TIGRESS

IETF 115

November 10, 2022

Presenters and Authors

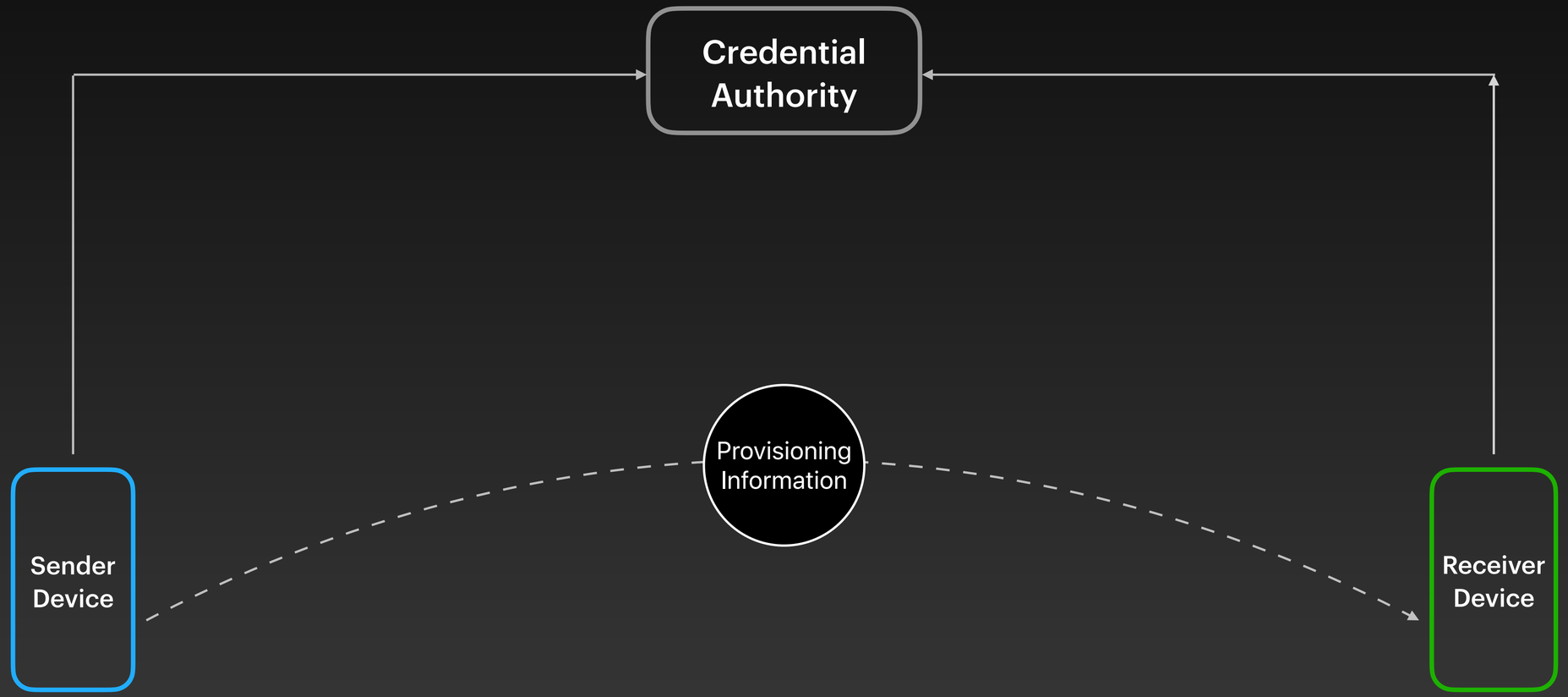
- Presenters
  - Brandon Leventhal, Casey Astiz, Dmitry Vinokurov
- Authors
  - Dmitry Vinokurov, organization: Apple Inc, email: dvinokurov@apple.com
  - Casey Astiz, organization: Apple Inc, email: castiz@apple.com
  - Alex Pelletier, organization: Apple Inc, email: a_pelletier@apple.com
  - Jean-Luc Giraud, organization: Apple Inc, email: jgiraud@apple.com
  - Alexey Bulgakov, organization: Apple, Inc email: abulgakov@apple.com
  - Matt Byington, organization: Apple Inc, email: mbyington@apple.com
  - Nick Sha, organization: Alphabet Inc, email: nicksha@google.com
  - Manuel Gerster, organization: Mercedes-Benz AG, email: manuel.gerster@mercedes-benz.com

