System Goals

1. Agent's lifecycle managed by users
2. Communication between agents controlledd by user-defined access control policies
3. Agents having limited trust in third-parties
4. Scalability of agents
5. One agent can't remove other agents
6. Limited vulnerablility window when agents are compromised
7. Maintain system utility

- the goal is to desgn a system that provides:
    1. agent discovery
    2. secure agent communication
    3. fine-grained access control to the services provided by an agent, while allowing users to retain control over the amanagement of the agents

System Assumptions

1. Secure User Authentication and Human Verification
2. Adv cannot register agents under the identity of a benign user.
3. Agents are reachable at their registered endpoints
4. All cryptographic primitives used by the system—signature schemes are secure
5. Secure channel
6. Network infrastructure enforces basic protections against denial-of-service attacks and packet flooding

Threat Model

C1. Adv can register and add adversarial agents to the provider; They could also add themselves to the contact policy of benign agents by performing social engineering on users

C2. An honest agent can later be compromised

C3. Self-replicate agents

C4. Adv agents can establish communication with a benign agent

C5. Syibil attack

C6. Adv may overhear, intercept, and synthesize any message

The Architecture

User Registration

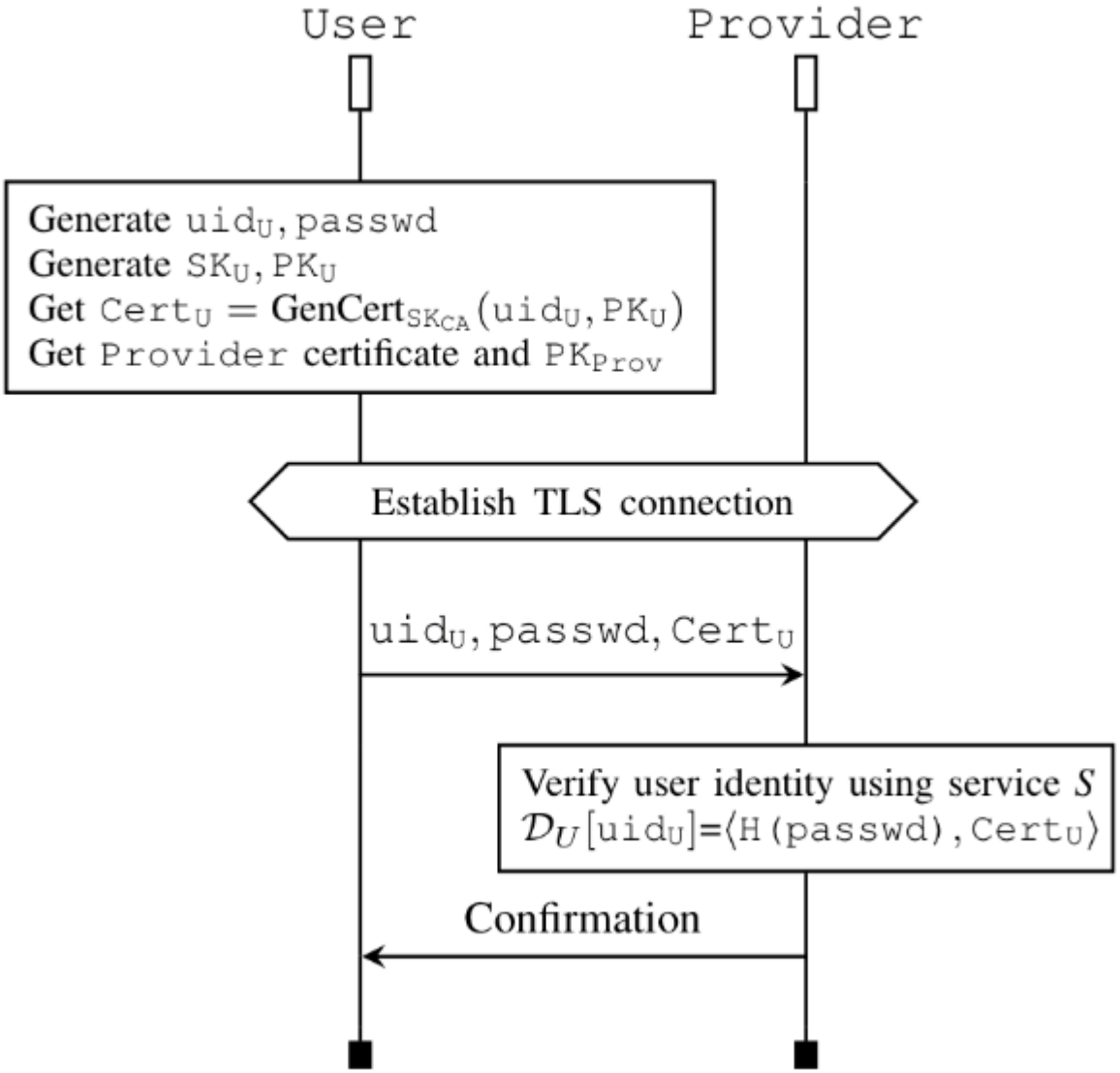


Fig. 7: **User Registration** protocol for user U.

