BARRETT: BlockchAin Regulated REmote aTTestation

Michail Bampatsikos, Christoforos Ntantogian, Christos Xenakis, Stelios C. A. Thomopoulos

ACKNOWLEDGEMENT: The research of the authors M. Bampatsikos and S. C. A. Thomopoulos is supported by Stavros Niarchos Foundation (SNF) in conjunction with EXODUS Ltd, under Grant No. 12149 "Support of scholarships for industrial PhD's and post-doc industrial positions and adjunct industrial researcher" whereas the research of the authors C. Xenakis and C. Ntantogian in is supported by the EU as part of the CUREX project (H2020-SC1-FA-DTS-2018-1 under grant agreement No 826404)

Overview

- Problem Statement
- Our proposal
- Related work
- Our contribution
- Assumptions and threat model
- BARRETT Architecture
- Data flow
- Performance Analysis
- Security Analysis
- Conclusions and Future work

Introduction – Problem Statement

- Today an increasing number of IoT healthcare devices connects to the Internet.
- They have low computational power; thus limited self-defense capabilities.
- One way to protect them is Remote Attestation (RA).
- However RA protocols can be used to attack these devices via:
  a) **Computational DoS (CDoS) attacks**
  b) **Deliberate Network Congestion**
- What do we do?

Our proposal: The BARRETT architecture

- Force the verifiers pay a fee for every Attestation Request (AR).
- CDoS and network congestion attacks become prohibitively expensive.
- The monetary cost deters the verifiers from maliciously using the RA protocol.
- The verifiers and provers are members of a Public Ethereum Network (PEN).
- Verifiers can send ARs to the provers only as Ethereum transactions.

Related Work

- In their RA design, Defrawy et al. proposed using timestamps to prevent CDoS.
- Brasser et al. used nonces to prevent CDoS attacks.
- SMARM allows interrupting RA so that the prover can perform its tasks.
- TM-Coin is the first RA design to combine an RA protocol with blockchain.
- Javaid et al. used Ethereum to counter Distributed Denial of Service (DDoS).

Our contribution

- We use Ethereum's fees and mining delays to protect the provers.
- A smart contract sets a limit to the number of ARs that reach a prover.
- Via Ethereum we provide nonrepudiation of actions and an immutable log.
- This log enables us to detect dishonest behavior from the verifier.

BARRETT's Assumptions and Threat model

- Healthcare IoT devices belong to a heterogeneous and dynamic network.
- We consider only adversaries that can interact only remotely with the prover.
- The adversaries are internal and use the RA protocol to harm the provers.
- Adversaries cannot physically interact with the device.
- Harming the prover by bypassing the BARRETT architecture is out of scope.

The BARRETT architecture

The BARRETT architecture comprises the following elements:

- The **PEN** which is the main component that protects provers.
- The **Verification Nodes (VNs)** which act as verifiers.
- The **provers**, which are IoT devices and the beneficiaries of BARRETT.
- A **smart contract** that regulates the number of ARs that reach a prover.

