Secure Communication using Authenticated Channels

Sylvain Pasini

Private PhD Defense

June 17th, 2009

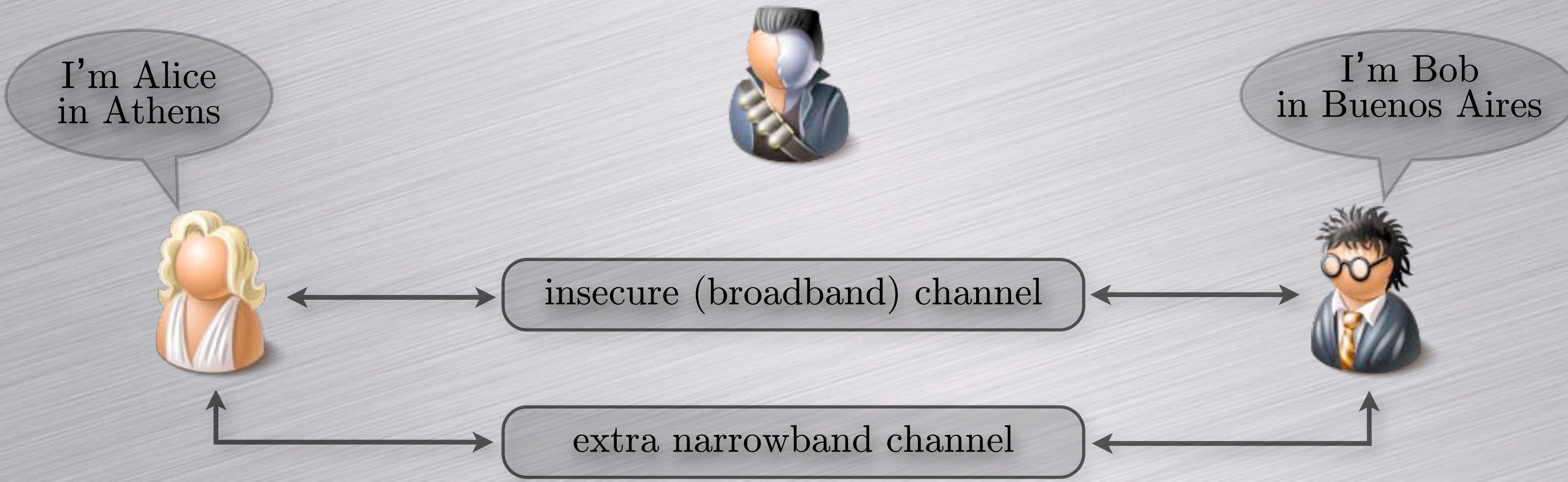
Outline

- Motivation
- SAS-based cryptography
    - Security model
    - Generic attacks, optimality
    - Overview of different protocols
- Signature schemes
    - Privacy protection
    - (Strengthening hash-and-sign implementations)

Motivation

How to establish a secure communication?

Hypothesis



Goal

Communicate securely

Symmetric cryptography

Need to share a secret key k.

Symmetric encryption is secure and fast.

Secret key exchange in reality

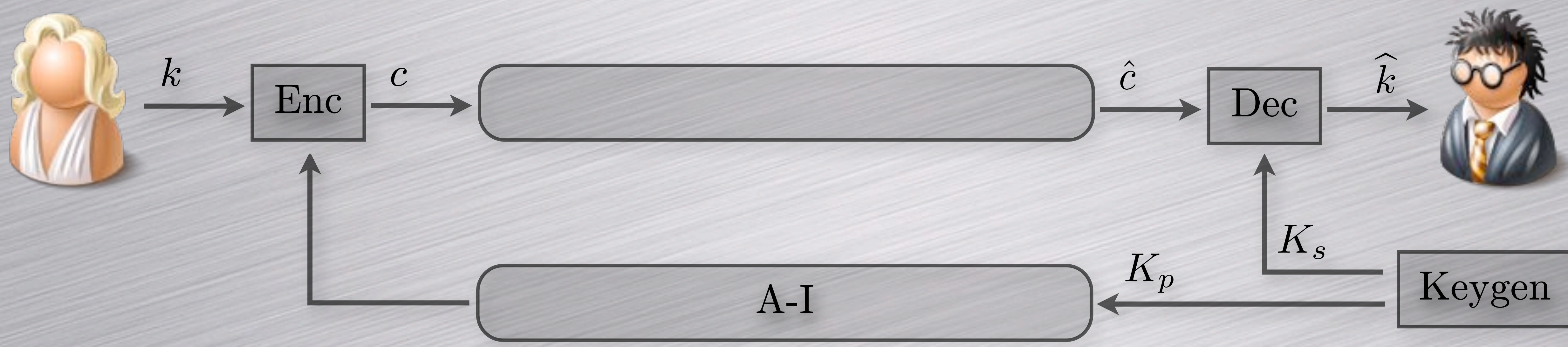
	Encounter	Telephone	Voice mail	E-mail
Confidentiality	☺			
Authenticity	☺	☺	☺	
Low cost		☺	☺	☺
Availability			☺	☺
Speed rate				☺

Confidential channel: expensive and bad availablility.

Can we avoid confidentiality and only use authentication?

Asymetric cryptography

Semi-authenticated key transfer:

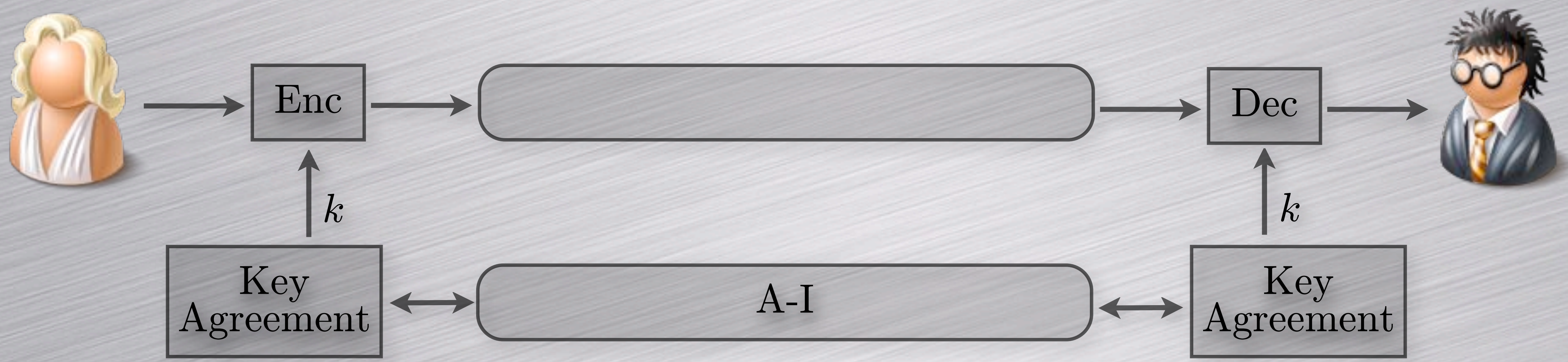


Confidentiality is no longer required.

Authentication is enough.

Key agreement

Merkle-Diffie-Hellman model:



Confidentiality is no longer required.

Authentication is enough.

