WAVE: A Decentralized Authorization Framework with Transitive Delegation

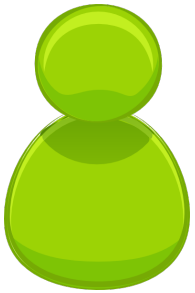
Michael P Andersen, **Sam Kumar**, Hyung-Sin Kim, John Kolb, Kaifei Chen, Moustafa AbdelBaky, Gabe Fierro, David E. Culler, Raluca Ada Popa

This material is based on work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1752814. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Roadmap

1. **The Problem**
2. WAVE's Approach
3. WAVE's Storage Layer
4. WAVE's Privacy layer
5. Implementation and Evaluation

Authorization for IoT



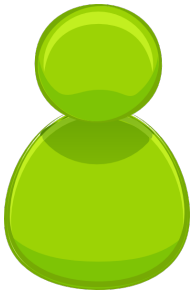
"Set temperature to 80 F"

Authorizati

Authorization

1. How does the tenant receive permission to adjust the temperature?

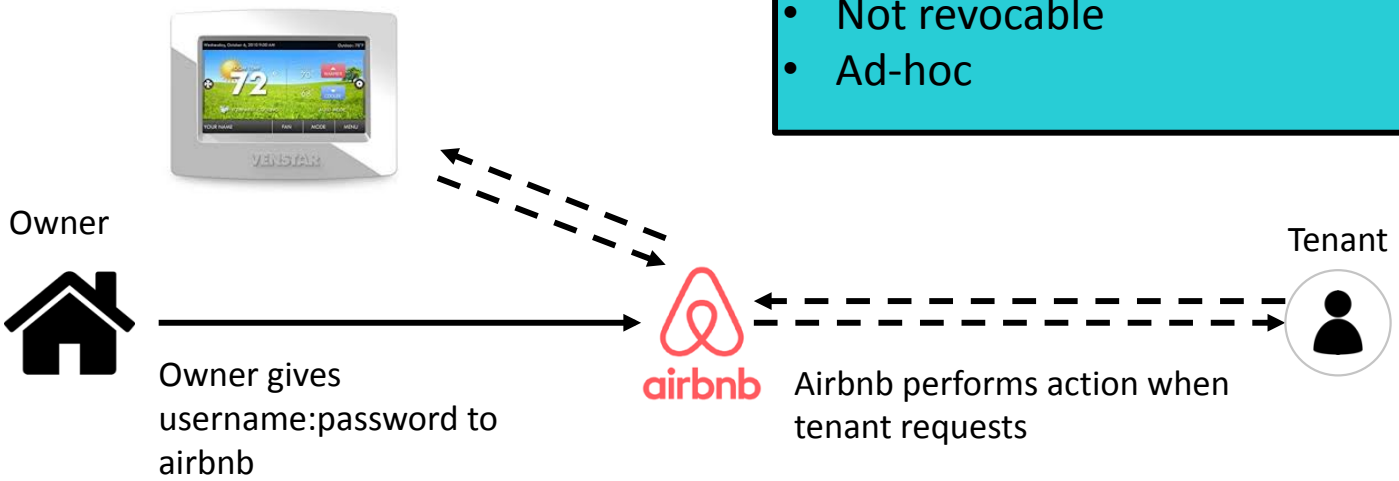
2. How does the thermostat know that the request was sent by someone who has permission?



"Set temperature to 80 F"

Authorization for IoT: Status Quo

- Shares identity, not just permission
- Not revocable
- Ad-hoc



The Problems

Existing authentication systems are centralized (often monolithic)

Transitive delegation is rare, leading to over-sharing

Attacks on centralized systems are common, and affect all the users

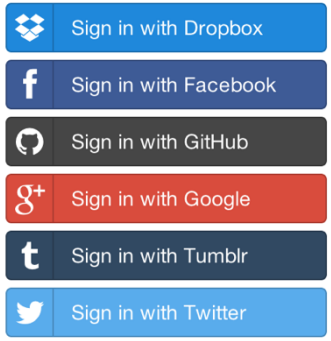
HACKER WARNING: Over a BILLION login and password details listed on the Dark Web

CYBERCRIMINALS have put on the Dark Web a massive database of 1.4billion e-mail, username and password details.

