Authenticating Internet Routing Using Zero-Knowledge Proofs

Jacob D. White
A-4: Advanced Research in Cyber Systems
Email: jdwhite@lanl.gov

Mentor: Michael Dixon
Co-Mentor: Boris Gelfand
Collaborator: Zachary DeStefano

August 9th, 2023
The Internet is a complex “network of networks”, allowing computers to route messages to each other across the globe.

Figure: US high-speed fiber optic connections (Lumen 2023)
■ Computers send data across *routers*

■ Organizations form a *network* of routers

■ Routers use *policies* and *protocols* to find and communicate with each other on the Internet
An organization manages an **Autonomous System** (AS) or domain of routers. Routing policies are defined internal or external to the domain.
An organization manages an Autonomous System (AS) or domain of routers. Routing policies are defined internal or external to the domain.

**Problem:** Routers who lie about how they route network data can cause serious disruptions and privacy issues!

**Traffic Diversion (Blackholing)**
Problem: Routers who *lie* about how they route network data can cause serious disruptions and privacy issues!
An organization manages an **Autonomous System** (AS) or domain of routers. Routing policies are defined internal or external to the domain.
An organization manages an Autonomous System (AS) or domain of routers. Routing policies are defined internal or external to the domain.

**Problem:** Routers who *lie* about their routing network data can cause serious disruptions and privacy issues!
Problem: Routers who *lie* about their routing network data can cause serious disruptions and privacy issues!
Key Questions:

1. How can we authenticate network operations, having routers learn from the global network, and behave according to policy?

2. How can we avoid leaking private information about organizations’ networks and relationships?
(Non-interactive) Zero-knowledge Proofs

Zero-knowledge proofs (ZKPs) allow us to prove that a claim IS true without revealing WHY it is true, even if the prover is considered untrusted and malicious.

Example Claim. “The packet can reach Router Y from X, even if Router Z goes offline”
Zero-knowledge proofs (ZKPs) allow us to prove that a claim *IS* true without revealing *WHY* it is true, even if the prover is considered untrusted and *malicious*.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miniscule Footprint</td>
<td>Some ZKP variants are tiny, often only slightly larger than a regular QR code</td>
</tr>
<tr>
<td>Fine-grained Control</td>
<td>ZKPs give fine-grained control over secret information, yet allows trustless verification</td>
</tr>
<tr>
<td>Composable</td>
<td>ZKPs can be collected and combined into new ZKPs without growing in size</td>
</tr>
<tr>
<td>Expanded Trust</td>
<td>Portable proofs extend our trusted view beyond that of our own system</td>
</tr>
</tbody>
</table>
Developed a prototype Rust library which provides useful gadgets for authenticating network routing attestations using ZKPs.

ZKPNet: An Overview

Network Applications

ZKP Protocol
ZKPNet Gadget API
ZKP Circuit API
Arithmetic constraints
Rust

Routing Protocol API
Network API
OS
Link Layer

Network routing / system backend (e.g. FRRouting)

ZKP backend (e.g. Arkworks)

ZKPNet library
Group A wants to send important data to Group C, but will need to go through Group B first. A and C first want to verify that B can deliver the data, but Group B is unwilling to reveal details about the network for security reasons. How does Bob prove this?
Results: ZKPNet Demo Benchmarks

Using realistic OSPF entries for internal routing, we have constructed zero-knowledge proof for route reachability for a single hop. Benchmarks were performed on a Apple M1 Max CPU with 32 GB of memory.

<table>
<thead>
<tr>
<th>ZKP Technique</th>
<th># of constraints</th>
<th>Proof Size (Bandwidth)</th>
<th>Proving Time (Latency / Delay)</th>
<th>Verification Time (Latency)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Proof</td>
<td>104</td>
<td>224 B*</td>
<td>468.03 ms</td>
<td>2.7165 ms</td>
</tr>
<tr>
<td>Depth-2 Recursion on Proof</td>
<td>13976</td>
<td>299 B*</td>
<td>TBD</td>
<td>TBD</td>
</tr>
</tbody>
</table>

*Estimated from Groth16 proof sizes with MNT4&6 curves
## Looking Ahead: Feature Support for Routing Auth.

