SAS-Based Group Authentication and Key Agreement Protocols

Sven Laur[1,2] and Sylvain Pasini[3]

[2]University of Tartu
[1]Helsinki University of Technology

[3]Ecole Polytechnique Fédérale de Lausanne

Brief outline

- User-aided data authentication
  - ▷ What is user-aided data authentication?
  - ▷ Why do we need it in practice?
- Two-party protocols
  - ▷ The simplest protocol
  - ▷ How to achieve optimal security?
- Generalisations for the group setting
  - ▷ Formal security model
  - ▷ Description of SAS-GMA protocol
  - ▷ How to combine SAS-GMA with key agreement protocols?

Motivation

Communication in wireless networks can be altered and modified.

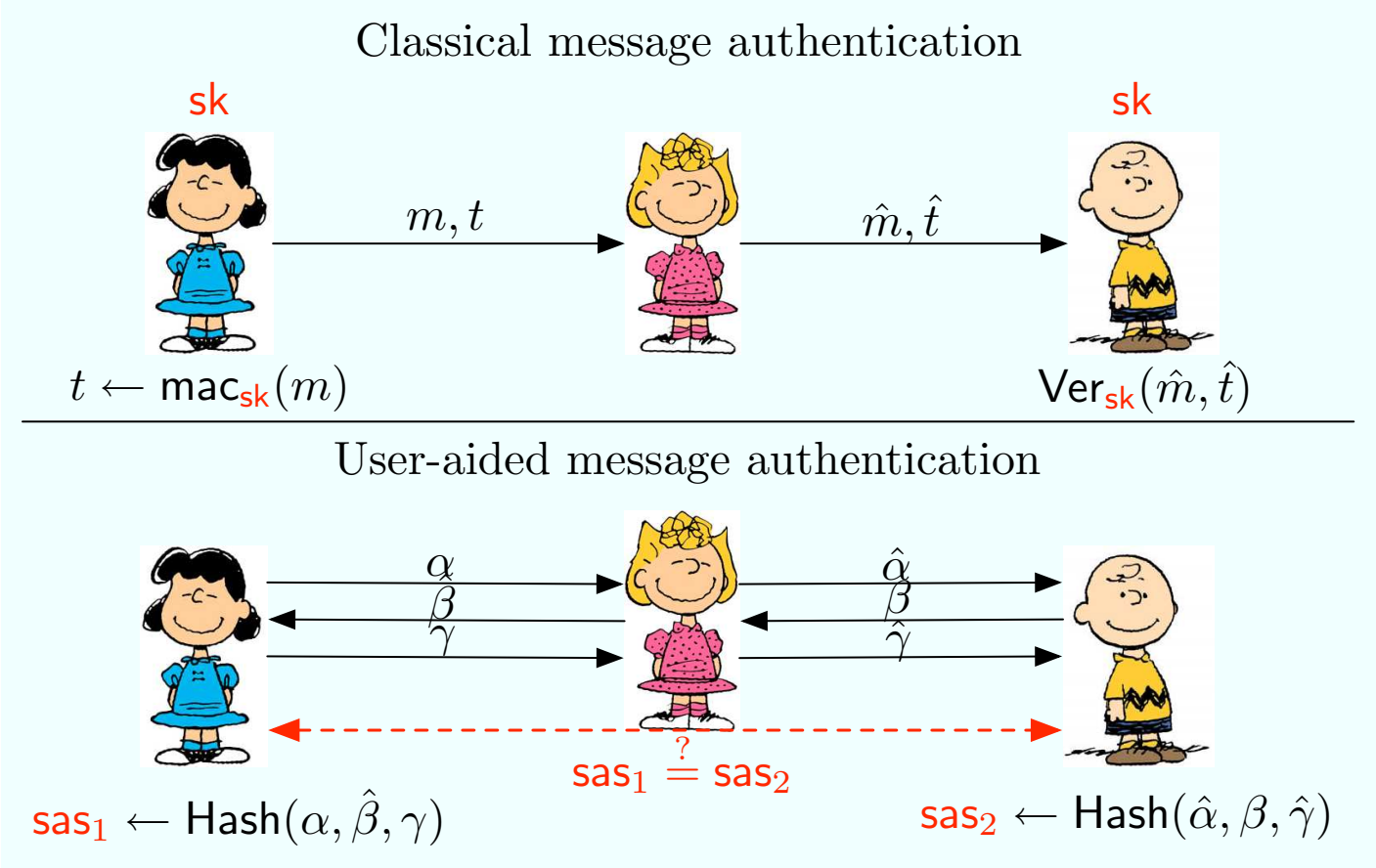
- ▷ Parties need a shared secret key to bootstrap other security mechanisms.
- ▷ Most key agreement protocols are secure against passive adversaries.
- ▷ It is infeasible to implement a global public-key infrastructure.