By DION DASSANAYAKE

PUBLISHED: 17:15, Tue, Dec 12, 2017 | UPDATED: 17:23, Tue, Dec 12, 2017

SHARE TWEET 49 3



Roadmap

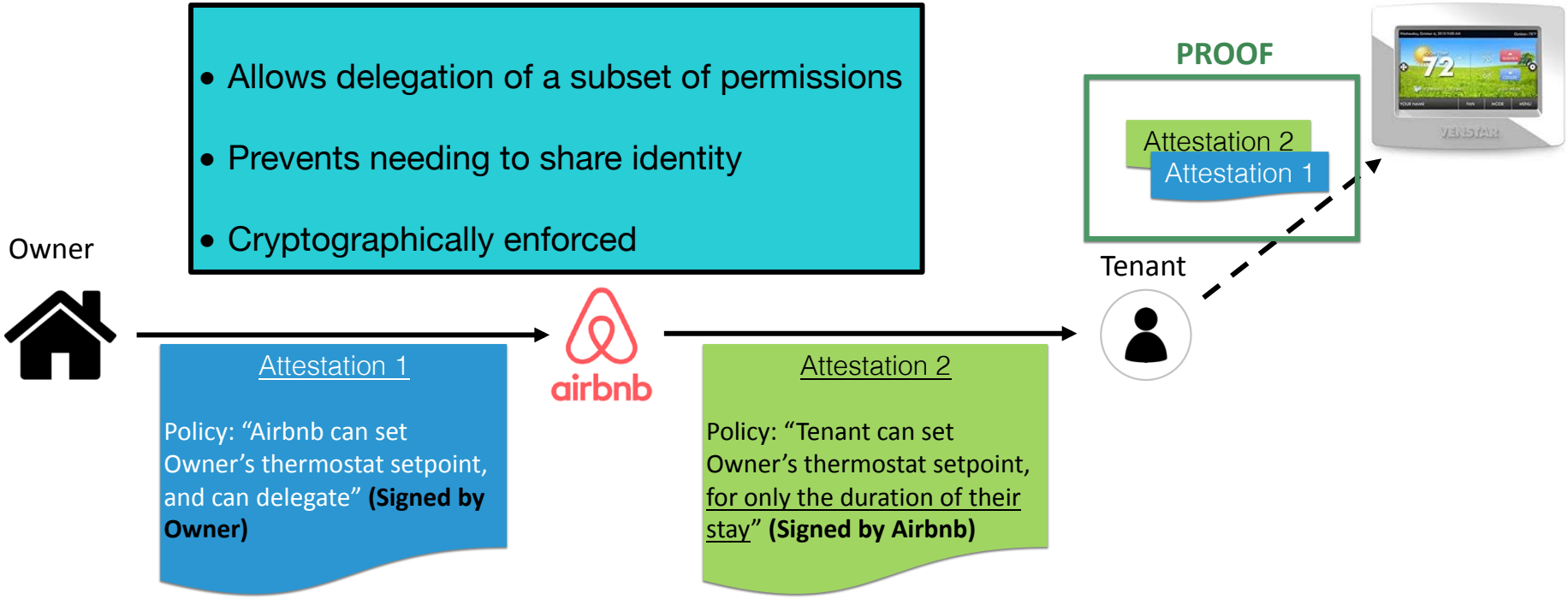
1. The Problem
2. **WAVE's Approach**
3. WAVE's Storage Layer
4. WAVE's Privacy layer
5. Implementation and Evaluation

