1 Protocols problem set

This document presents four possible protocols that we propose to use for the protocol side of the crypto ladder. The proposed protocols are, ordered by increasing proof difficulty in the crypto world:

- Basic-Hash (Example 1), a RFID tag/reader access protocol, meant to provide unlinkability, a strong notion of anonymity, between the tags. This is the base pattern for electronic passport authentication.
- Signed DH (Example 2), a simple key-exchange providing secrecy and authentication, which is the historic pattern for SSH/TLS.
- Signed DH+KEM (Example 3), a variant of the previous one with a KEM added into the mix, following the pattern for the ongoing hybridization of SSH/TLS.
- A simplified NTOR protocol (Example 4), a DH key-exchange using long term DH keys for authentication instead of signatures, which is the base pattern for Wireguard and PQXDH.

Rational behind this current set:

- provide a variety of security properties (secrecy, authentication, privacy);
- have examples linked with widely deployed real-world applications;
- start with a very simple example and increase difficulty.

Open questions:

- should we formalize the expected security properties for each problem ? The issue is that no common formalism might make sense for both the ProVerif/Tamarin side and the more crypto like tools.
- should we consider advanced compromise scenarios for the key-exchanges (long term key compromise, ephemeral key compromise) ?
- is this too many examples ? too few ?

Notations Secret keys and key pairs are denoted by using the subscripts sk/pk to distinguish between secret/public parts. We rely on usual algorithms for KEM, signatures and DH. We use a generic hash function Hash without a fixed arity, modelers may use the most canonical way to represent those computations in their tools.

1.1 Basic-Hash

The protocol is given in Figure 1

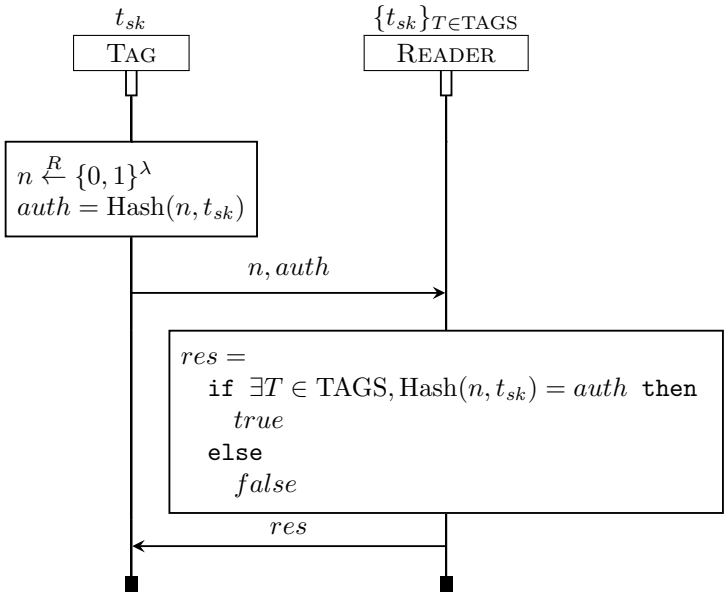

Figure 1: Basic Hash

Description For a set TAGS of valid identities, a RFID reader has a database of valid secret keys $\{t_{sk}\}_{T\in\text{TAGS}}$. A RFID tag with identity T and key t_{sk} authenticates to the reader by sampling a fresh challenge n, and sending the pair $(n, \text{Hash}(n, t_{sk}))$ to the reader, which looks into its database to see if the hash can be mapped to a valid identity and then answers true or false.

Properties This protocol should provide two guarantees against an active machine-in-the-middle attacker:

- authentication - if the reader accepts for some identity, the corresponding tag did act.
- unlinkability - it should be impossible to decide whether two sessions of the protocol correspond to the same tag or not. That is, it should be impossible to distinguish a scenario where there is a single tag executing n times the protocol in sequence with a scenario where n distinct tags all execute the protocol each a single time.

1.2 DH/KEM based key exchanges

The three protocols are given in Figures 2,3,4.