Maintaining public-key infrastructure can be difficult in practice:

- ▷ missing certificate chains in web browsers
- ▷ malicious alterations and additions of root certificates
- ▷ maliciously corrupted programs and computers

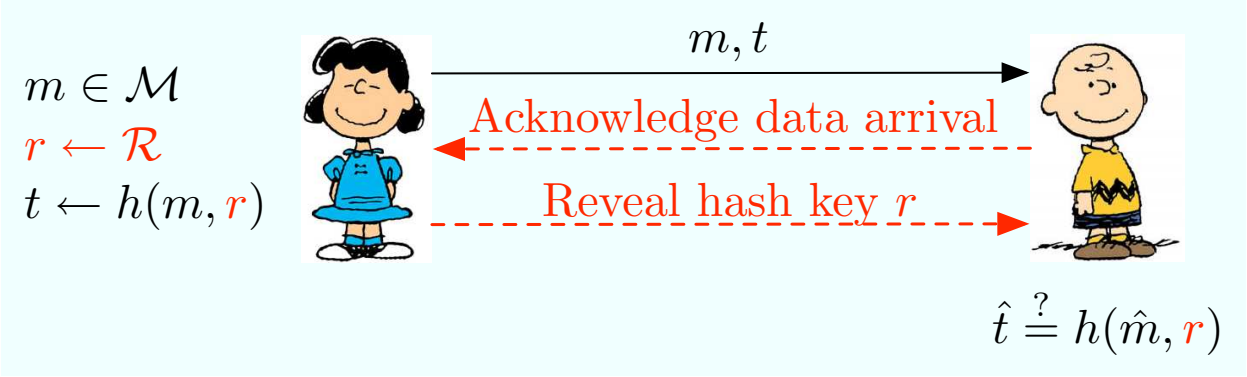
User should be able to detect malicious behaviour with high probability!

Security model



Out-of-band messages usually consist of 4–6 decimal digits

Simplified MANA-II protocol



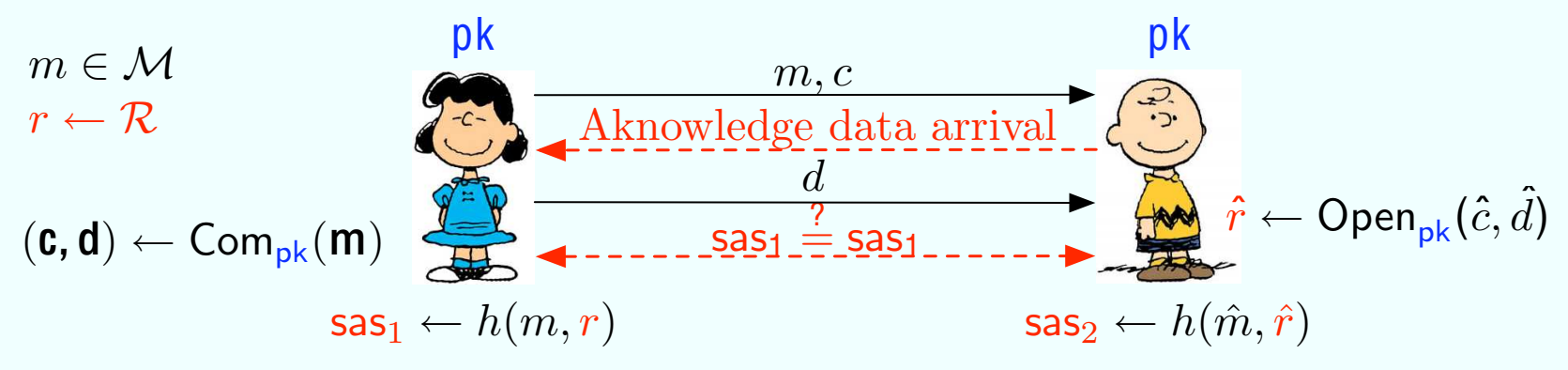
Due to temporal restrictions, we end up in the classical setting