- uid = 'example@email.com'

Agent Registration by User

- Aim to ensure each agent is bound to its user and a specific user's device

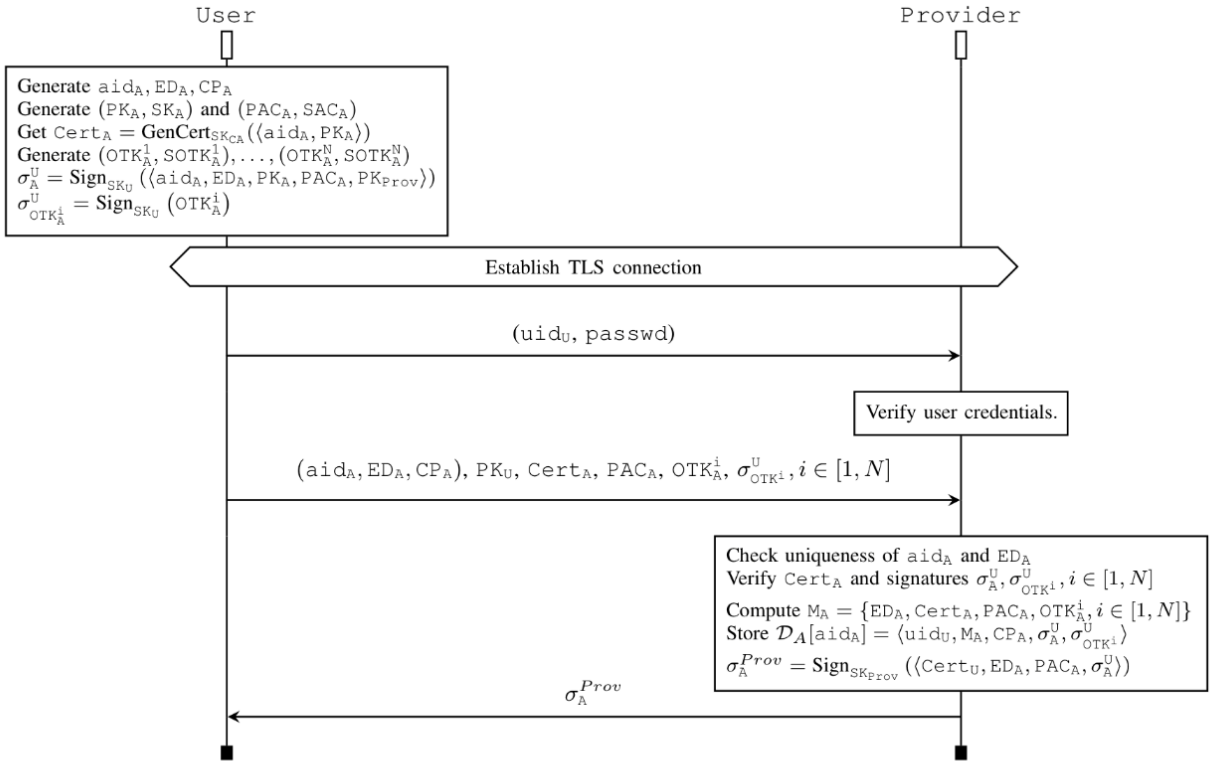


Fig. 8: **Agent Registration** protocol for agent A by user U.

-
- aid = uid || $name_A$
- ED is agent's end point descriptor
- sk, pk for tls
- pac, sac are long-term key pairs for access control: used for the receiving agent to derive a shared key with the initiating agent for encryption the access control token in inter-agent communication
- a bath of N one-time key pairs for deriving tokens for controlling access to the agent
- CP is contact policy
- at the end, agent receive a big signature so that it can prove to other agent it has been approved by provider

Agent Management

Access control management

1.Agent contact policy

- CP: specify which initiating agents are authorized to establish contact, defined by user + OTK
- ***e.g. ("*@company.com:email_agent", 10)*** means email agents from company.com can establish contact, can use 10 one time keys.

2.One-Time Key Generation

- provider distribute otk
- each otk is consumed when an initiating agent contacts the provider to obtain info for a receiving agent

Policy Updates and Revocation

- users can update their agents' CP from the privider, i.e. adding new rules to onboard collaboratoers, removing collaboraters, block contact, deactivate their own agents
- key rotation

Inter-Agent Communication

e.g. Agent B wants to contact agent A

1. B queries provider to verify A's registration
2. If permitted by A's CP, B receives A's metadata and a otk
3. A and B use otk to derive a shared key
4. A encrypt and access control token(ACT) for B (*ACT is scoped to a specifc task*)
5. B attach this token for every subsequent request to A untill it expires (question: is the token linked to a service or to an agent? If to an agent, what if the agent is compromised?)

