Security Properties of Key Transparency

Esha Ghosh

IETF 118 - November 10, 2023

Key Transparency Systems

Service Provider maintains a directory of userid to public key mappings

Userid	Public Key
Alice	pk_A
Bob	pk_B
…	…

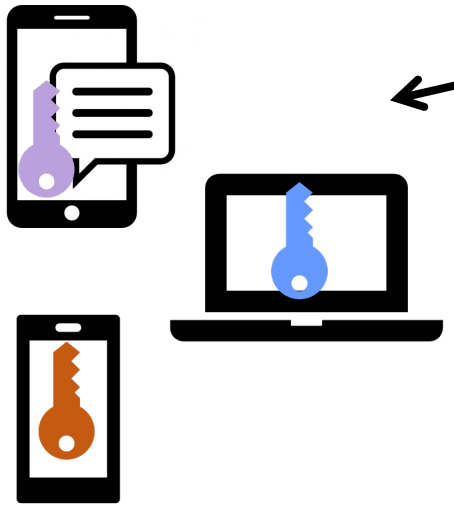
User Interactions:

1. Search/Lookup: own key or other contact's key
2. Update: own key
3. Monitor/Audit: checking for log consistency

Key Transparency Systems

Epoch	TreeHead
1	h_1
2	h_2
…	…

Verify the proofs against the latest TreeHead; alert users if proofs do not check out



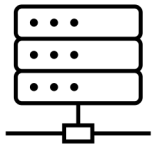
Clients get cryptographic proofs of correctness

Create TreeHeads periodically

Service holds public identity keys for every client device

Clients Devices

Auditable Key Directory Service



Security Properties

- When the log operator is honest: Correctness Properties
- When the log operator is malicious: Consistency Properties

Correctness Properties

1. When a user looks up a key, the result they receive is the same result that any other user searching for the same key would have seen

2. When a user modifies a key, other users will see the modification the next time they search for the key

Consistency Properties

1. When a user looks up a key, the result they receive is not the same result that any other user searching for the same key would have seen, it will be detected

2. When a user modifies a key, but other users do not see the modification the next time they search for the key, it will be detected

Consistency Properties

- Bob's[Owner] latest key: 
- When Alice[Receiver] queries for Bob's latest key, she sees a fake key: 

Consistency Properties

- Dissemination of TreeHeads
- State at clients:
  - User's own state
  - User's contact's state
- When is the inconsistency detected
- Who detects the inconsistency
- 3rd Party Auditing
- Owner Signing

Dissemination of TreeHeads

- All users should receive the same TreeHead for the same epoch: globally consistent view of the log.

- To detect forking of the log, the TreeHeads should be disseminated among the participants.

  - Bulletin Board: Disseminates TreeHeads through a third-party Bulletin Board

  - Gossip Channel: Dissemination of TreeHeads happen through gossip among the participants (comparing TreeHeads directly with each other)

Key Owner’s state

- Only the owner of the key (Bob) will be able to authoritatively decide that a distributed key was fake
- To detect this, Bob has to lookup the history of his own key in the log often
- Each time Bob changes his key, he has to check that the key change is correctly included in the log
- Bob also has to remember the epochs at which he updated his key.

User’s contact’s state

- Each user’s device may keep need to keep some state for their contacts
  - e.g.: The last keys of the user’s contacts, version numbers and possibly other auxiliary information for their contact’s keys

Who detects the inconsistency

- Bob [Key Owner] may detect the inconsistency
- Alice [Recipient] may detect the inconsistency

When is the inconsistency detected

- At least 2 checks needs to happen:
  - Alice [R] needs to ensure that the key she is seeing is committed by the server in the latest TreeHead
  - Bob [O] needs to see this key distributed on his behalf while monitoring for his own key after the fake key distribution

When is the inconsistency detected

- Bob [O] detects the inconsistency the first time he comes online

since the distribution of the fake key and monitors his key history

- Bob [O] cannot detect the inconsistency the first time he comes online since the distribution of the fake key and checks his key history: additional checks need to be performed (by Alice [R] or other parties) for this to be detected

Third Party Auditing (3PA)

- Third party auditor downloads and authenticates the log's content
- The auditor is trusted to run this correctly and attest to the result
- **This party is added for efficiency**: if the clients do not want to trust an external auditor, they can run the audit function themselves.

Owner-Signing

- If a malicious server publishes a TreeHead at a certain time and compromises a user's device some time after that

- The clients who hold that TreeHead will not accept any keys that the user's device did not authorize before being corrupted

Takeaways

- Several subtle dimensions of security properties
- It would be great to think about what are the desirable properties for a KT system
- Various combinations of the consistency properties offer different efficiency tradeoffs