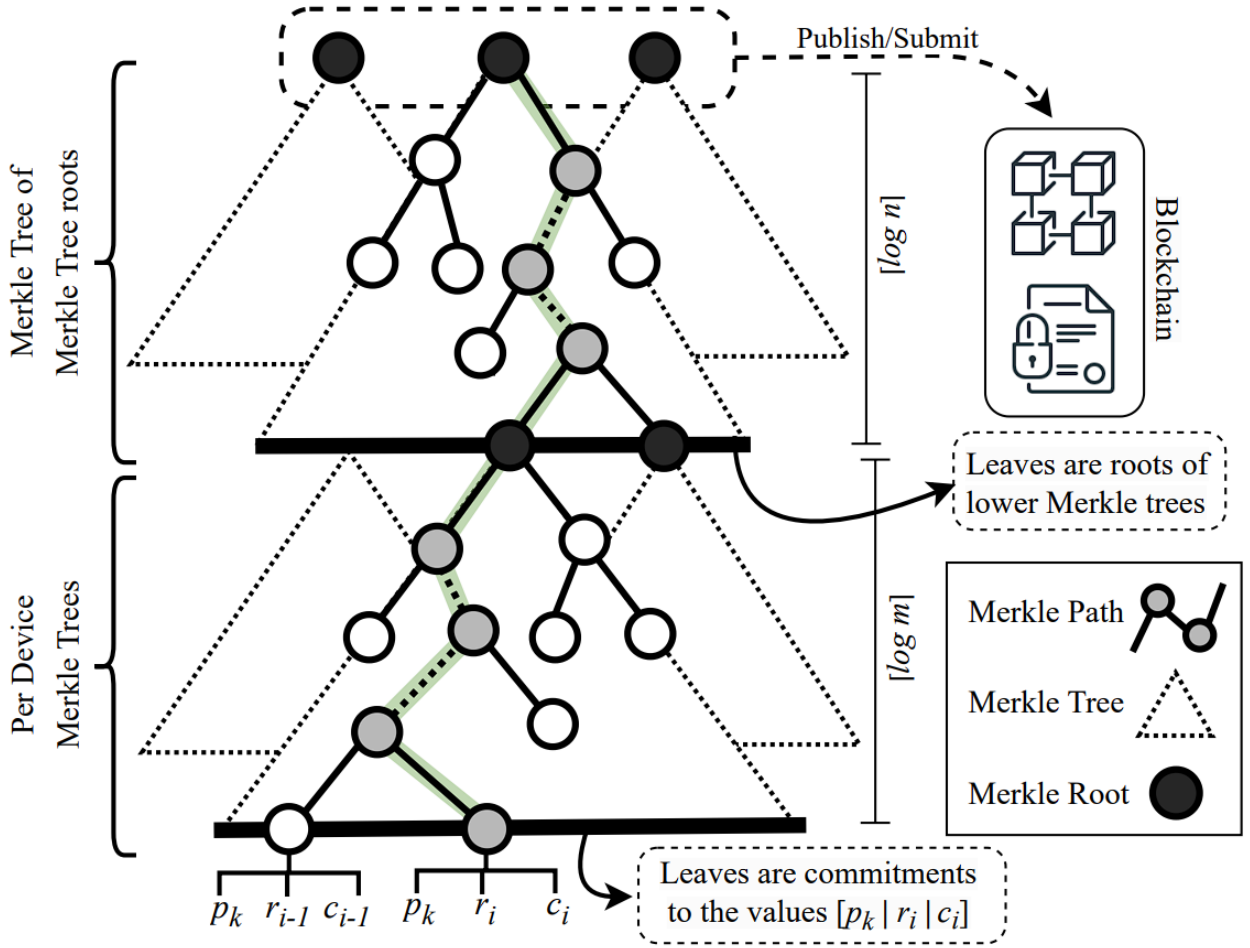
[SCRAP] Petzi, et. al. (2022). SCRAPS: Scalable Collective RA for Pub-Sub IoT Networks with Untrusted Proxy Verifier. USENIX Security 22 (pp. 3485-3501).

[PROVE] Dushku, et. al. (2023). PROVE: Provable remote attestation for public verifiability. Journal of Information Security and Applications, 75, 103448.