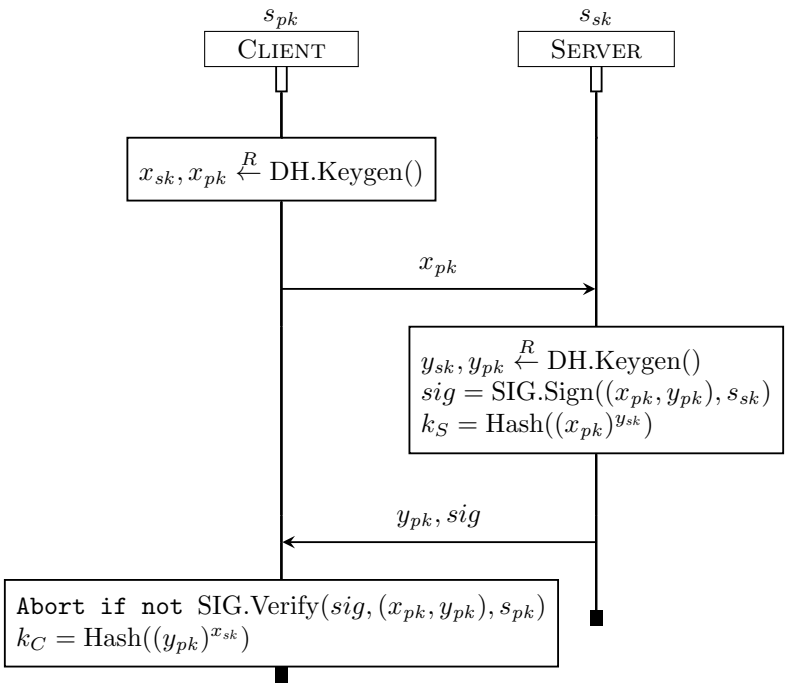

Figure 2: signed Diffie-Hellman key exchange

Description In all the three key-exchanges, the server has a long term secret keypair s_{sk}, s_{pk}. (which is a signature key for Examples 2,3 and a DH key for Example 4). This key is used to authenticate the server, while the client and the server exchange fresh DH or KEM values to derive a shared secret key.

Properties This protocol should provide two guarantees against an active machine-in-the-middle attacker:

- authentication - if a client derives a key with some parameters (the set of public keys), a corresponding server session derived the same key with the same parameters.
- secrecy - nobody except the two matching sessions can derive the key. The key might be proven to be indistinguishable from a random value.

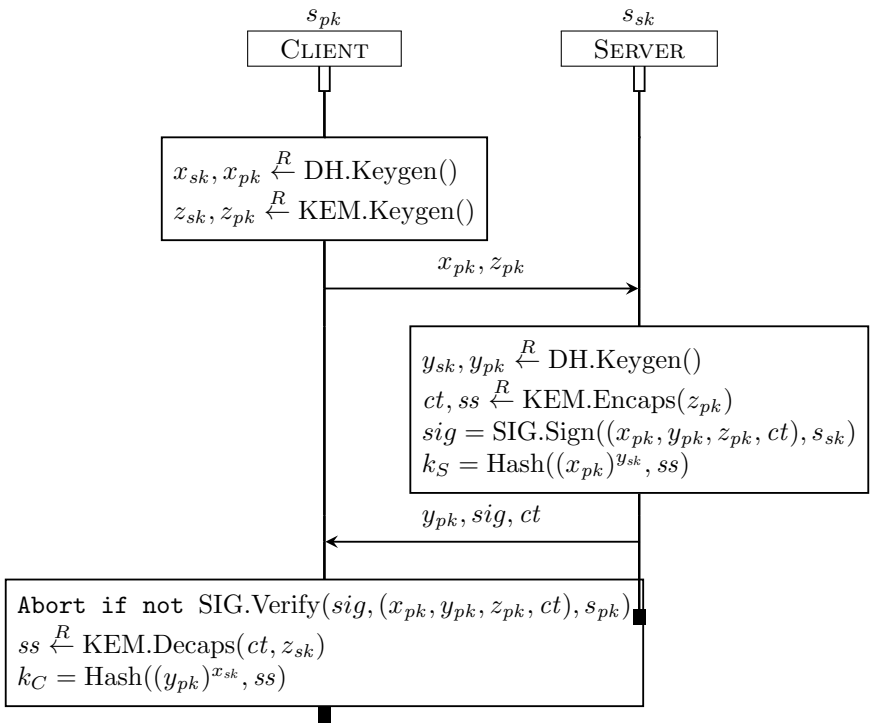

Figure 3: Hybridization of a signed Diffie-Hellman key exchange

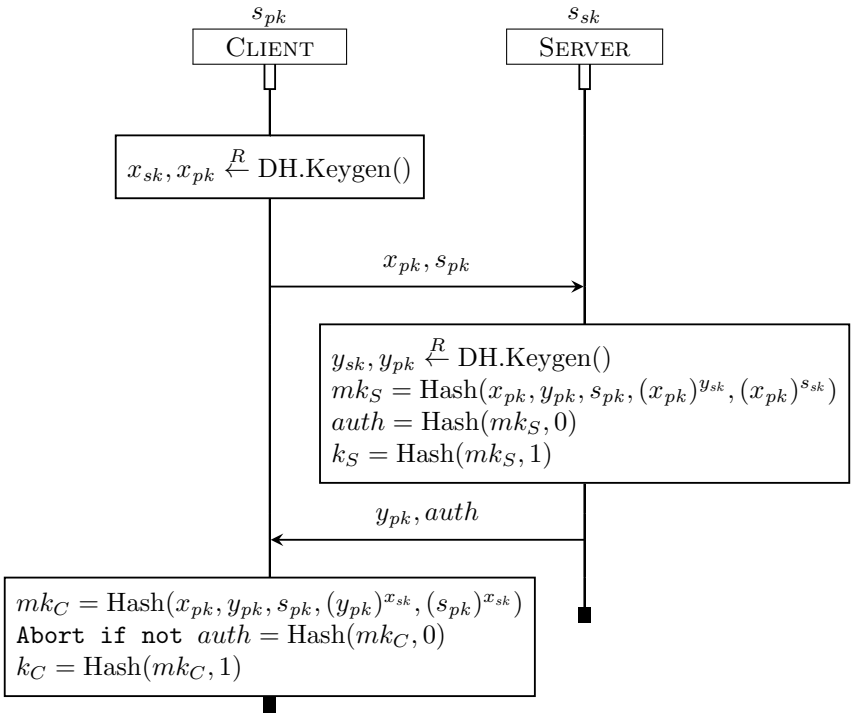

Figure 4: Simplified NTOR