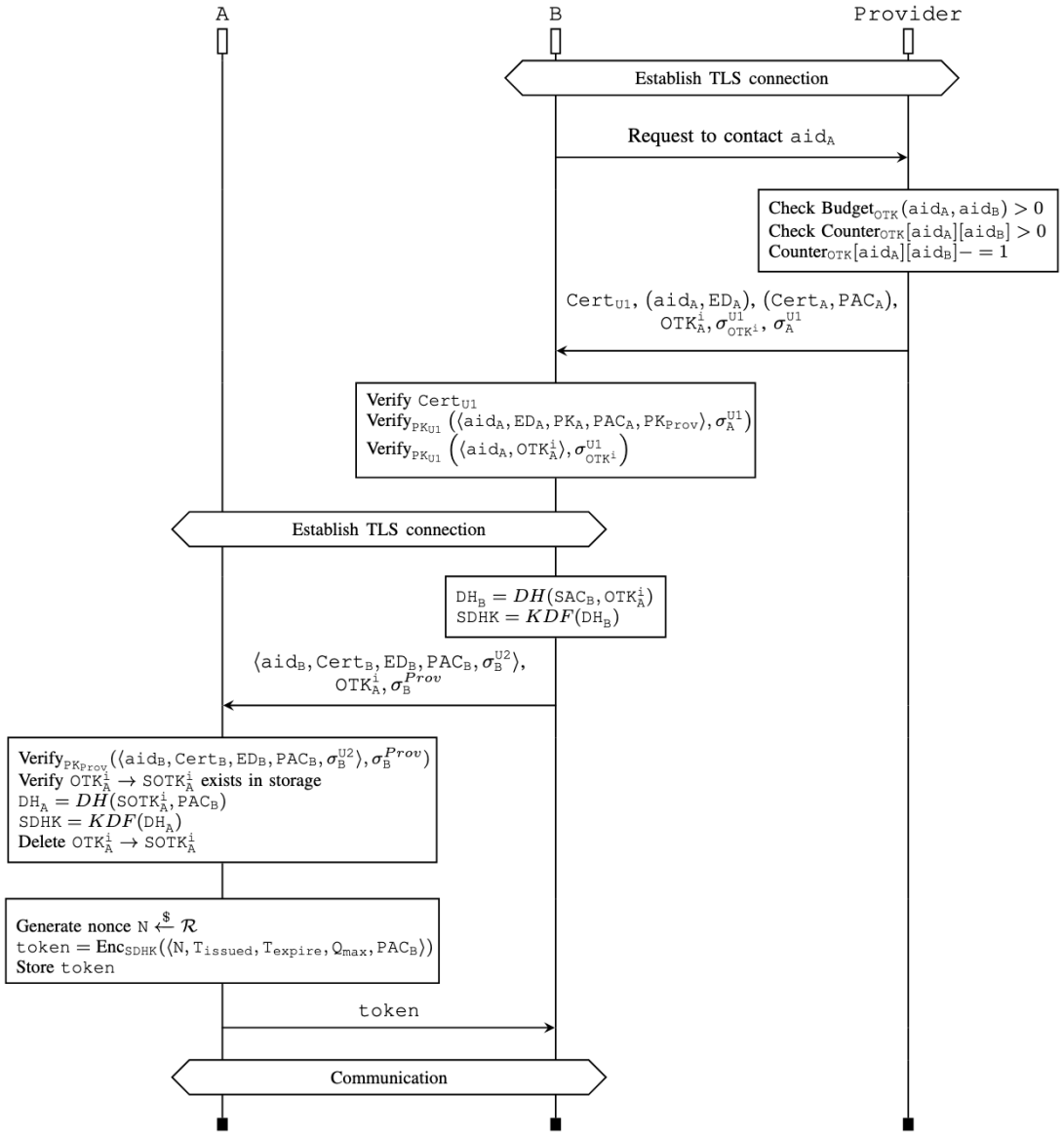


Fig. 9: **Agent Communication** protocol between initiating agent `B` and receiving agent `A` to obtain an Access Control Token with the `Provider` mediation if first communication between agents or the previous token expired.

Formal Analysis

- used ProVerif (symbolic model)
- under an attacker that can observe, intercept, modify, replay, reorder, and synthesize arbitrary messages on the network
- proved:
    - the secrecy of ACT token
    - authentication of communication between agents and provider
    - authentication of communication between agents

(question: but this still does not capture the main safty concerns regarding agent security, which is a compromised agent --> ok. they use actual agents with this capabilities to evaluate it...)

Attack Evaluation

They implement 8 attacker agents and explain on which steps the agents can't accomplish the protocol.

(Is this justification enough?)

Reflection

- The main design is a central provider that acts both as a bookkeeping for agents relationship and capabilities and identity registration, and an encrypted token linked to a specifc service which is used for authenticating allowed agent. Very similar with Kerberos.
- None of them seems agent unique -- This makes we wonder what are intuitive assumptions unique to an agent in comparison to a human user?