IDEAS NCBR

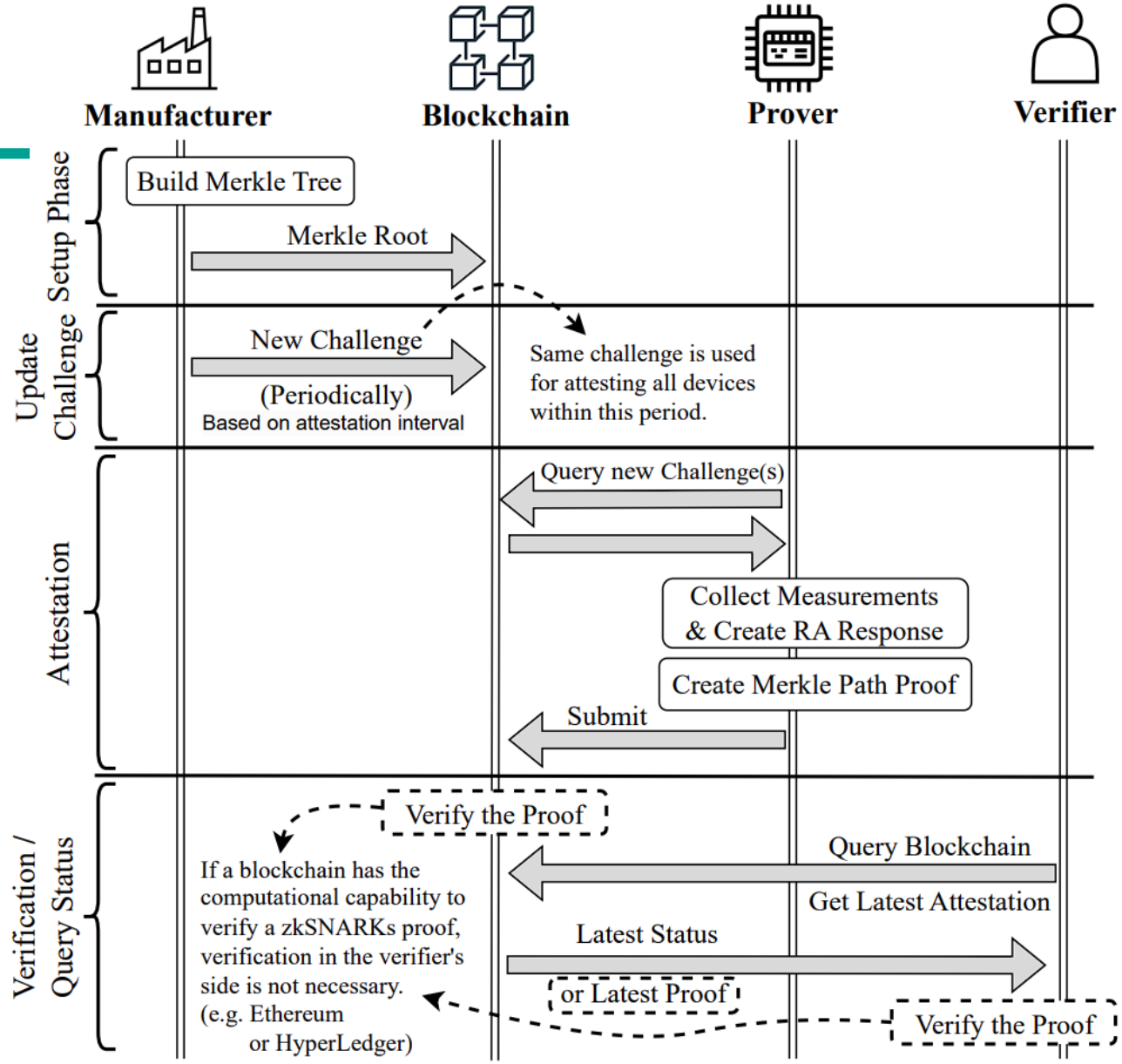
zRA: Transparent & Non-Interactive RA

- Commitments of future responses
  - $H^3_{pos}(p_k \mid r_i \mid c_i)$
- Global challenges
  - Easy to maintain, but . . . [will discuss]
  - Periodically publish one global challenge
- Proves of knowledge to a pre-image
  - A leaf of the public Merkle tree
  - Public inputs of the ZK Circuit
    - c_i => check freshness
    - p_k => verify authenticity
    - $root$ => attestation
- Easily updatable
  - Extend, update, revoke, . . .
  - Just update the Merkle root ☺