Agenda

- Problem Statement
- Requirements Discussion
- Questions

Problem Statement

Today, there is no widely accepted way of transferring digital credentials securely between digital wallets independent of software and hardware manufacturer



Problem Statement

- Digital Credentials and hardware are owned by the provisioning partner
- Provisioning Partners need to keep track of share networks and permissions associated with each user
- Provisioning Partners want to enable sharing provisioning information between a sender and a receiver independent of software and hardware manufacturer

Example Use Case
Hotel

- When you stay at a hotel, the hotel operator owns the property as well as the physical locks.
- During a visit, you are being granted temporary access to a room. You’ve paid for two people to stay in that room.
- You checked in, and provisioned a digital hotel key to your digital wallet and would like to share it to your spouse.
- The Hotel (Provisioning Partner) provides you provisioning information to send to your spouse so they can provision their own digital hotel key.
- At the end of the stay, the provisioning partner revokes both of the digital hotel keys.

Requirements Discussion

Breakdown Summary

- Assumptions
- Sender and Receiver
- Transferred Data
- Intermediary Server Requirements

Assumptions

- Provisioning Partner may not allow for two users to use the same credential / cryptographic keys.
- User may not have the ability to revoke a credential that a Provisioning Partner has issued
- User may not have the ability to create new digital keys without Provisioning Partner interaction
- Security: Communication between Sender / Receiver and Provisioning Partner should be trusted.
- The choice of intermediary shall be defined by the application initiating the credential transfer.
- Sender and Receiver shall both be able to manage the shared credential at any point by communicating with the Provisioning Partner. Credential lifecycle management is out of scope for this proposal.
- Any device OEM with a digital credential implementation adherent to Tigress {{Tigress-00}} shall be able to receive shared provisioning information, whether or not they can originate provisioning information themselves or host their own intermediary.

Sender and Receiver

- (Req-Connectivity) Sender and Receiver shall be allowed to be online at different times. Sender and Receiver shall never need to be online at the same time.
- (Req-init) Solution should allow Sender to send the share invitation to Receiver over any messaging channel, with various degrees of security.
- (Req-P2P) A goal of credential transfer covered in this document shall include transfer from one device to another (group sharing shall not be a goal).

Transferred Data

- (Req-ArbitraryFormat) The solution shall support arbitrary message formats to support both keys that implement published standards like CCC as well as proprietary implementations of digital keys.
- (Req-RoundTrips) Solution shall allow for stateful requests between Sender and Receiver to support stateful actions like key signing requests
- (Req-Preview) Solution should allow for receiver to know what is being added to their digital wallet.

Security and Privacy

- (Req-Security) Solution should provide security of the provisioning data transferred (confidentiality, integrity and availability).
- (Req-Revoke) Solution shall maintain access control, allowing Sender to revoke before the share has been accepted, and for Receiver to end transfer at any time.

Intermediary Server Requirements

If the solution requires an intermediary server, it should have the following requirements.

- (Req-Privacy) An Intermediary server shall not be able to correlate users between exchanges, or create a social graph. Intermediary server shall not be an arbiter of Identity.
- (Req-Notify) Solution should support a notification mechanism to inform devices on the content update on Intermediary server.
- (Req-Opaque) Message content between Sender and Receiver must be opaque to an Intermediary.

Intermediary Server Requirements 2

If the solution requires an intermediary server, it should have the following requirements.

- (Req-IntermediaryAttestation) An Intermediary shall implement mechanisms to prevent abuse by share initiating device, verifying that the device is in good standing and content generated by the sender device can be trusted by the Intermediary. The trust mechanism could be proprietary or publicly verifiable (e.g. WebAuthN).
- (Req-ReceiverTrust) The Receiver device should be able to evaluate the trustworthiness of the Intermediary using a list of trusted/approved intermediaries.

Questions and Discussion

Document Links

- Github:
  - Requirements: https://github.com/dimmyvi/tigress-requirements/blob/main/draft-tigress-requirements.md

  Datatracker: https://datatracker.ietf.org/doc/draft-tigress-requirements/04/

  - Sample Implementation + Threat Model: https://github.com/dimmyvi/tigress-sample-implementation/blob/main/draft-dvinokurov-tigress-sample-implementation.md

  Datatracker: https://datatracker.ietf.org/doc/draft-tigress-sample-implementation/01/

  - Proposed Solution: https://github.com/dimmyvi/secure-credential-transfer/blob/main/draft-secure-credential-transfer.md

  Datatracker: https://datatracker.ietf.org/doc/draft-art-tigress/01/