In a nutshell

- Goal:
  - Alice and Bob want to communicate securely
- Hypothesis:
  - no prior exchanged data (no PSK, no PKI)
- A secure channel can be setup with a secret key
- A secret key can be setup by
  - exchanging (and authenticating) a public-key
  - or running an authenticated key agreement

In a nutshell (2)

Claim

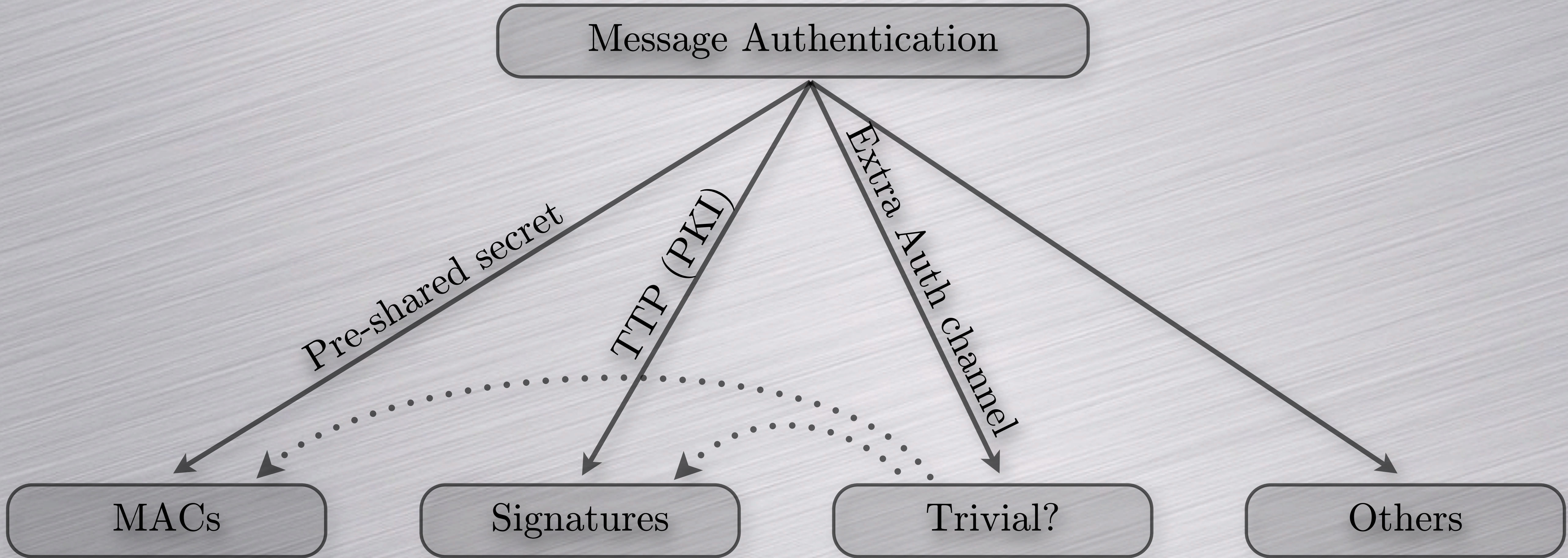
As long as parties are able to authenticate data,

they are able to setup a secure communication.

Motivation

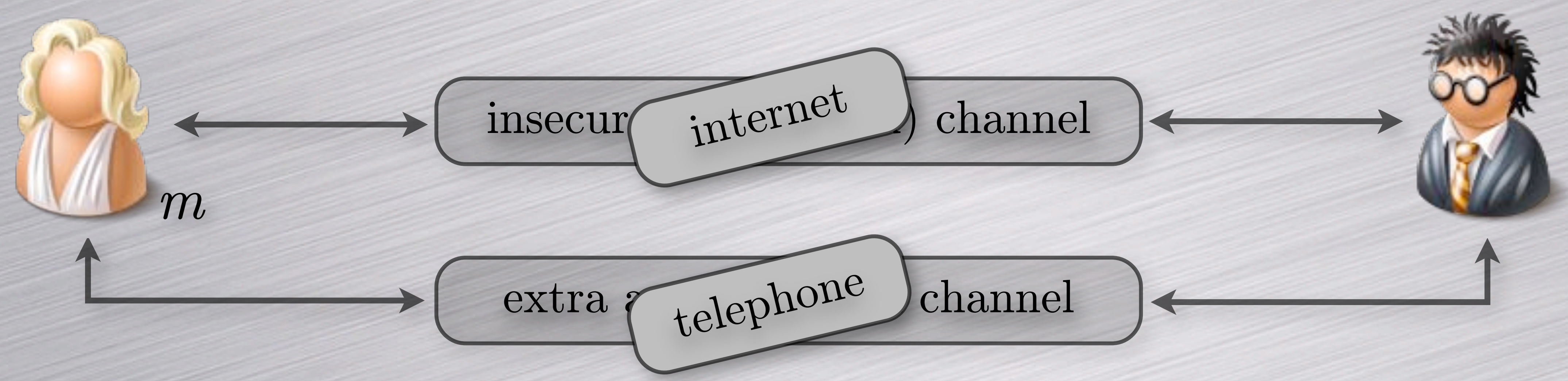
How to authenticate messages?

Authentication Overview



Trivial solution

Goal: authenicate the message m.



User-friendly...

Example of an RSA 1024-bit key:

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEApZTXilQgosFxe
vR9ewub/qE1/BoHXCkpzWwopTHkiY2e8pMxMXOc/
DzKV0qgsdC3X9pQODRy+awoANAgttPX
h6JM4ZlYgaEN6azJSyrK0SlOLDn
+YmjjhaKEn1ufLbroQ6Cpg0lj3lXvHEN52P32IfhY08ivC
0pBmO4Y eyErBiE=

By telephone... good luck!

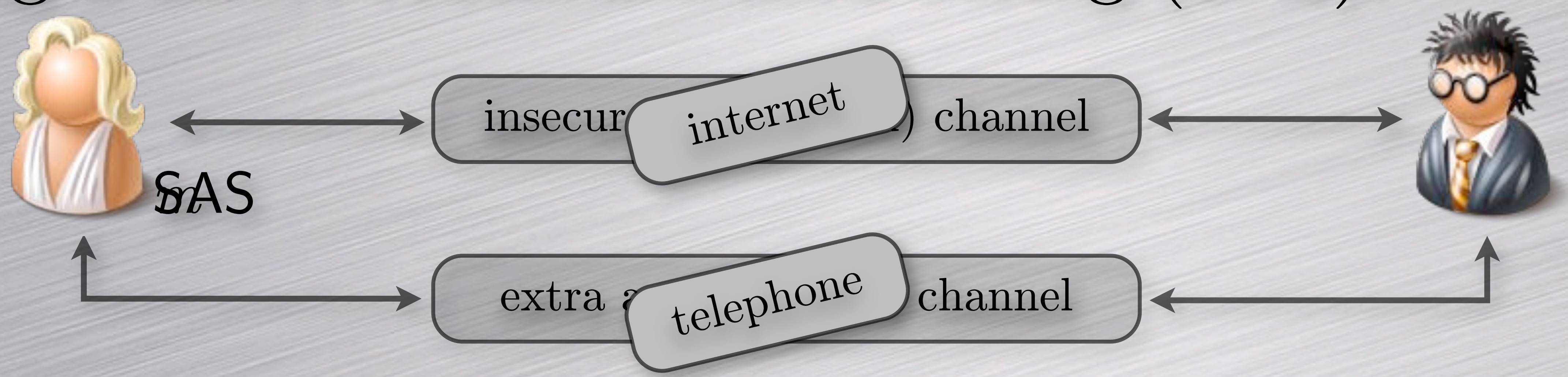
Objective

Design user-friendly protocols.
So, authenticated data should be as short as possible.

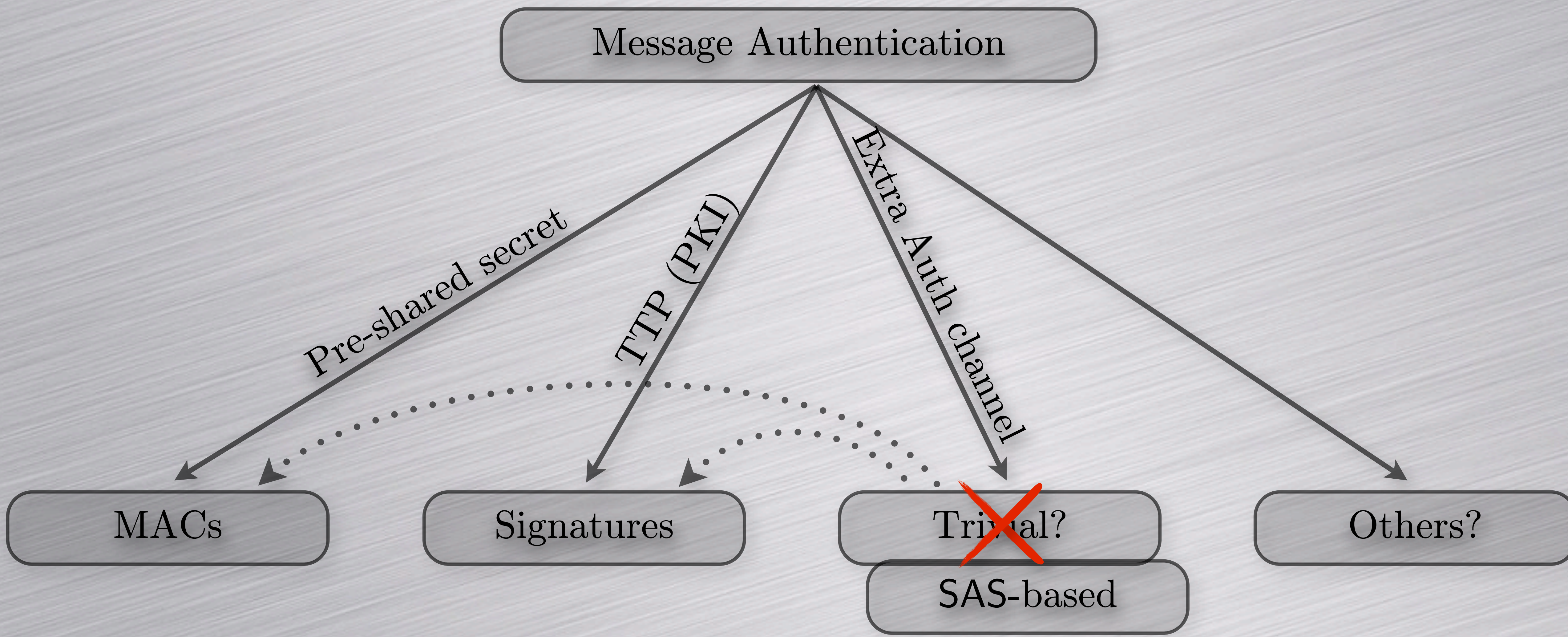
In practice...