WAVE's Approach

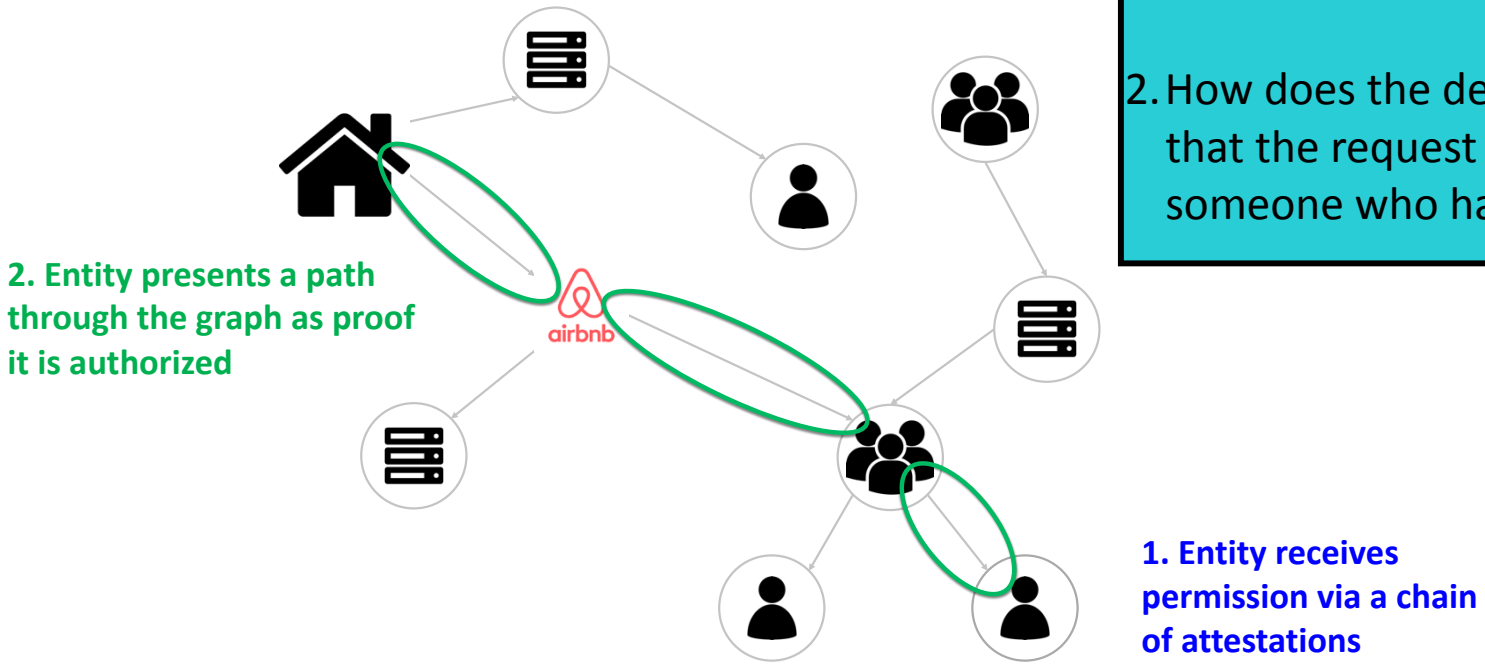
Make the flow of trust fine-grained.

WAVE Captures Trust Relations

- Allows delegation of a subset of permissions
- Prevents needing to share identity
- Cryptographically enforced



Global Permissions Graph



Authorization

1. How does one receive permission?
2. How does the device know that the request was sent by someone who has permission?

Two Technical Challenges in WAVE

1. How are attestations stored, disseminated, and discovered, without relying on a single trusted party?
   - Storage layer
2. How to protect the privacy of attestations?
   - Privacy layer

Roadmap

1. The Problem
2. WAVE's Approach
3. **WAVE's Storage Layer**
4. WAVE's Privacy layer
5. Implementation and Evaluation

Storage Layer Goals

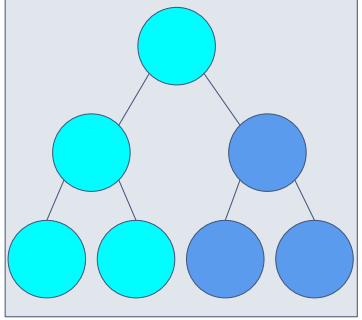
Storage provider is untrusted, so it must be verifiable that it is not:

- Hiding objects (such as revocation entries)
- Forging existence of non-existent objects

Blockchain is a natural solution but unfortunately doesn’t scale

WAVE's Storage Layer (First Try)

Merkle Tree **Log** of **operations**



Contains all the authorization objects

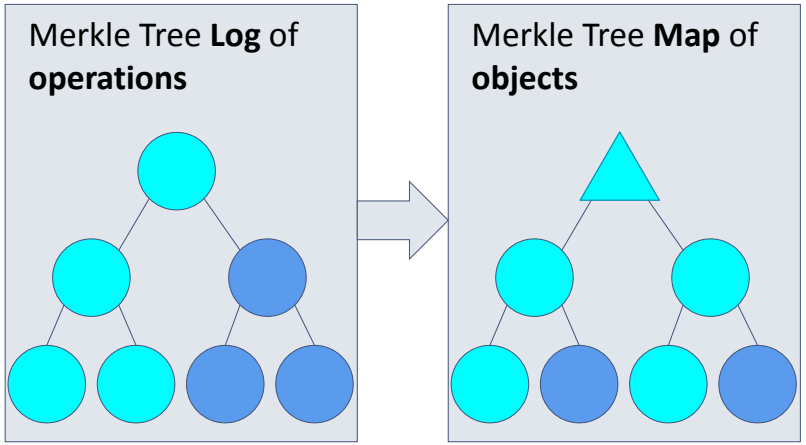
Can prove:
- Append-only
- Value exists in log

Use log of operations backed by Merkle Tree [Certificate Transparency, Laurie et al. 2013]

How to make sure the server can't hide objects?

- Server must be able to prove that an object doesn't exist
- Not supported by Merkle Tree Log!