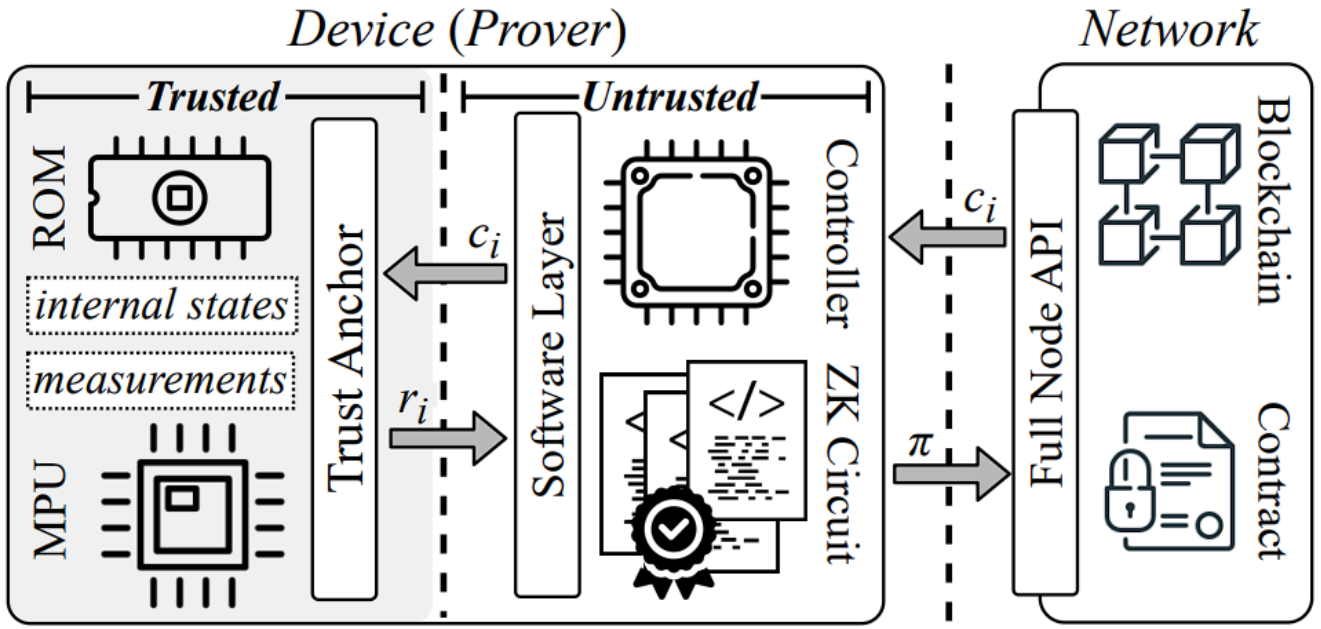
zRA Protocol

- Manufacturer
  - Commits to future responses
  - $H^3_{pos}(p_k \mid r_i \mid c_i)$
  - Global challenges
    - Easy to maintain, but . . . [will discuss]
  - Periodically publish one global challenge
- Device
  - Query latest challenge
  - Generate response [trust anchor]
  - Prove knowledge of pre-image to a leaf in the Merkle tree
- Verifier
  - Everyone can verify the proof
  - No prior knowledge is needed