### Table: Feature Support for Routing Auth.

<table>
<thead>
<tr>
<th>Feature</th>
<th>ZKPNet</th>
<th>BGPSec (best auth)</th>
<th>S-BGP</th>
<th>so-BGP</th>
<th>RPKI (deployed)</th>
<th>IRR (worst auth)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Route Integrity</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔️</td>
<td>✔️</td>
<td>❌</td>
</tr>
<tr>
<td>Comm/Bandwidth Efficiency</td>
<td>❔</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>✔</td>
</tr>
<tr>
<td>Dynamic/Adaptive Recovery</td>
<td>❔</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Trustless Authentication</td>
<td>✔</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>Privacy Preservation</td>
<td>✔</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
</tr>
</tbody>
</table>

### Legend

- ✔ Complete
- ❔ Partial
- ❌ Missing/Bad
- ❔ Varies
Future Work: ZKP Compiler for Verifiable Routing

```
as-num: 64496
import: {
  from AS64497 at 192.0.2.1
  action pref=0;
  accept community .contains(GRACEFUL-SHUTDOWN);
  from AS64497 action pref=10 accept ANY;
  from AS64496:AS-SECRET # ...
} except {
  from AS64497 at 192.0.2.1 accept RS-BOGONS-V4;
  # ...
}
```

```
router bgp 64496 # ...
neighbor 192.0.2.1 route-map AS64497-in in
neighbor 192.0.2.1 route-map AS64497-out out
!
route-map AS64497-in permit 10
  set local-preference 0
  match community graceful-shutdown
route-map AS64497-in deny 10
  match ip address prefix-list bogons-v4
!
# ...
```
Future Work: Using ZKPs to Inform RL-based SDNs

Software Defined Networking (SDN) routers take a different approach: adopt **Reinforcement Learning** (RL) techniques to decide optimal routing policies.

SDN requires much more data (often sensitive!) to inform routers.

Human-on-the-loop approach gives verifiable ZKP claims, allowing RL-based routers to reason about secret info as well!
Conclusion

- ZKPs can provide both privacy and authenticated routing guarantees, ensuring conformance to both protocol and policy specifications.

- Since ZKPs do NOT rely on key infrastructure, they are a promising tool for authenticating routing in a distributed environment.

- ZKPs will likely increase proving and verification times, with many overhead and maintenance challenges to consider before widespread adoption.
Backup
Integrating ZKP Information into RL-based SDNs (Backup)
Hardware routing not very complex – “on-chip” accelerators to perform specialized routing tasks very quickly

... also not very flexible

SDNs allow software itself to decide how to best route incoming packets / react to changing scenarios
Reinforcement Learning (RL) Overview

Agent (here, router) performs action given current state, environment (here, ML model/network sim) impacted, new state produced with reward/punishment for said action, back to agent.
Background: Resilient and Secure Cyber Networks

Traditional routers use heuristic networking protocols, such as BGP, to route and deliver messages between clients.

Traditional protocols are not resilient to drastic changes injected by adversaries.

Recent research has focused on learning-based software defined networking (SDN) routers that use reinforcement learning to ingest network state data and optimally route.
SDN (e.g. AI-based) routers require extra information about the network from other hosts to quickly adapt to new changes.

**Problem I:** Some hosts, including neighbors, may be malicious.

**Problem II:** Network details and/or messages may contain sensitive or proprietary information.
SDN (e.g. AI-based) routers require extra information about the network from other hosts to quickly adapt to new changes.

**Problem I:** Some hosts, including neighbors, may be malicious.

**Problem II:** Network details and/or messages may contain sensitive or proprietary information.
Solution: ZKPs for Network Security Properties

We can authenticate relevant peer-provided information used by smart routers using zero-knowledge proofs.

Properties that are true on one end of the network can be communicated to the other side with little-to-no trust.

We will use succinct ZKPs, so they will be small enough to add minimal overhead to the network.
Single-Prover ZKPs (Backup)
Zero-Knowledge Proof for *Where’s Waldo*?

**Example.** Proving that you know the solution to *Where’s Waldo?*
Example. Proving that you know the solution to *Where’s Waldo*?