WAVE's Storage Layer (Second Try)



Contains all the authorization objects

Can prove:
- Append-only
- Value exists in log

Contains objects indexed by their hash

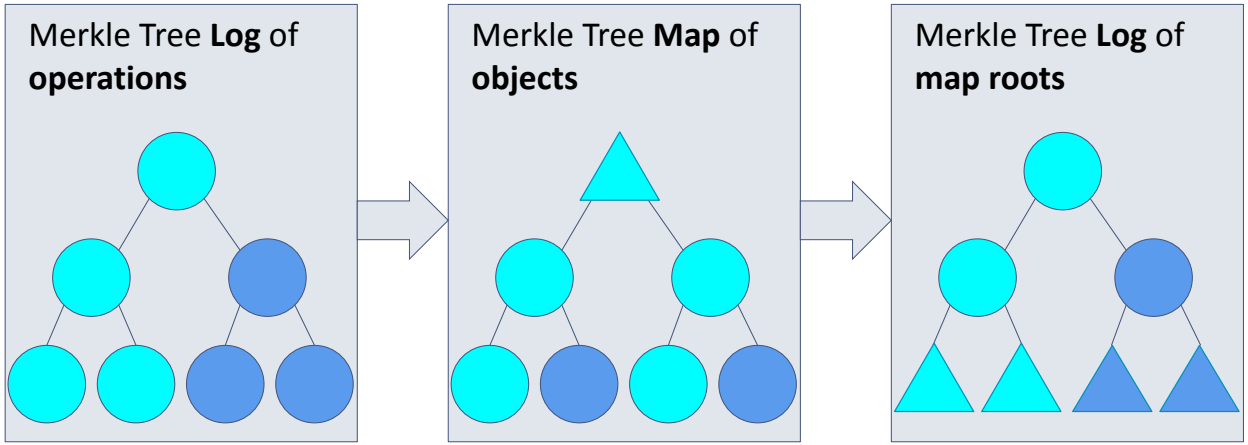
Can prove:
- Value does not exist
- Value exists

Use another Merkle tree to construct map of objects [Verifiable Log-Derived Map, Eijdenberg et al. 2015]

However, server could serve requests using an older version of the map

- How to fix this?

WAVE's Storage Layer (Final)



Contains all the authorization objects

Can prove:
- Append-only
- Value exists in log

Contains objects indexed by their hash

Can prove:
- Value does not exist
- Value exists

Contains all the root hashes of the map

Can prove:
- Append-only
- Value exists in log

Use another log to store progression of **map root hashes**

Auditors make sure that each request is served using the latest map version

Roadmap

1. The Problem
2. WAVE's Approach
3. WAVE's Storage Layer
4. **WAVE's Privacy layer**
5. Implementation and Evaluation

Private Attestations

With this storage model, global permissions graph is *publicly accessible*

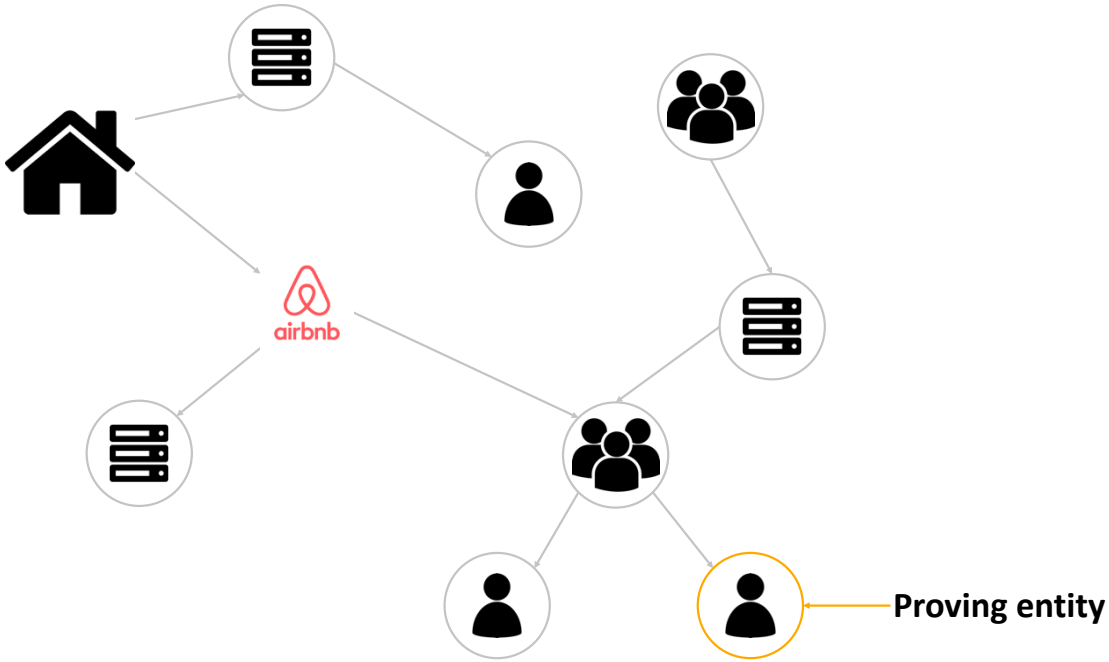
- Leaks, e.g., who is renting which house on Airbnb