Goal: authenicate the message m.

Using a Short Authenticated String (SAS):



Authentication Overview

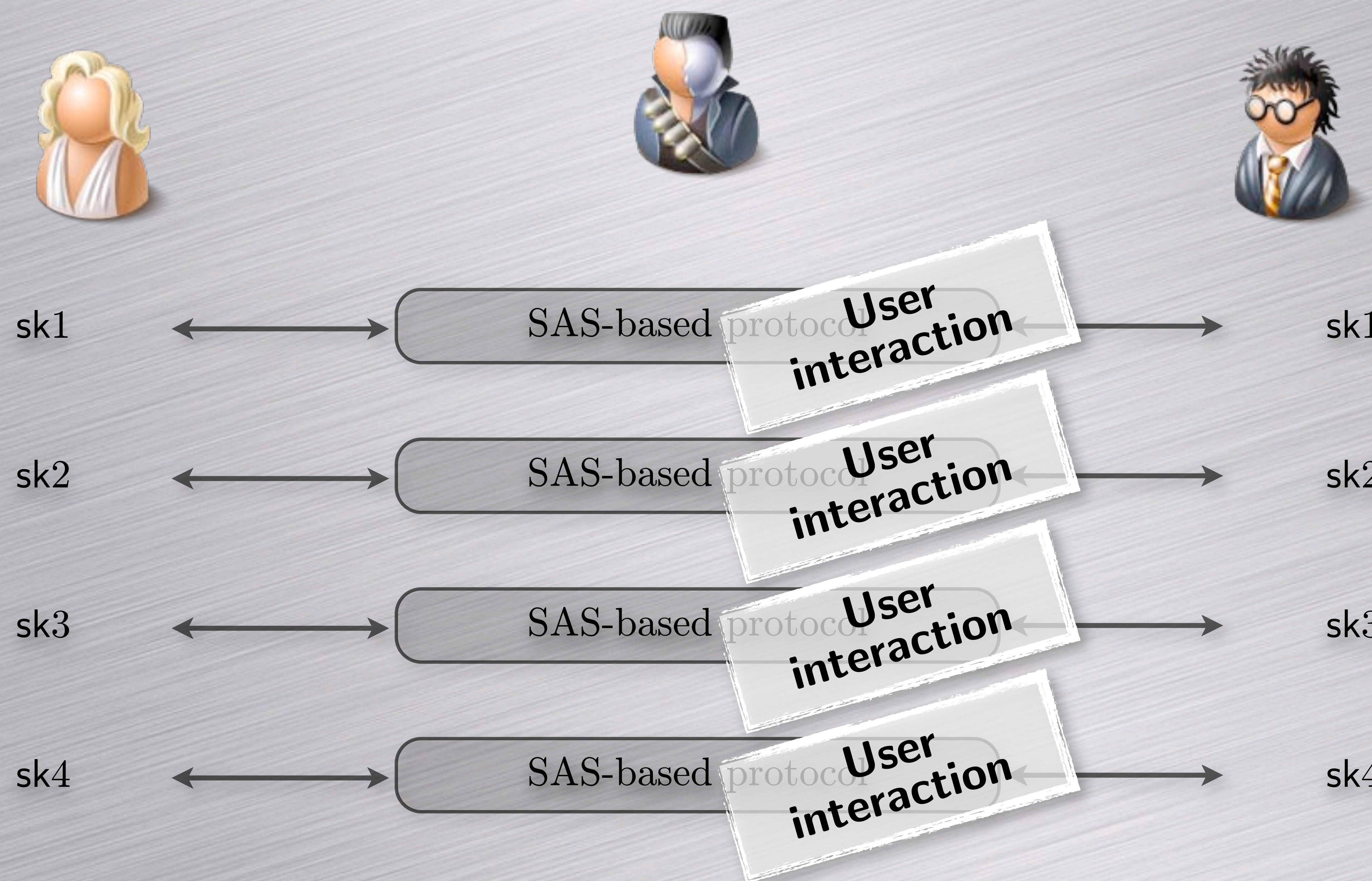


- PSK

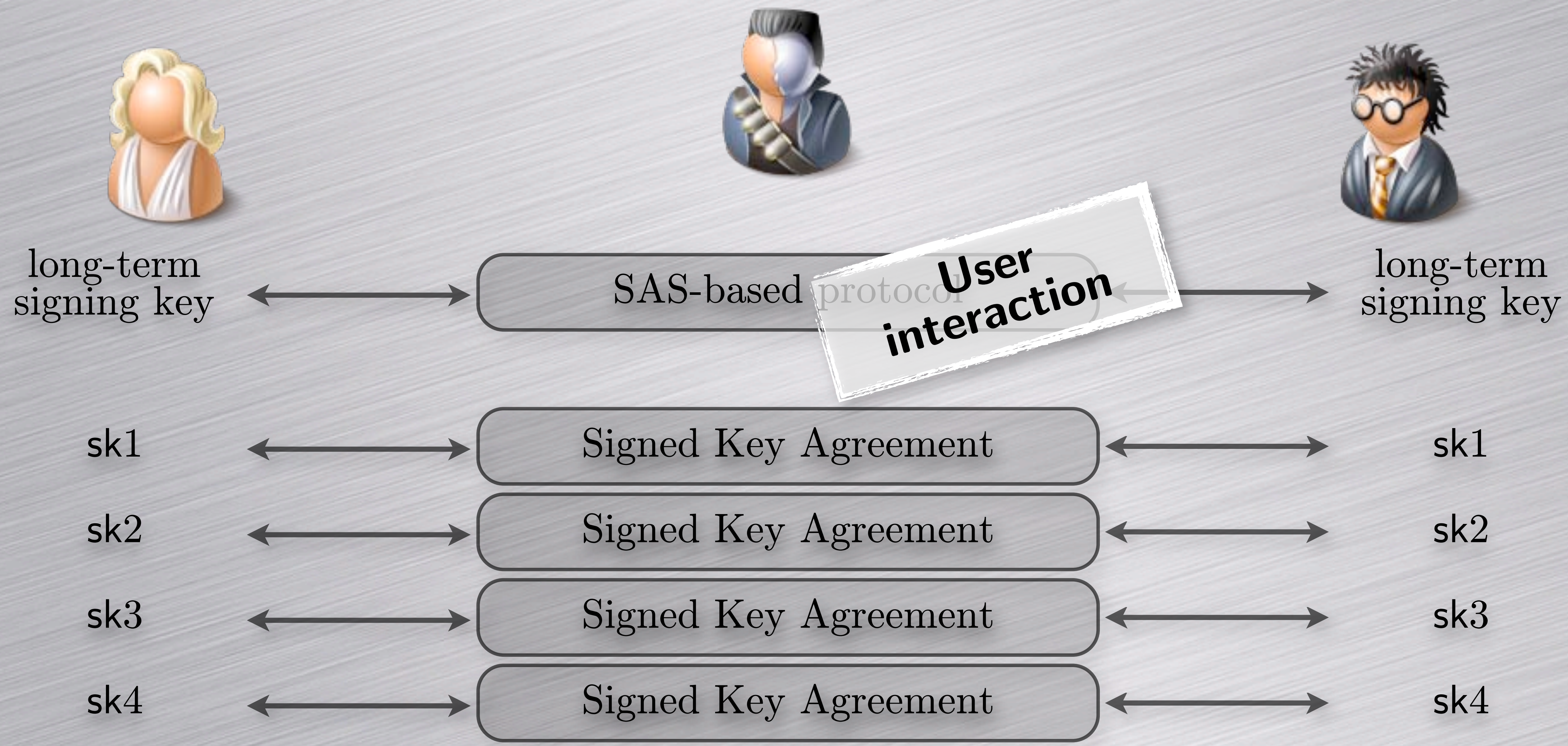
- TTP
+ no user interaction

+ no PSK
+ no TTP
- user interaction

Using SAS-based Protocols



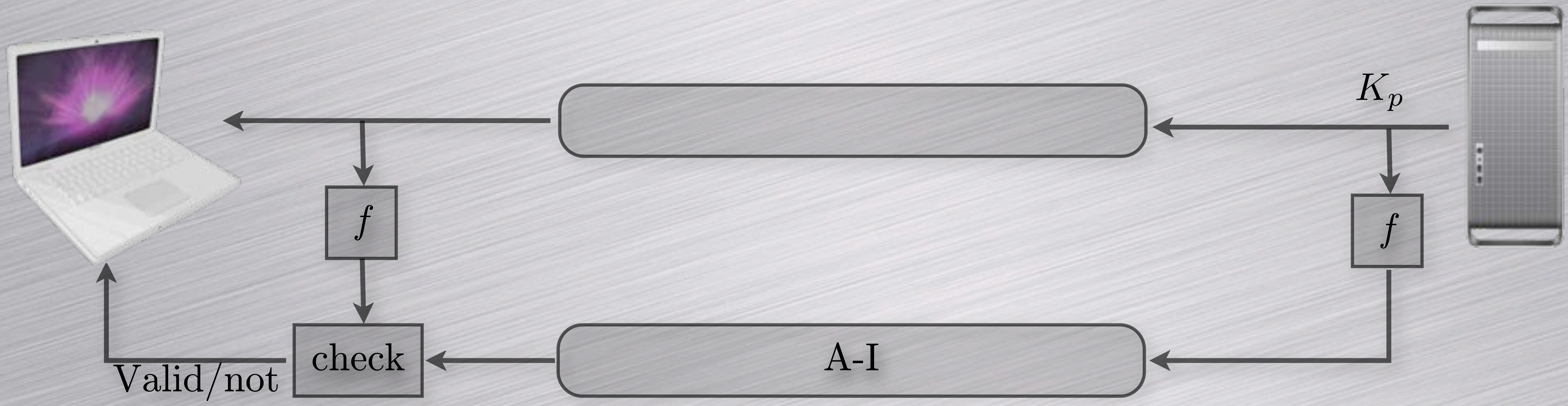
Trusted Setup



+ no PSK, + no TTP, + one user interaction

Example: Secure Shell (SSH)

Goal: authenticate the server's public key.



Check done the first time only (trusted setup)