- ▷ The secret key r is released only after the adversary has delivered $\hat{m}, \hat{t}$.
- ▷ The protocol is secure if h is almost universal hash function.

Due to the classical Simmon's lower bounds, we lose half bits:

$$\Pr\left[\text{Successful deception}\right] \geq \frac{1}{\sqrt{|\mathcal{R}|}} \ .$$

A quick fix



We can escape the lower bound if we use commitments to temporarily hide the hash key r until the adversary transfers $\hat{m}, \hat{c}$.

- ▷ The commitment scheme must be a non-malleable.
- ▷ Since we compare the hash values $h(m, r)$ and $h(\hat{m}, \hat{r})$ over the out-of-band channel, we can achieve the new lower bound

$$\Pr\left[\text{Successful deception}\right] \approx \frac{1}{|\mathcal{T}|} \ .$$

Elimination of notification messages

The notification message can be replaced with a random nonce $r_2 \leftarrow \mathcal{R}_2$:

- ▷ The nonce r_2 can be sent over the in-band channels.
- ▷ The nonce r_2 must completely re-randomise the final hash values sas_i.

The simplest option is to compute $\mathsf{sas}_i \leftarrow h(m, r_1) \oplus r_2$:

- ▷ The Vaudenay SAS protocol [Vau05].
- ▷ The optimised SAS-MCA protocol [PV06].

Alternatively, we can treat the nonce r_2 as a sub-key of the hash function and compute the final hash value as $\mathsf{sas}_i \leftarrow h(m, r_1, r_2)$:

- ▷ The MANA IV protocol [LN06].

Group setting

Message authentication for groups

Description: Each participant contributes an input m_i and its identity id_i. At the end of a successful protocol run all participants should obtain

- ▷ a list of messages $\boldsymbol{m} = (m_1, \ldots, m_n)$;
- ▷ a list of corresponding identities $\mathcal{G} = (\mathsf{id}_1, \ldots, \mathsf{id}_n)$.

An adversary succeeds in deception if two honest parties disagree on the output message $\boldsymbol{m}$ or on the group description $\mathcal{G}$.

Requirements:

- ▷ The number of in-band rounds should be minimal.
- ▷ All participants should obtain the same hash code!
- ▷ $\Pr[\text{Successful deception}] \approx \frac{1}{|\mathcal{I}|}$.

SAS-GMA protocol

First round. Each participant $\mathcal{P}_i$:

1. Generates a sub-key $r_i \leftarrow \mathcal{R}$.
2. Creates a commitment $(c_i, d_i) \leftarrow \mathsf{Com}_{\mathsf{pk}}(i, r_i)$.
3. Broadcasts m_i, c_i and receives messages $\hat{m}_j, \hat{c}_j$ from other members.

Second round. Each participant $\mathcal{P}_i$:

1. Releases its decommitment d_i and receives $\hat{d}_j$ from other members.
2. Reconstructs the corresponding sub-keys $(j, \hat{r}_j) \leftarrow \mathsf{Open}_{\mathsf{pk}}(\hat{c}_j, \hat{d}_j)$.
3. Assembles the output message $\hat{\boldsymbol{m}}$ and the group description $\hat{\mathcal{G}}$.
4. Computes the corresponding hash code $\mathsf{sas}_i \leftarrow h((\hat{\boldsymbol{m}}, \hat{\mathcal{G}}), \hat{r}_1, \ldots, \hat{r}_n)$

Third round. Protocol fails if the SAS messages sas_i are different.