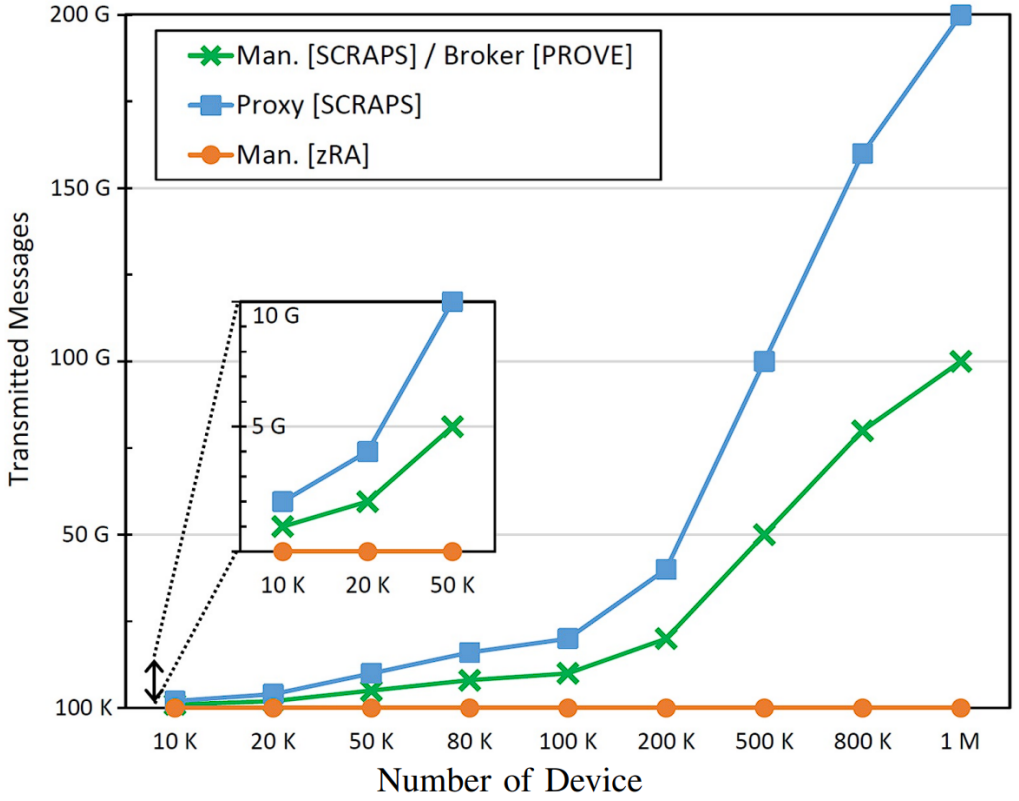
Infrastructural requirements of zRA

- zRA demands minimal infrastructure to achieve secure attestation
- Prover side
  - Minimum trust anchor
    - MPU + ROM
  - Independent from GenRes()
  - Asynchronous attestation
- Verifier side
  - Nothing required ☺
    - No knowledge of device
    - No trusted party
    - No blockchain



Performance of zRA vs. S-o-t-A

	Prover			Prx.Vrf. [2] Broker [10]	Verifier
	Device	Time	Energy		
SCRAPS [2]	Cortex M-33●	1.07 s	N/A	55.4 ms	-
PROVE [10]	Virtex-7■	4.6 ms	N/A	≈7 ms	-
zRA	Core™ i5✪	0.6 s	479 mJ▲	-	<1 ms
	Cortex-A53✣	21.8 s	14.46 J♥		
	Cortex-A17★	11.9 s	53.08 J♥		