The fingerprint is of the form

bc:a1:12:30:bc:17:08:eb:31:43:eb:e1:15:12:ca:1a (hexa)

It is better, but who **really** check this?

SAS-based Cryptography

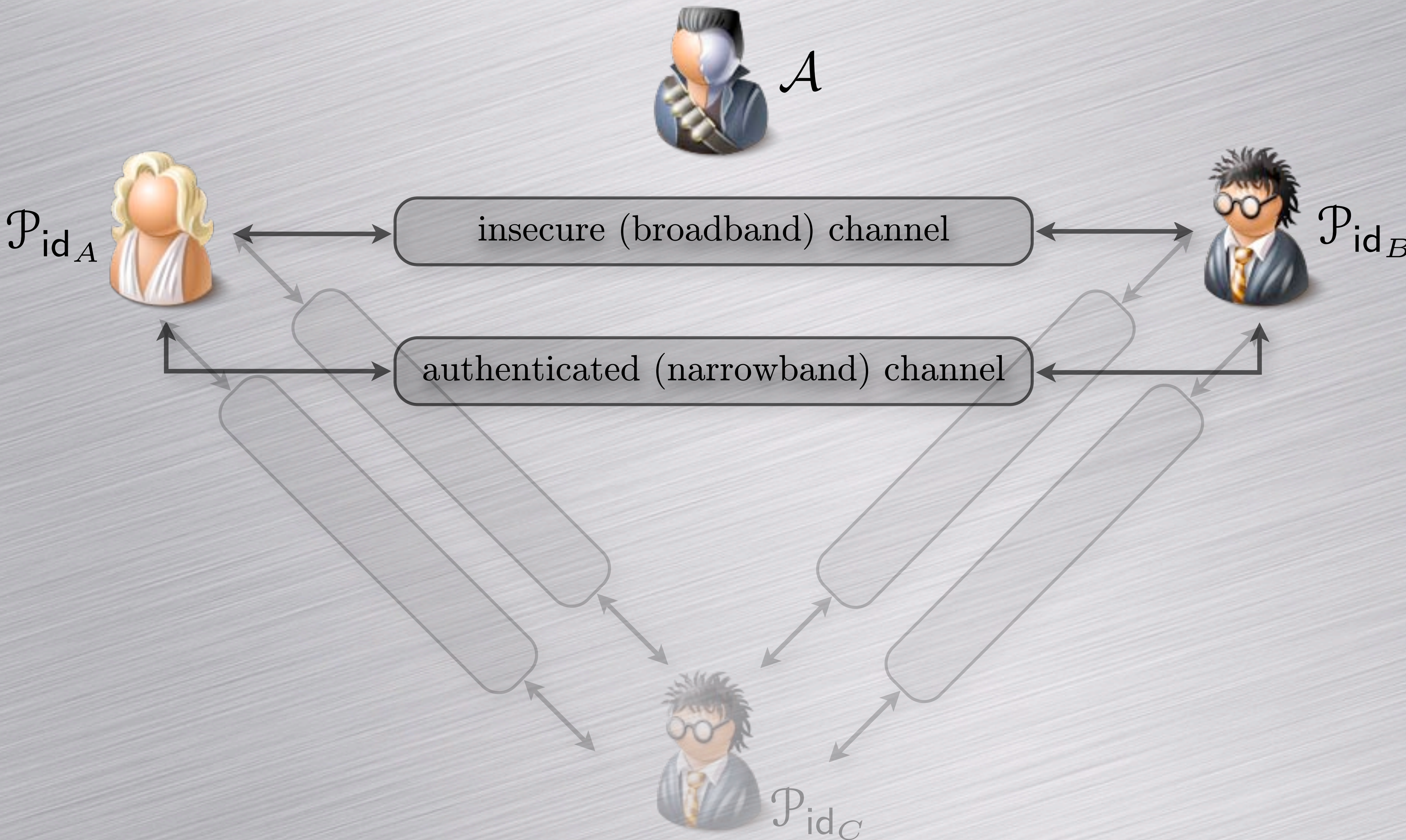
SAS also known as...

- MANual Authentication (MANA)
  - Gehrmann, Mitchell, Nyberg, and Laur.
- Short Authenticated String (SAS)
  - Vaudenay and Pasini.
- Two-channel cryptography
  - Mashatan and Stinson.
- User-aided data authentication
  - Peyrin, Vaudenay, and recently Laur and Pasini.