Storage is untrusted; can't rely on it for access control

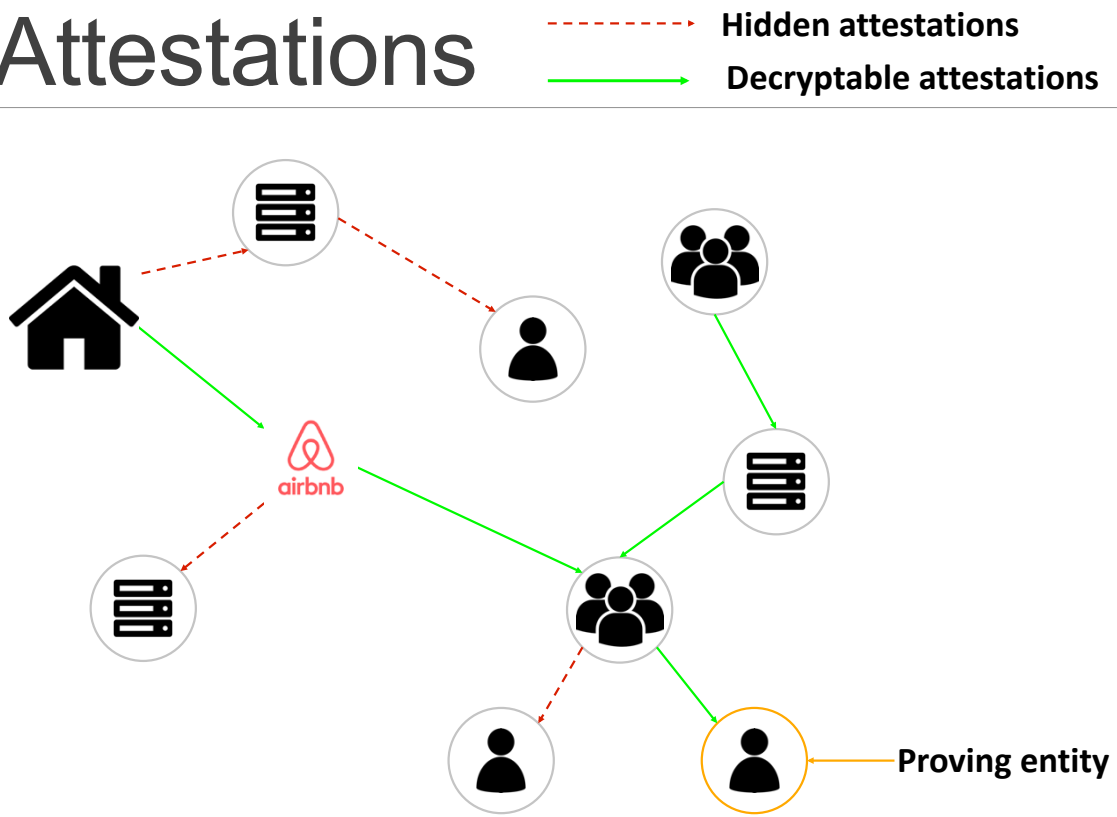
Instead we rely on cryptography

- Attestations are encrypted
- They can only be decrypted by an entity who can use them in a proof