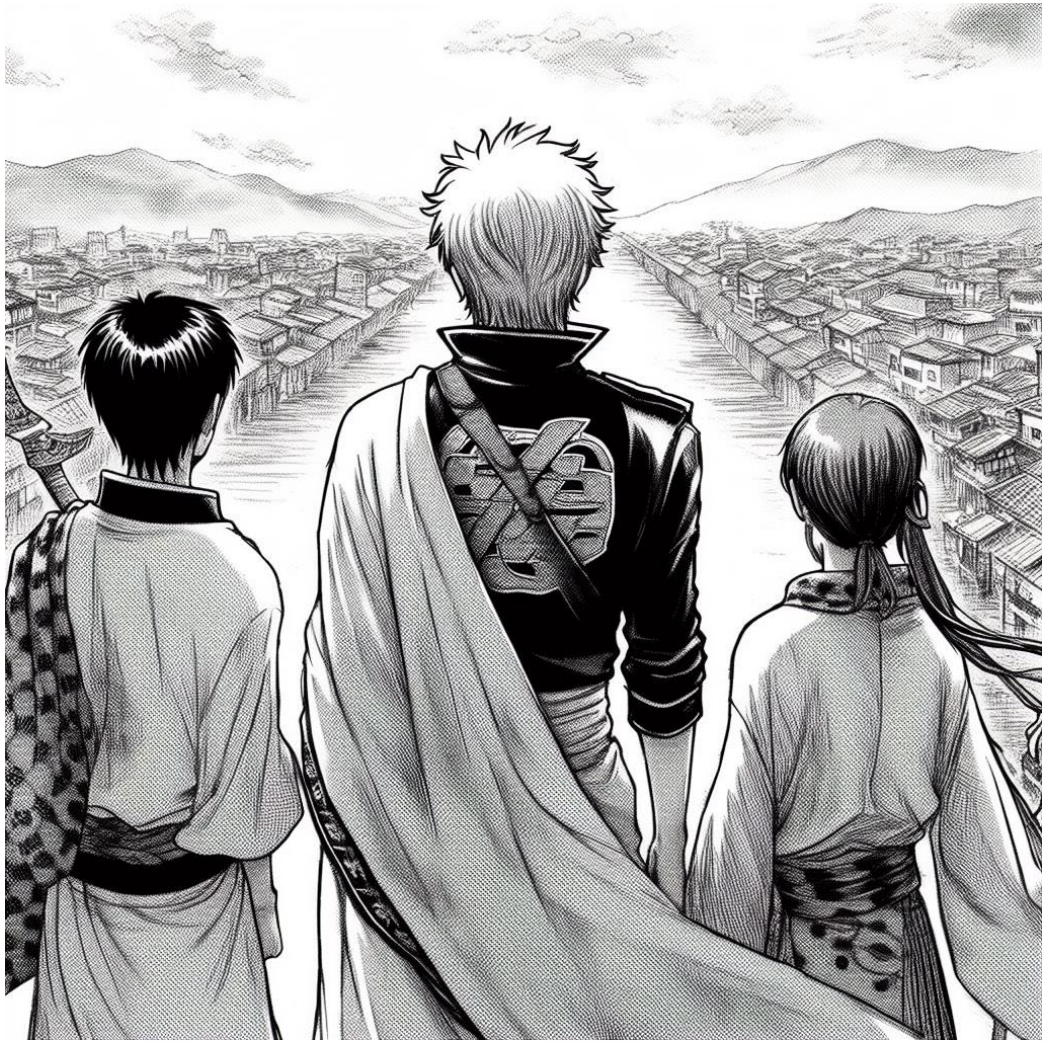


Challenges

- Computation complexity of ZKP schemes
  - Especially in prover side
  - High energy consumption
  - Not very efficient
- Communication complexity
  - Proofs can be large
  - More public inputs => larger proof
  - Some ZKP schemes require trusted setup

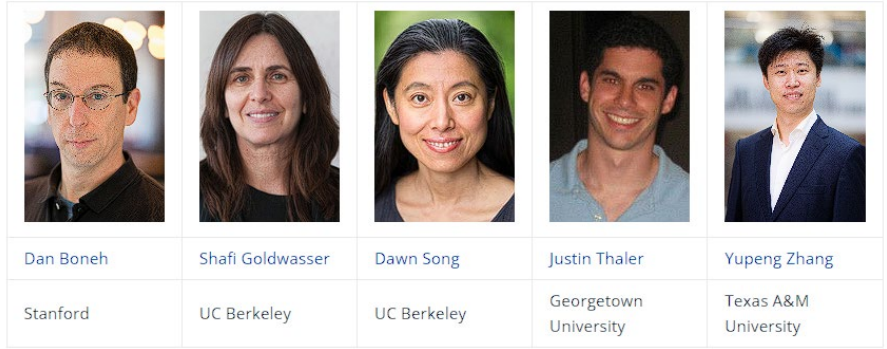
Current and Future Research Areas

- Efficient implementations of ZKP
  - Especially, the prover
  - Both on HW and SW
- Communication-intensive applications
  - Can exponentially reduce communications
- Delegation of proof generation
  - Edge-computing
  - Trustless protocols
- Smart cities
  - Smart grid
  - Connected vehicles



ZKP links

- Compilers
  - **Circom** --> https://iden3.io/circom --> very good documentation & libraries
  - Noir --> https://aztec.network/aztec-nr/
  - Halo2 --> https://docs.axiom.xyz/zero-knowledge-proofs/getting-started-with-halo2
- In-depth Courses
  - https://zk-learning.org



Dan Boneh	Shafi Goldwasser	Dawn Song	Justin Thaler	Yupeng Zhang
Stanford	UC Berkeley	UC Berkeley	Georgetown University	Texas A&M University

  - https://www.youtube.com/@blockchain-web3moocs635

Any Questions?

IDEAS NCBR