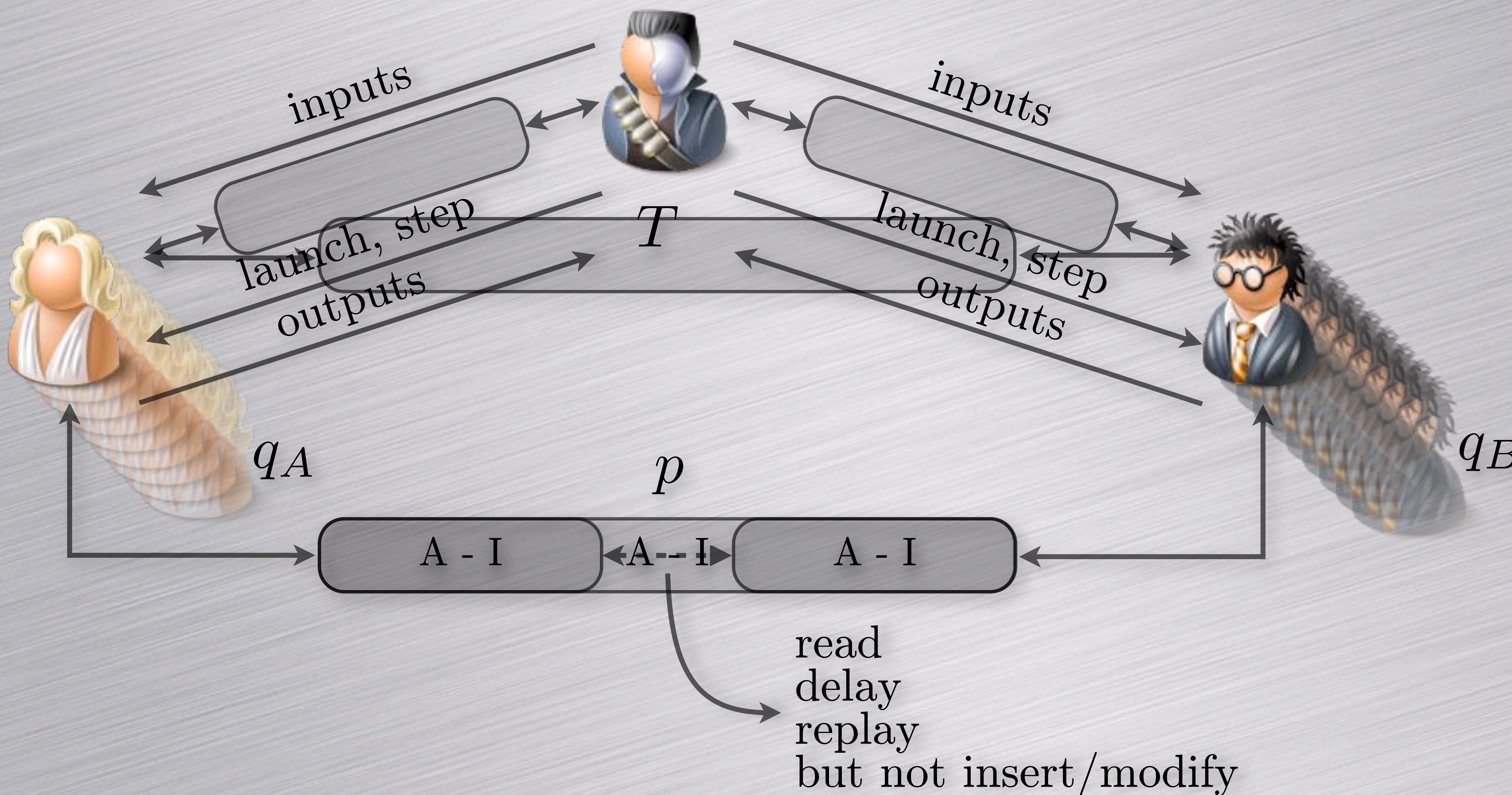
SAS-based Cryptography

Security model

Network model

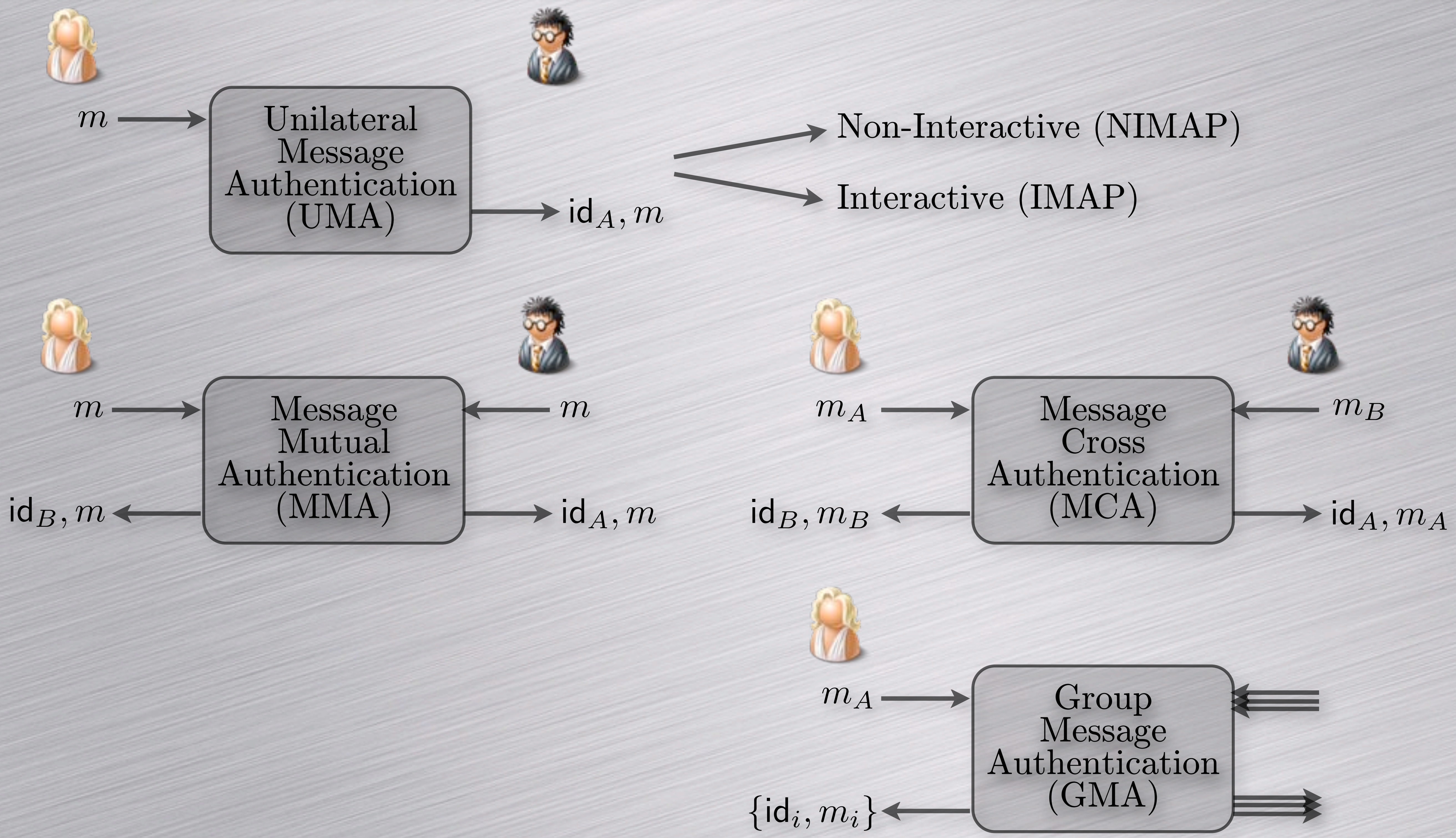


Adversarial model



The attack cost is measured by (T,q,p).

Message Authentication



SAS-based Cryptography

Generic attacks, optimality

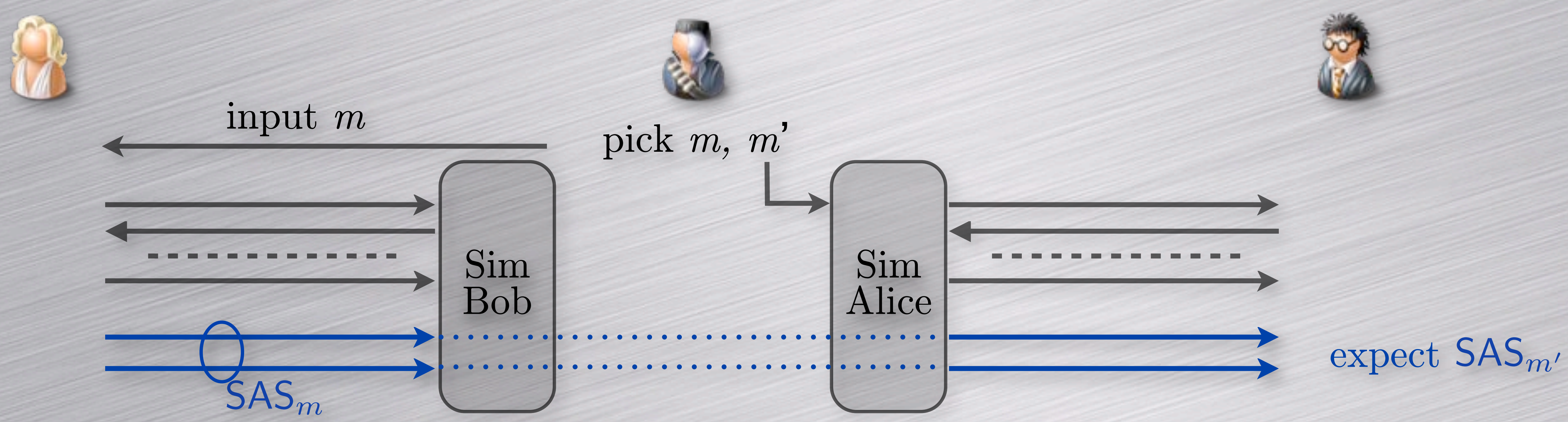
What is the maximal security?