The first simple substitution attack

The adversary ignores commitments and alters only messages m_i.

- ▷ Successful deception implies that two honest node $\mathcal{P}_\alpha$ and $\mathcal{P}_\beta$ have same hash values $\mathsf{sas}_\alpha = \mathsf{sas}_\beta$ but different outputs $(\hat{\boldsymbol{m}}_\alpha, \hat{\mathcal{G}}_\alpha) \neq (\hat{\boldsymbol{m}}_\beta, \hat{\mathcal{G}}_\beta)$.
- ▷ Since there is at least one varying sub-key $r_k \leftarrow \mathcal{R}$, we can upper bound the successful deception probability by

$$\Pr\left[r_k \leftarrow \mathcal{R} : h(x_0, \ldots, r_k, \ldots) = h(x_1, \ldots, r_k, \ldots)\right] \leq \varepsilon_{\mathrm{u}} \ ,$$

where $x_0 \neq x_1$, varying sub-keys r_k can be located in two different places and other sub-keys can be arbitrarily fixed.

The hash function h must be ε_{u}-almost universal w.r.t. each sub-key pair.

The second simple substitution attack

The adversary treats commitments as black boxes but still substitutes some of them with different commitments.

- ▷ If the sender and the receiver of this commitment are honest, then the corresponding sub-keys r_k and $\hat{r}_k$ are independent.
- ▷ As a result, we can upper bound the deception probability by

$$\Pr\left[r_k \leftarrow \mathcal{R} : h(\ldots, r_k, \ldots) = \mathsf{sas}\right] \leq \varepsilon_{\mathrm{r}} \ ,$$

where all inputs except r_k can be arbitrarily fixed.

The hash function h must be ε_{r}-almost regular w.r.t. each sub-key.

The actual security guarantees

Let n be the maximal size of the group $\mathcal{G}$ and h be ε_{u}-almost universal w.r.t. each sub-key pair and ε_{r}-almost regular w.r.t. each sub-key. Then for any t there exists $\tau = t + \mathrm{O}(1)$ such that if the commitment scheme is $(\tau, \varepsilon_{\mathrm{b}})$-binding and $(\tau, \varepsilon_{\mathrm{nm}})$-non-malleable, then the SAS-GMA protocol is $(t, n \cdot \varepsilon_{\mathrm{nm}} + \varepsilon_{\mathrm{b}} + \max\{\varepsilon_{\mathrm{u}}, \varepsilon_{\mathrm{r}}\})$-secure in the stand-alone model.

Intuition behind the proof: The non-malleability of commitments allows us to reduce any attack to the simple substitution attacks presented above.

Key management is easy

Main principle. Use a key agreement protocol that is secure against passive adversaries and detect active attacks with user-aided data authentication.

- ▷ If we authenticate the whole protocol transcript, then each participant knows that his or her messages have reached the target.
- ▷ If nobody complains, then the adversary was passive.

If we combine the SAS-GMA protocol with the Burmester-Desmedt key agreement protocol, we obtain three-round key agreement protocol.

Another trick. If we authenticate the public keys of group members, then we can form sub-groups without re-running the SAS-GMA protocol.

Final comments

The non-malleability requirement is essential. However, the required security level is low, as we are destined to fail with probability 10^{-4}–10^{-6}.

- ▷ Hash commitments are sufficient in practice.
- ▷ The use of cryptographically secure commitments is overkill.

Since the SAS-GMA does not rely on shared secrets, we can employ the protocol in any computational context as long as

- ▷ participants can separate protocol messages from other messages;
- ▷ the SAS message uniquely determines the protocol instance.

Questions? Answers?