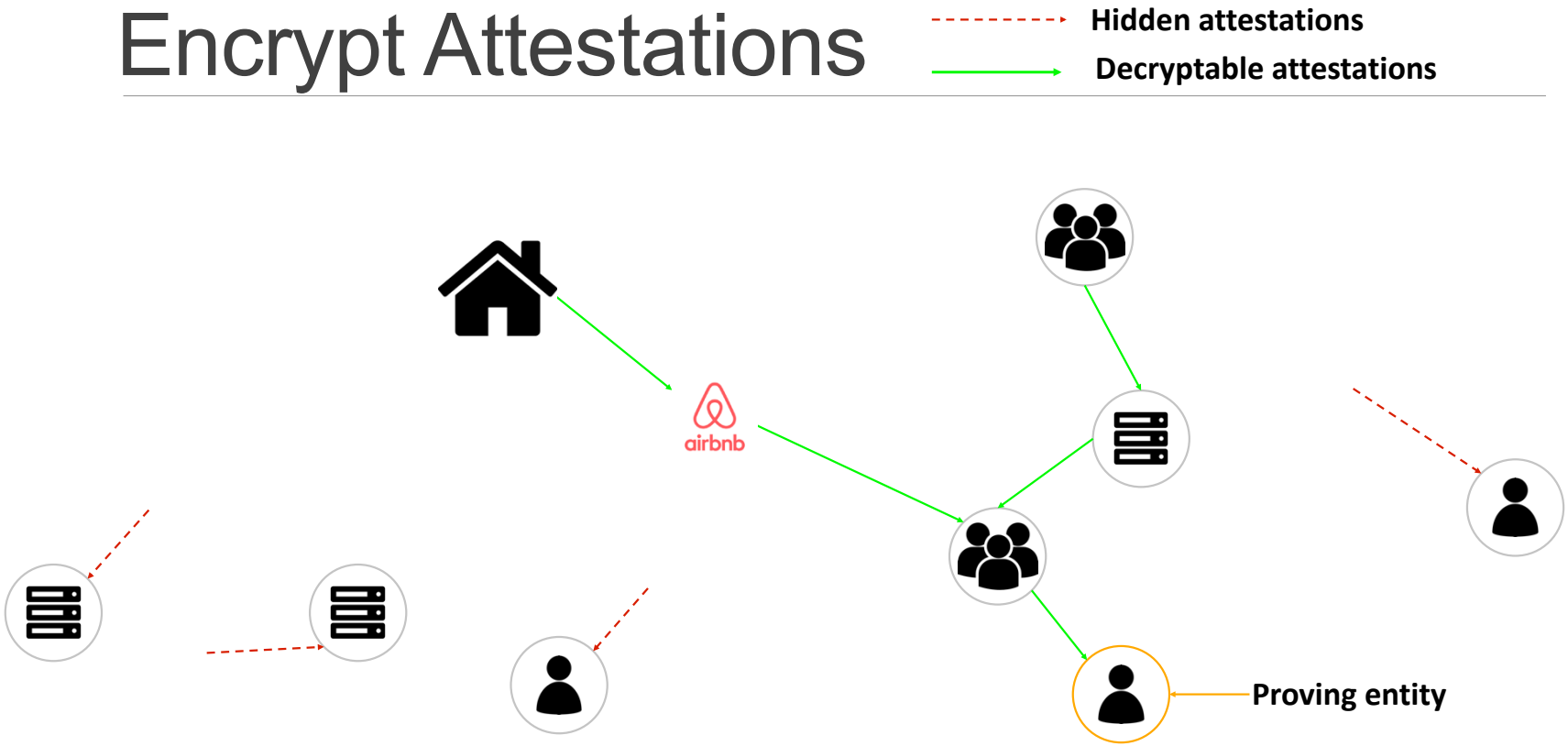
Encrypt Attestations



Encrypt Attestations



Encrypt Attestations



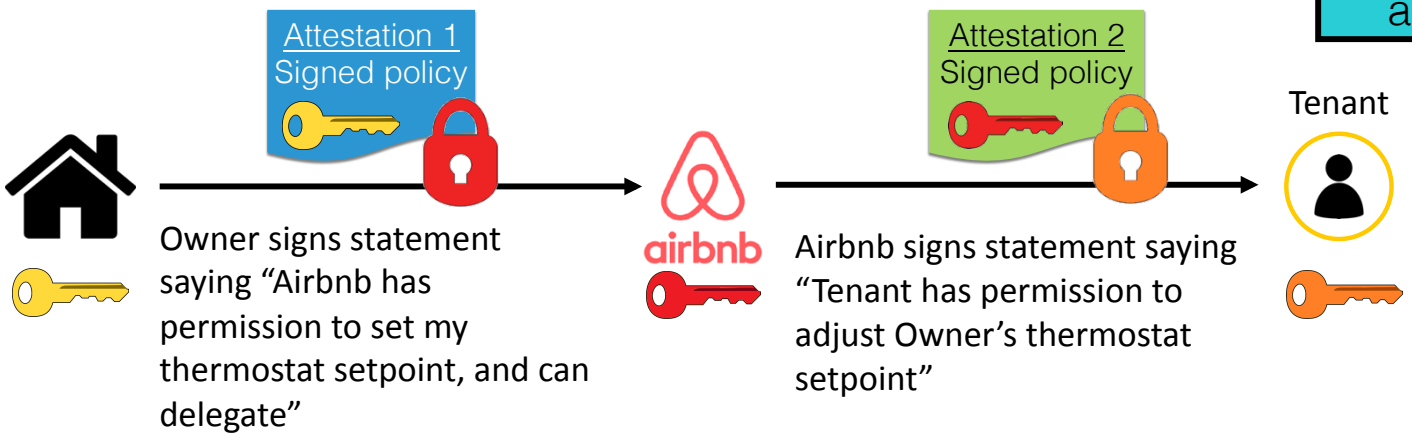
Our Technique: Reverse-Discoverable Encryption (simplified)

Attestations are encrypted using recipient's public key

Attestations include secret key of granter

- Allows decryption of upstream attestations

We actually use *policy-aware* encryption to restrict access further.