- Suppose
  - no limit on the insecure channel
  - limit on the authenticated channel: k bits
  - fixed bound on complexities q and T

- The protocol is at least (T,q,p)-secure
- Every protocols are at most *(T,q,P)*-secure
- When $p \rightarrow P$, the protocol is said ***optimal***

Generic One-shot Attack

- Consider a generic UMA
  - goal: impersonation of Alice
  - one protocol run only



$$p = \Pr[\text{success}] = \Pr[\mathsf{SAS}_m = \mathsf{SAS}_{m'} \text{ and } m \neq m']$$

Generic One-shot Attack (2)

$$p = \Pr[\text{success}] = \Pr[\mathsf{SAS}_m = \mathsf{SAS}_{m'} \text{ and } m \neq m']$$

$$\begin{aligned}\Pr[\mathsf{SAS}_m = \mathsf{SAS}_m] &= \Pr[\mathsf{SAS}_m = \mathsf{SAS}_m \text{ and } m \neq m'] \\ &\quad + \Pr[\mathsf{SAS}_m = \mathsf{SAS}_m \text{ and } m = m']\end{aligned}$$

$$\begin{aligned} p &= \Pr[\mathsf{SAS}_m = \mathsf{SAS}_{m'}] - \Pr[\mathsf{SAS}_m = \mathsf{SAS}_{m'} \text{ and } m = m'] \\ &\geq \underline{\Pr[\mathsf{SAS}_m = \mathsf{SAS}_{m'}]} - \underline{\Pr[m = m']} \end{aligned}$$

$$\begin{aligned} &\geq \Pr[\mathsf{SAS}_m = \mathsf{SAS}_{m'} | \mathcal{D} \text{ is uniform}] \\ &= \frac{1}{n} \end{aligned}$$

$$2^{-t}$$

$$p \geq \frac{1}{n} - 2^{-t}$$

Optimal SAS Distribution

Set of possible SAS: $\mathcal{S} = \{s_1, \ldots, s_n\}$

Let p_i denote $\Pr[\mathsf{SAS} = s_i]$

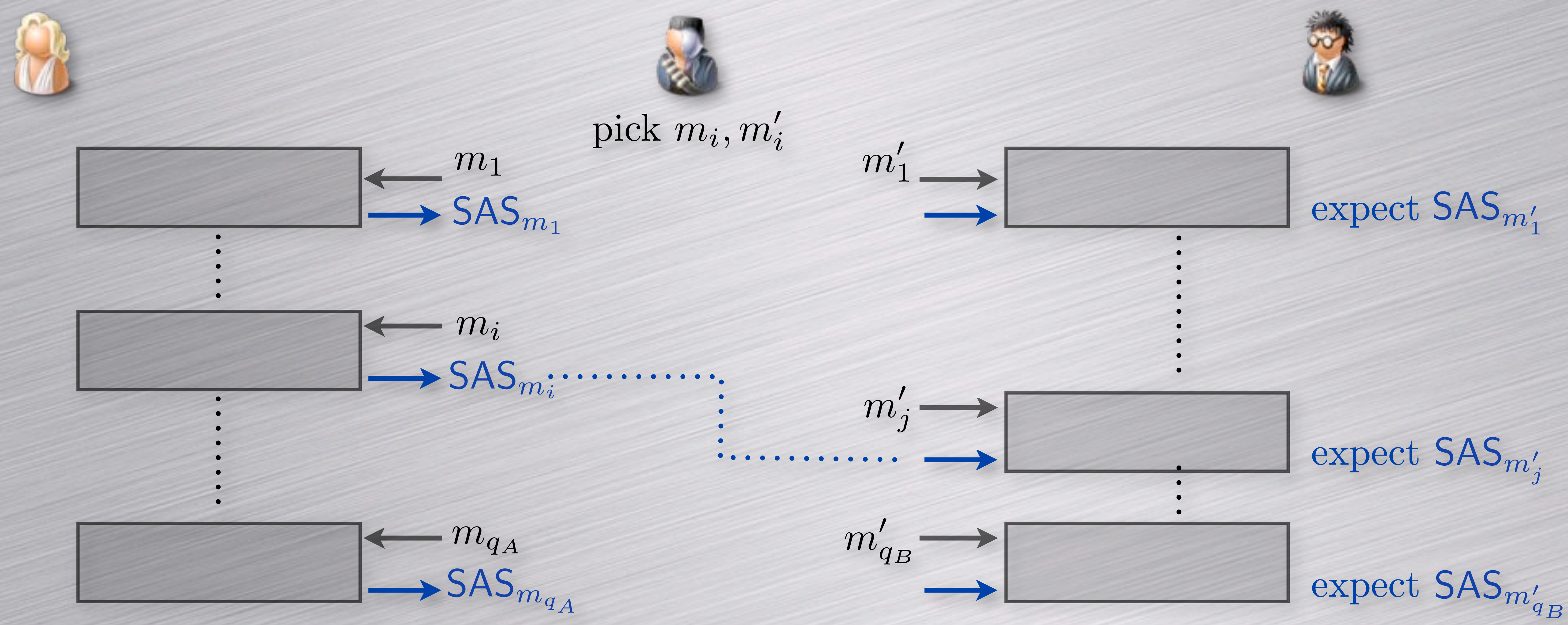
Let $\mathsf{SAS}_1, \mathsf{SAS}_2 \in_{\mathcal{D}} \mathcal{S}$

$$p = \Pr[\mathsf{SAS}_1 = \mathsf{SAS}_2] = \sum_{i=1}^{n} p_i^2$$

$\mathcal{D}$ is uniform	$\mathcal{D}$ is non-uniform
$p_i = 1/n$	$p_i = 1/n + \delta_i$ with $\sum_{i=1}^{n} \delta_i = 0$
	$p = \sum (1/n + \delta_i)^2$
$p = \sum (1/n)^2$	$= \sum (1/n)^2 + 1/n \sum \delta_i + \sum \delta_i^2$
$= 1/n$ optimal	$> 1/n$

Generic Multi-shot Attack

Now, the adversary may use several protocol runs...



$$p = \Pr[\exists i, j \text{ such that } \mathsf{SAS}_{m_i} = \mathsf{SAS}_{m'_j} \text{ and } m_i \neq m'_j]$$

Generic Multi-shot Attack (2)

$$
\begin{aligned}
p &= \Pr[\exists i, j \text{ such that } \mathsf{SAS}_{m_i} = \mathsf{SAS}_{m'_j} \text{ and } m_i \neq m'_j] \\
&\geq \underline{\Pr[\exists i, j \text{ such that } \mathsf{SAS}_{m_i} = \mathsf{SAS}_{m'_j}]} \\
&\quad \underline{- \Pr[\forall k, \ell \text{ such that } \mathsf{SAS}_{m_i} = \mathsf{SAS}_{m'_j} : m_i = m'_j]}
\end{aligned}
$$

$$
\begin{aligned}
&\leq \Pr[\forall k, \ell : m_i = m'_j] \\
&\leq q_A q_B 2^{-t}
\end{aligned}
$$

$$
\begin{aligned}
&\geq \Pr[\exists i, j \text{ such that } \mathsf{SAS}_{m_i} = \mathsf{SAS}_{m'_j} | \mathcal{D} \text{ is uniform}] \\
&\geq 1 - \exp^{-\frac{q_A q_B}{n}}
\end{aligned}
$$