**Traditional Proof:** Circle Waldo’s location
Example. Proving that you know the solution to Where’s Waldo?

Traditional Proof: Circle Waldo’s location

Problem

This kind of proof leaks all information about Waldo’s location, much more than simply that you have knowledge of the location (not zero-knowledge)!
Zero-Knowledge Proof for *Where’s Waldo?*

**Zero-knowledge Protocol**

1. Cut out a Waldo shaped hole in a much larger piece of paper
2. Position the hole over Waldo’s location

The sheet acts as an **obfuscating mask** for Waldo’s location

To verifiers, the book underneath could hypothetically be in any random orientation

Slide under paper
Zero-knowledge Proofs: High Level View

Proof

Convert

&

ZKP

Claim

Example Claim.
“The packet can reach Router Y from X, even if Router Z is offline”
Zero-knowledge Proofs: High Level View

The Zero Knowledge Proof replaces the need for sensitive proof information (effectively completely redacting the original proof).

The Claim can be quickly verified without any knowledge of the original proof.
Zero-Knowledge Proofs and Network Authentication

Zero-knowledge proofs (ZKPs) allow us to prove that a claim IS true without revealing WHY it is true, even if the prover is considered untrusted and malicious.

zkSNARKs are special ZKPs that are tiny and non-interactive

Inputs:
- Audit Logs
- Schematics
- Policies
- Signal Analysis
- Encryption Keys
- Attestations
  etc.

Outputs:
- Results
- ZK Proof of Network Integrity

Homomorphically Encrypted*

*with tweaks
Zero-knowledge proofs (ZKPs) allow us to prove that a claim IS true without revealing WHY it is true, even if the prover is considered untrusted and malicious.

**Features of (Non-interactive) Zero-knowledge Proofs**

- **Ideal Secrecy**: Secrets are NOT revealed even if the cryptography is completely broken.
- **Miniscule Footprint**: Proofs are tiny, often only slightly larger than a regular QR code (~3k Bits).
- **Fine-grained Control**: Exacting control over need-to-know while enabling trustless verification.
- **Composable**: ZKPs can be collected and combined into new ZKPs without growing in size.
Cryptographic Proof Systems

*Cryptographic* proof systems have variable completeness and soundness. For non-interactive zero-knowledge proofs we care about:

**(Completeness)** \( \mathbb{P}[\text{true statement AND verifier accepts}] = 1 \)
“Everything true is provable”

**(Soundness)** \( \mathbb{P}[\text{false statement AND verifier rejects}] = 1 - \varepsilon \)
“Low chance that a proof of a false statement is encountered”

We sacrifice minimal amount of soundness (have to break crypto to produce counter-example) in order to get valuable proof properties
zkSNARK Construction for Verified Computation [BCGTV13]

```c
int myFunction(int a) {
    int b = a * a - 4;
    return 3 * b + a;
}
```

Rank-1 Constraint System (R1CS):

\[
\begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 0 \\
\hline
a & 1 & a & 1 & a & 1 \\
0 & 0 & t_0 & 0 & t_0 & 0 \\
\hline
b & 0 & b & 0 & b & 0 \\
\end{bmatrix}
\]

Arkworks

- Computation
- Arithmetic Circuit
- R1CS
- QAP
- LPCP
- LIP
- zkSNARK

Arkworks backend

- Proof Representation Of Network Robustness
- Zero Knowledge Added
- Succinctness Added
- Interactivity Removed

Verifier Net View
Prover View

zkSNARK for Network Integrity
Spare (Ignored/Skipped)
Alert: RPKI is Vulnerable and Risky!

Weaknesses:

- Centralized trust is a point of failure
- Can’t certify entire route / network
- Keys are a target and hard to manage

Internet Infrastructure

Registries & Trust Anchors

Internet Service Provider (ISP)

Global Internet

Too Much Blind Trust!

Spoofable!
Secure and Robust ISP Network Routing

Existing: RPKI

- Trust sources are points of failure
- Only certifies info of route’s origin
- Authentication needs centralized keys
- Can only decide route on public (often local) info
Secure and Robust ISP Network Routing

Existing: RPKI
- Trust sources are points of failure
- Only certifies info of route’s origin
- Authentication needs centralized keys
- Can only decide route on public (often local) info

Our Solution: ZKPNet
- Trust sources are distributed
- Correctly verifies arbitrary info
- No auth keys, only trusted setup
- Also decides with secret global info