Our Technique: Reverse-Discoverable Encryption (simplified)

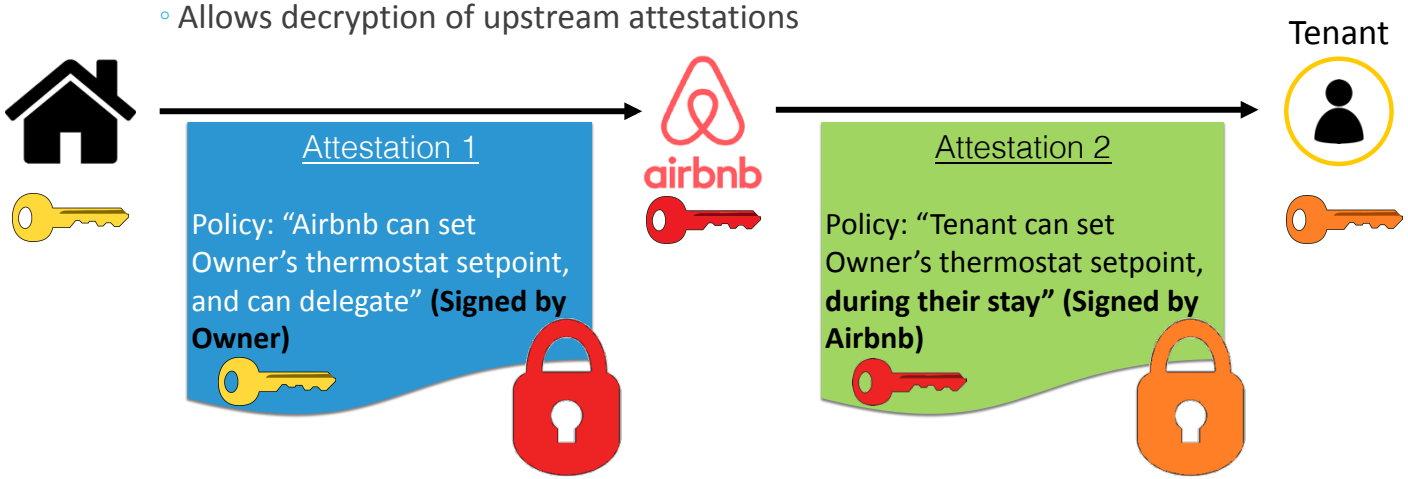
Each entity has a keypair for encrypting attestations

Attestations are encrypted using recipient's public key

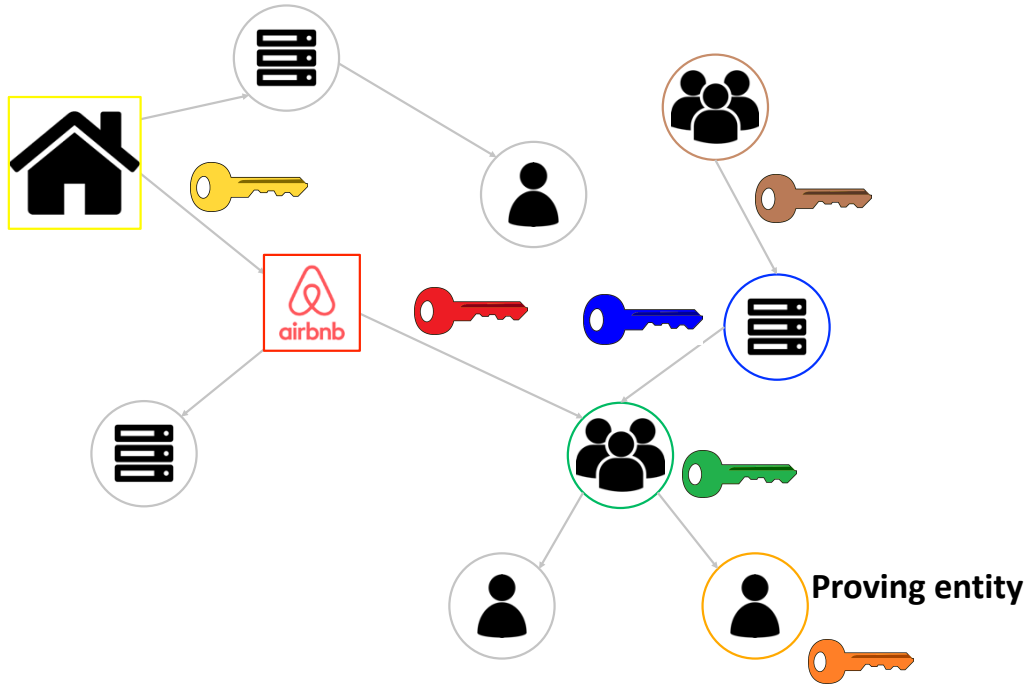
Attestations include secret key of granter

- Allows decryption of upstream attestations

We actually use *policy-aware* encryption to restrict access further.