$$
p \geq 1 - \exp^{-\frac{q_A q_B}{n}} - q_A q_B 2^{-t}
$$

Optimal SAS Split

1 k-bit SAS	k 1-bit SAS

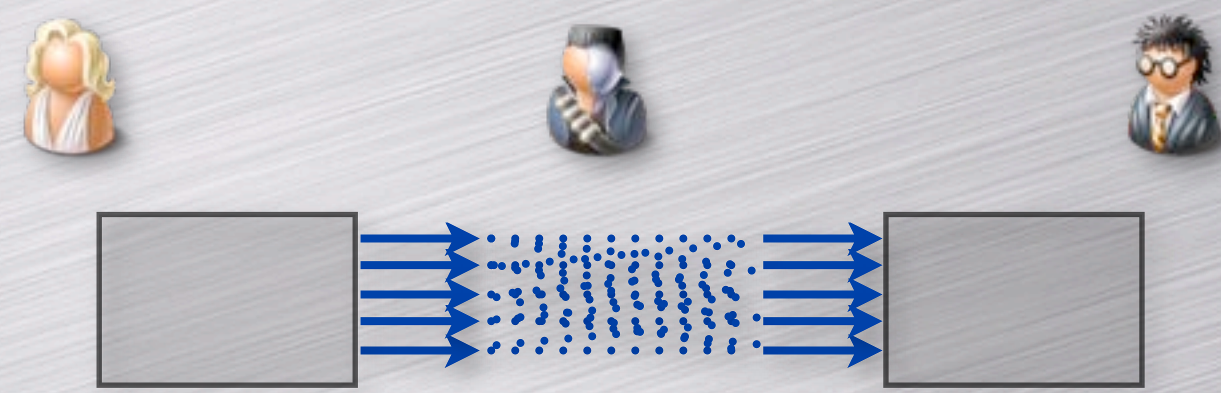
1 k-bit SAS

Size of SAS catalog: 2^k

$$p \approx \frac{1}{2^k}$$

k 1-bit SAS

Size of SAS catalog: 2

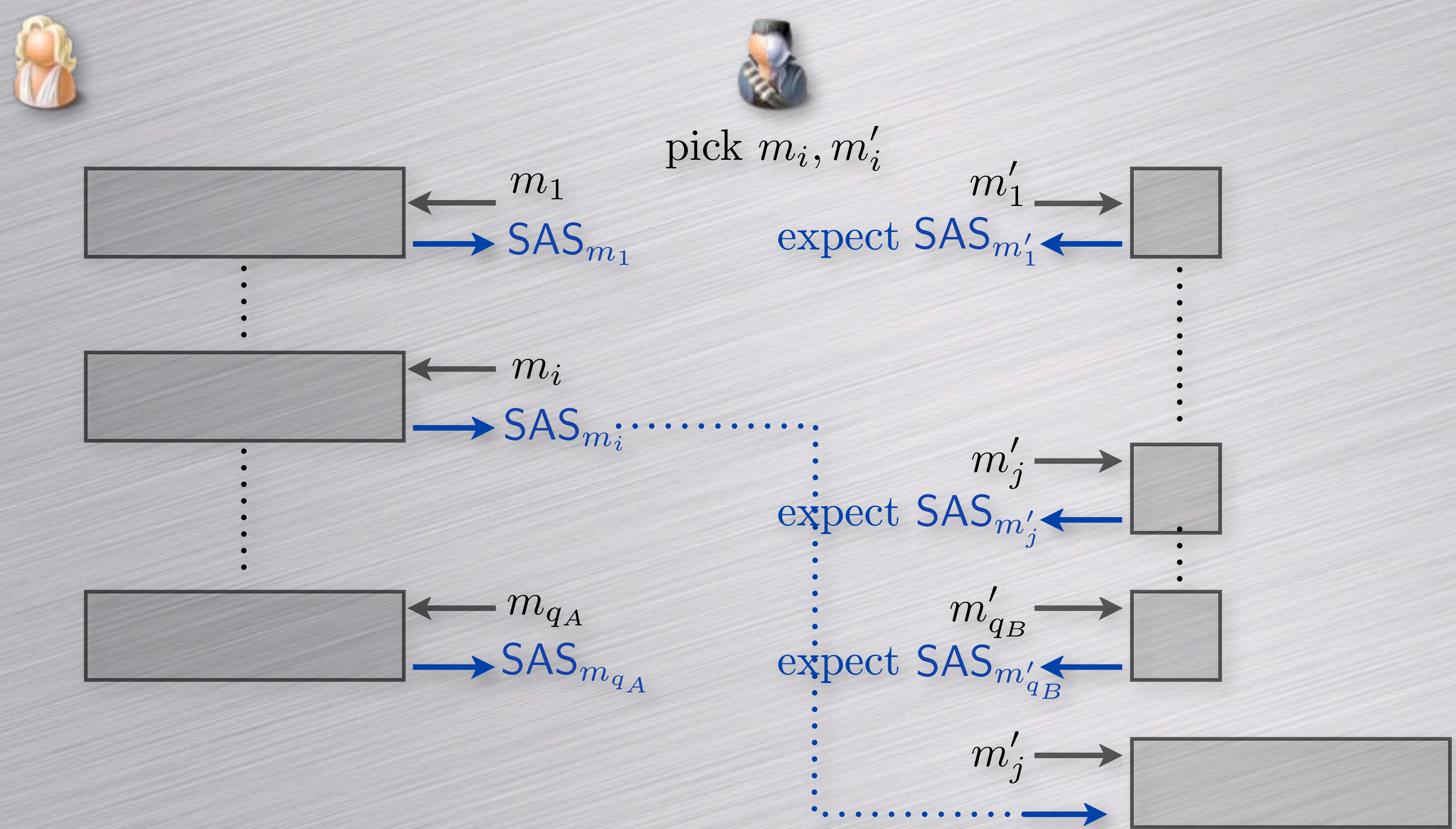


If Alice release a '0' and a '1', then the attack succeeds...

$$p \approx 1$$

Generic Multi-shot Attack (NI)

Now, the protocol is **non-interactive**.

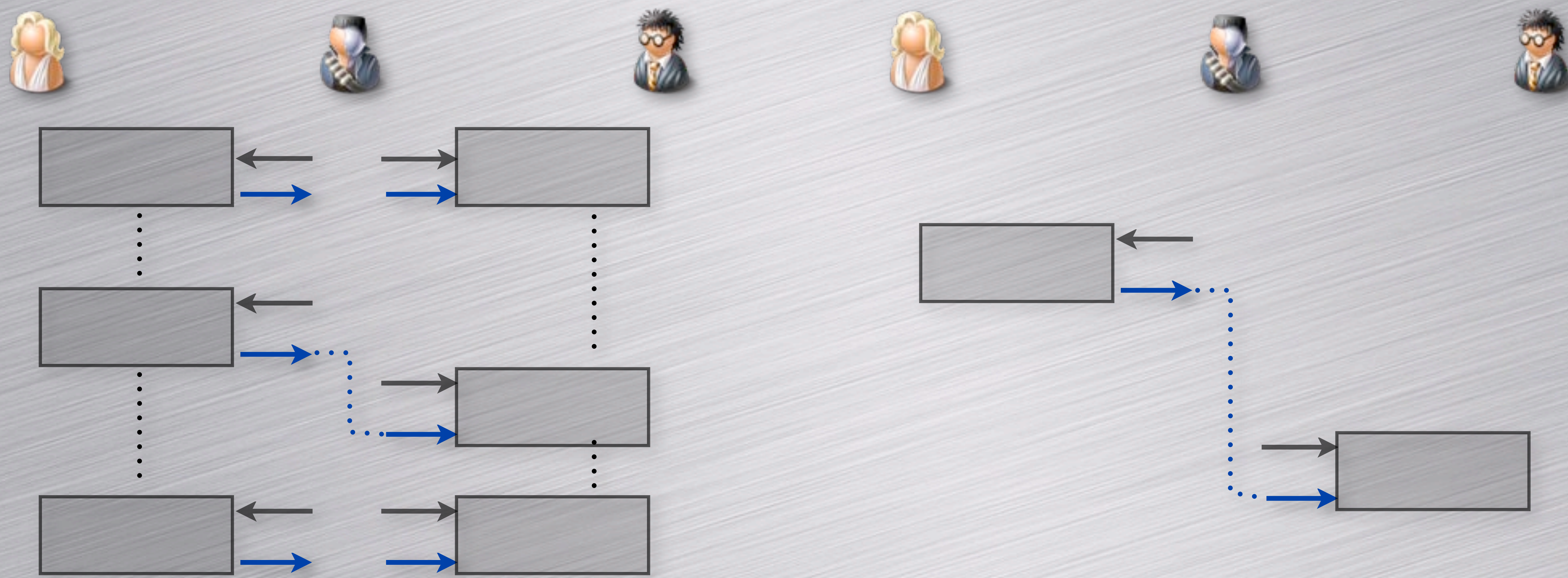


q_B is now an offline complexity.

