Cybersecurity – Identity Management
Prof. Dr. Anja Lehmann
Master Project | Winter Term 22/23

MLS Multi-Client Authentication

In this joint project with Phoenix R&D, we want to explore current solutions for cryptographic identities used by multi-client messaging systems and explore a novel approach based on the emerging Messaging Layer Security (MLS) protocol that offers superior security guarantees compared to existing messaging applications: MLS transcript-based multi-client authentication.

Authentication in Messengers – Status Quo

In the past years, the use of end-to-end encryption has become quite ubiquitous in the messaging space. A number of messengers have adopted the Signal protocol or a derivative thereof. In that respect, the security properties at the protocol layer have become somewhat comparable among messengers.

The situation is very different when it comes to client authentication, i.e. the ability of users to verify the person they are actually exchanging messages with. This is complicated somewhat by the fact that even conversations between only two users are technically group conversations between the sets of clients of each of the users, thus changing the challenge from client authentication to multi-client authentication. The mechanisms used in practice are very heterogeneous and provide wildly different security guarantees. The following is a broad categorisation of such mechanisms:

Unverifiable authentication (*Examples: iMessage, Skype private chats*)
While users and their clients use cryptographic identity keys to authenticate end-to-end encrypted messages, end-users have no visibility of these identity keys. With this lack of transparency, a malicious or compromised service operator can potentially orchestrate man-in-the-middle (mitm)-attacks without the users' knowledge.

Visible fingerprints/public keys (*Examples: PGP, WhatsApp, Signal, Wire*)
Users can inspect comparable fingerprints that are typically derived from one or more of the following: user identity key, client identity key, session state and user metadata. This gives users the ability to detect a mitm-attack. While this provides good security guarantees in theory, experience shows that users don't compare fingerprints in practice, thus decreasing the chance of detecting a mitm-attack quite substantially. Also, clients have to compare fingerprints for clients involved in a conversation for true client authentication (with the exception of WhatsApp, which recently introduced their new approach to multi-client authentication).

Linked identities (*Example: Keybase*)
Particularly in multi-client scenarios (where users can have more than one client) where each client has its unique cryptographic identity key, security can be improved by additionally and automatically linking these identity keys, e.g. by signing or cross-signing new identity keys. Because of its automated nature, this mechanism reduces the risk of mitm-attacks without any particular user action.

Multi-client authentication
Most popular messaging apps such as Signal and WhatsApp allow the authentication of individual clients, but either don't consider the multi-client nature of most conversations at all (Signal) or have only recently begun exploring linking the individual clients of a user (WhatsApp). More sophisticated schemes such as the one used by Keybase exist, but have not seen adoption by mainstream

messaging applications. With MLS as an emerging IETF standard as base, there is now the opportunity to innovate on existing approaches and lift the standard for end-to-end authentication in secure messaging.

Desired Security Properties

We now informally define a few security guarantees that MLS transcript-based multi-client authentication should provide.

Identity binding

The key material of all clients of a given user should be cryptographically bound to the user's and to their own respective identity, for example a user and/or device names. Other users should be able to verify this binding. For the initial verification, the verifier is allowed to rely on a trusted third party, i.e. the identity provider of the user's identity.

Evolving/Continuous authentication key material

If a client has initially verified another user's identity binding at some point, they should be able to authenticate changes to the user's set of clients without trusting the identity domain owner. A change to the user's key material could be:

- adding a new client
- removing a client
- updating the key material of an existing client

This precludes mitm-attacks through the user's identity provider.

Record of intent

Other user's clients have to be able to verify that a given client intended to be added to another user's identity.

Message authentication

When receiving messages from a client of a given user, the recipient should be able to authenticate the messages or the key material used to authenticate the message using the key material bound to the user's identity.

Explicit verification of identity root

Users should be able to publish a static value that serves as additional trust anchor when initially verifying the binding between identity and key material.

Overview of MLS Transcript-based Multi-client Authentication

The MLS protocol is a key establishment mechanism between multiple clients. It uses "groups" as fundamental entities that are composed of "members". In addition to negotiating shared keys between members of a group, it also tracks the group membership and provides membership agreement among members. Operations like adding and removing members to/from a group are represented as so-called "handshake messages" and are added to an append-only log called a "transcript". Cryptographically speaking, the transcript is a combination of a hash chain and a signature chain.

To achieve the security guarantees of multi-client authentication described above, we can re-use the MLS transcript and its security properties with regard to membership in a group. We use the notion of a group to designate the set of clients pertaining to a user account. We assume that each user account initially has one client and further clients are only added subsequently. We map the operations of adding & removing clients from an account to adding & removing members of a group. The resulting transcript can serve as an auditable log to determine which clients are currently associated with a user account. This mechanism would fall in the "linked identities" category.

The following list describes the functional operations of the system:

Add a new client: Add a new client to the set of trusted clients by generating the appropriate log entry.

Remove a client: Remove a client from the set of trusted clients by generating the appropriate log entry.

Update a client: Update the key material associated with a given client by generating the appropriate log entry.

Verify integrity of a transcript: Given a transcript, verify that every log entry is valid.

Verify if a given client is trusted: Given a transcript, verify if a given client is present in the transcript and has not yet been removed.

Goal of the Master Project

The goal of this master project is to work out a full protocol specification, security model, cryptographic security proof and prototypical realisation of the MLS transcript-based multi-client authentication based on OpenMLS.

- Formalise the desired security properties
- Investigate further desirable properties
- Propose MLS extensions that are functionally required
- Write a specification of the concept
- Conduct a security proof for the specification w.r.t. the formalised security properties
- Implement an MLS transcript analyser (in Rust)
- Implement the necessary MLS extensions to perform the above-mentioned operations (in Rust)
- Optional: Analyse how this concept can be integrated with something like certificate transparency

Requirements

We expect a solid understanding and interest in cryptography and provably-secure constructions. You should have successfully attended and enjoyed the (Introduction to) Cryptography lecture. To implement the goals at a proof-of-concept level, at least part of the group must have sufficient Rust development skills (or the motivation to acquire them).

This master project will be co-supervised by Phoenix R&D, namely by Raphael Robert & Konrad Kohbrok.

About Phoenix R&D

Phoenix Research & Development is a young company, but we are not new to the game. We as individuals have been active in the area of secure messaging in both industry and academia for over 10 years. Phoenix R&D specialises in private and secure messaging technologies. We have co-initiated and co-authored both the IETF MLS protocol as well as its open-source implementation – OpenMLS.

Contact

Prof. Anja Lehmann: anja.lehmann@hpi.de