Reverse-Discoverable Encryption



Roadmap

1. The Problem
2. WAVE's Approach
3. WAVE's Storage Layer
4. WAVE's Privacy layer
5. **Implementation and Evaluation**

First Release of WAVE Version 3

Feature	WAVE 2	WAVE 3
Delegation	Yes	Yes
Decentralized	Yes	Yes
Scalable	No (blockchain)	Yes
Encrypted Attestations	No	Yes
Fully General	No (IoT pubsub)	Yes
Full Implementation	Yes	Yes

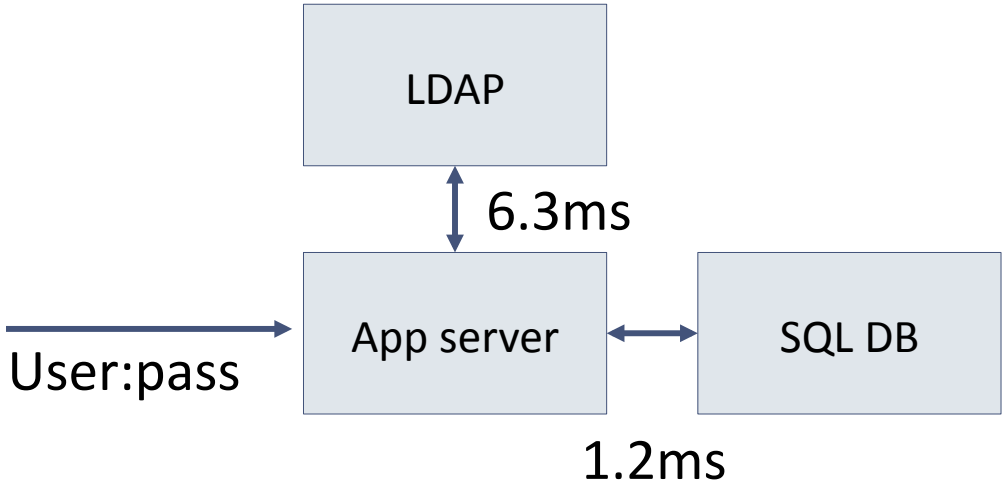
WAVE Version 2: github.com/immesys/bw2

WAVE Version 3: github.com/immesys/wave

Operation Times [ms]

	Operation	AMD64	ARMv8
Granting permissions →	Create attestation	39.4	754
Creating accounts →	Create entity	25.7	344
Verifying proofs →	Decrypt attestation as verifier	0.62	10.3
Discovering new attestations →	Decrypt attestation as subject	25.6	413
	Decrypt delegated attestation	31.9	642

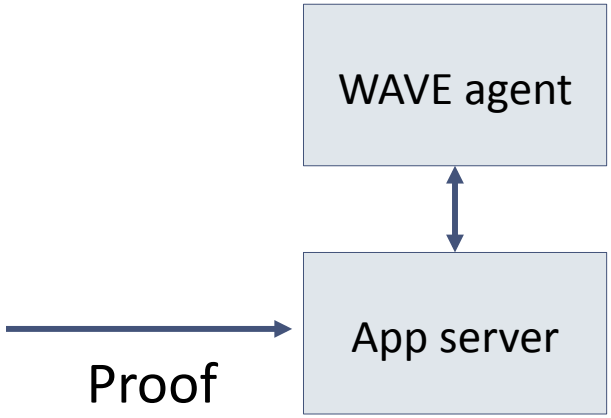
Use Case Comparison (Critical Path)



1. Authenticate
   - LDAP Bind
2. Check Auth Policy
   - SQL Lookup

Total: 7.5 ms

Use Case Comparison (Critical Path)



Proof Times:
Length 1: 2.8ms
Length 3: 6.2ms

1. Validate proof (yields policy)

Total: < 7 ms for common patterns

Conclusion

WAVE is an authentication/verification engine that makes trust relationships fine-grained

It can run at global scale without a central trusted party

It is a REAL artifact we have operated for 2 years, securing over 800 IoT devices in California!