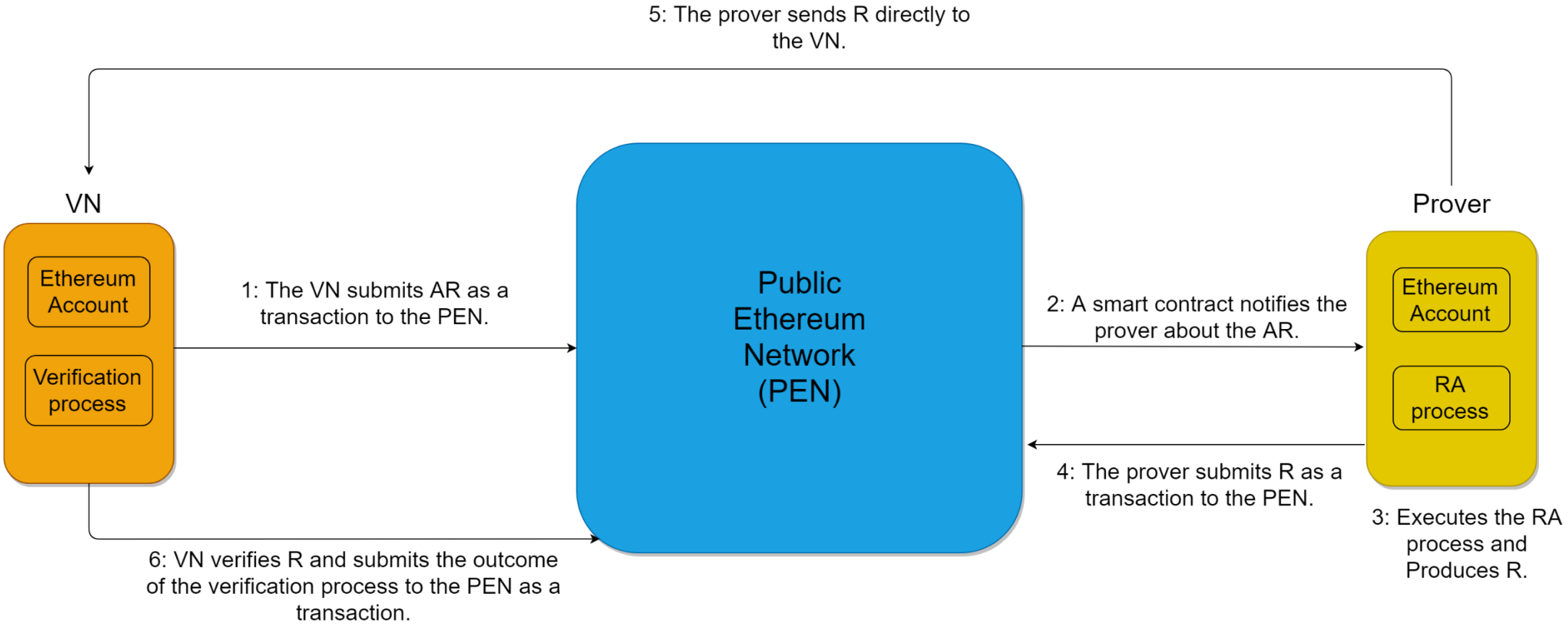
BARRETT – Components' features

Feature / Element	Ethereum Account	Ledger Copy	Shared Secret Key	Mining	Submits transactions
Full VN	✓	✓	✓	✓	✓
Light VN	✓	✗	✓	✗	✓
Prover	✓	✗	✓	✗	✓

Design Decisions

- We chose a PEN over consortium or private Ethereum because:
  a) It offers higher availability and robustness of data.
  b) Its mining delays are longer.
- BARRETT was not designed around an RA type for the following reasons:
  a) Compatibility with many types of RA.
  b) To protect provers in a variety of RA settings.

BARRETT – Flow of data



Performance Analysis

- The PEN imposes significant storage requirements to the **full** VNs.
- The mining process brings a considerable performance burden to the **full** VNs.
- Provers and VNs must dedicate 928 bits for storing information relevant to their Ethereum account.
- The above storage requirement is acceptable even for class-1 IoT devices.
- Mining can impose relatively long delays on the RA communication exchange.
- These delays may cause problems in RA scenarios that are not delay-tolerant.

Security Analysis

- Through the PEN BARRETT provides:
    a. Accountability and non-repudiation of actions.
    b. Data immutability.
    c. Constant data availability and single point of failure eradication.
- Ethereum fees and delays are **not enough** if ARs arrive from different verifiers.
- This is where the smart contract comes.

Conclusion and Future work

Conclusion:

- BARRETT is only suitable for RA scenarios that tolerate delays.
- In some cases, our architecture may become very expensive for honest VNs.

Future work will include:

- Expansion of the threat model's scope.
- A methodology that regulates fees for transactions that contain ARs.

Thank you!