Overview of Generic Attacks

- Optimal SAS are
  - uniformly distributed
  - sent in one piece
- Generic attack against *any* UMAP:
  - one-shot attack with $p \approx \frac{1}{n}$
  - multi-shot attack with $p \approx 1 - \exp^{-\frac{q_A q_B}{n}}$
  - [NIMAP] multi-shot attack with $p \approx 1 - \exp^{-\frac{q_A T}{n}}$
  - If the best attack is the generic one: **optimal**
- [LN06] an optimal protocol has at least 3 (interactive) moves

One-shot versus Multi-shot [Vau05]



$p \geq p_{\mathsf{ms}}$ $\boxed{p_{\mathsf{ms}} \leq q_A q_B p_{\mathsf{os}}}$ $p_{\mathsf{os}} \geq \dfrac{p_{\mathsf{ms}}}{q_A q_B}$

$(T, 2, p_{\mathsf{os}})$ implies $(T,\ q_A + q_B,\ q_A q_B p_{\mathsf{os}})$

SAS-based Cryptography

Unilateral Message Authentication Protocols

CRHF-based NIMAP [BSSW02]

Used in SSH, GPG, ...

Input: m

$$\xrightarrow{\quad m \quad}$$

$$h = H(m) \xrightarrow{\quad h \quad} \text{check } h \stackrel{?}{=} H(\widehat{m})$$

CRHF-based NIMAP [BSSW02]

Known message attack:

Input: m

$m \rightarrow$ 2nd preimg $\xrightarrow{m'}$

$h = H(m) \xrightarrow{h}$ check $h \stackrel{?}{=} H(m')$

H should be WCR (80 bits)

Chosen message attack:

$\xleftarrow{m}$ collision

$\rightarrow$ collision $\xrightarrow{m'}$

$h = H(m) \xrightarrow{h}$ check $h \stackrel{?}{=} H(m')$

H should be CR (160 bits)

PV-NIMAP [PV06a]

CRHF-based NIMAP:

collision attack due to a predictible *H(m)*

Main idea of PV-NIMAP

Avoid the authenticated message to be predicible.

Input: m

pick r, $c = \mathsf{R}(m,r)$	c,m,r	check $\widehat{c} \stackrel{?}{=} \mathsf{R}(\widehat{m},\widehat{r})$
$(c,d) = \mathsf{commit}(m)$	$\xrightarrow{c,d}$	$\widehat{m} = \mathsf{open}(\widehat{c},\widehat{d})$
$h = H(c)$	$\xrightarrow{h}$	check $h \stackrel{?}{=} H(\widehat{c})$

PV-NIMAP: intuitive security

Case $\hat{c} \neq c$:

m

$(c, d) = \mathsf{commit}(m) \xrightarrow{c,d}$ $\xrightarrow{\widehat{c},\widehat{d}} \widehat{m} = \mathsf{open}(\widehat{c}, \widehat{d})$

$h = H(c) \xrightarrow{h}$ check $h \stackrel{?}{=} H(\widehat{c})$

Require to find a 2nd preimage on H.

Case $\hat{c} = c$:

m

$(c, d) = \mathsf{commit}(m) \xrightarrow{c,d}$ $\xrightarrow{c,\widehat{d}} \widehat{m} = \mathsf{open}(c, \widehat{d})$

$h = H(c) \xrightarrow{h}$ check $h \stackrel{?}{=} H(c)$

Require to defeat the binding property.

PV-NIMAP [PV06a]

Theorem

Assume a (T, ε_c)-binding commitment

a (T, ε_h)-WCR function

Then, PV-NIMAP is $(T,\ q_A + 1,\ q_A(\varepsilon_c + \varepsilon_h))$

- c is sent over the insecure channel
  - ε_c as small as desired
- WCR-resitance on H (not CR)

What about interactivity?

- Interactivity allows to avoid offline attacks.
- As a consequence, SAS are shorter.
- As example, Vau-SAS-IMAP:

Input: m

pick $R_A \in \{0,1\}^k$ | pick $R_B \in \{0,1\}^k$

$(c,d) = \mathsf{commit}(m, R_A) \xrightarrow{m,c}$

$\xleftarrow{R_B}$

$\xrightarrow{d}$ $\widehat{R}_A = \mathsf{open}(\widehat{m}, \widehat{c}, \widehat{d})$

$\mathsf{SAS} = R_A \oplus \widehat{R}_B \xrightarrow{\mathsf{SAS}}$ check $\mathsf{SAS} \stackrel{?}{=} \widehat{R}_A \oplus R_B$

Vau-SAS-IMAP [Vau05]

Theorem

Assume a (T, ε_c)-equivocable or extractable commitment

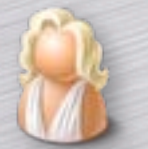
Then, Vau-SAS-IMAP is $\left(T,\ q_A + q_B,\ q_A q_B (2^{-k} + \varepsilon_c)\right)$

- Optimal
- Vau-SAS-IMAP requires 20-50-bit SAS
- PV-NIMAP requires 100-bit SAS

SAS-based Cryptography

Bilateral Message Authentication Protocols

Vau-SAS-MCA



Alice		Bob
Input: m_A		Input: m_B
pick $R_A \in \{0,1\}^{\cancel{k}\,\ell}$		pick $R_B \in \{0,1\}^k$
$(c_A, d_A) = \mathsf{commit}(\cancel{0}\|m_A, R_A)$	$\xrightarrow{m_A,\, c_A}$	
	$\xleftarrow{m_B,\, \cancel{c_B}\, R_B}$	$\cancel{(c_B, d_B) = \mathsf{commit}(1\|m_B, R_B)}$
	$\xrightarrow{d_A}$	$\widehat{R}_A = \mathsf{open}(\cancel{0}\|\widehat{m}_A, \widehat{c}_A, \widehat{d}_A)$
$\cancel{\widehat{R}_B = \mathsf{open}(1\|\widehat{m}_B, \widehat{c}_B, \widehat{d}_B)}$	$\xleftarrow{\cancel{d_B}}$	
$\mathsf{SAS} = \widehat{R}_B \oplus h(\widehat{m}_B, R_A)$		$R_B \oplus h(m_B, \widehat{R}_A)$
$\cancel{\mathsf{SAS} = R_A \oplus \widehat{R}_B}$	$\xrightarrow{\mathsf{SAS}}$	check $\mathsf{SAS} \stackrel{?}{=} \cancel{\widehat{R}_A \oplus R_B}$
check SAS are the same	$\xleftarrow{\mathsf{SAS}}$	

- Two interleaved Vau-SAS-IMAP
- May a 3-move (optimal) protocol exist?

PV-SAS-MCA [PV06b]

		
pick x_A ~~Input: m_A~~		~~Input: m_B~~ pick x_B
$m_A = g^{x_A}$		$m_B = g^{x_B}$
pick $R_A \in \{0,1\}^\ell$		pick $R_B \in \{0,1\}^k$
$(c,d) = \mathsf{commit}(\cancel{m_A}, R_A)$	$\xrightarrow{m_A,\, c}$	
	$\xleftarrow{m_B,\, R_B}$	
	$\xrightarrow{d}$	$\widehat{R}_A = \mathsf{open}(\cancel{\widehat{m}_A}, \widehat{c}, \widehat{d})$
$\mathsf{SAS} = h(m_A \\| \widehat{m}_B, R_A, \widehat{R}_B)$		
$\mathsf{SAS} = \widehat{R}_B \oplus h(\widehat{m}_B, R_A)$	$\xrightarrow{\mathsf{SAS}}$	check $\mathsf{SAS} \stackrel{?}{=} R_B \oplus h(m_B, \widehat{R}_A)$
check SAS are the same	$\xleftarrow{\mathsf{SAS}}$	
$\mathsf{sk}_A = \widehat{m}_B^{x_A}$		$\mathsf{sk}_B = \widehat{m}_A^{x_B}$

- PV-SAS-AKA
- Comparison with MANA IV [LN06]

PV-SAS-MCA [PV06b]

Theorem

Assume a (T, ε_c)-equivocable commitment

a (T, ε_h)-almost strongly universal function

Then, PV-SAS-MCA is

$$\left(T,\ q_A + q_B,\ q_A(q_A - 1 + q_B)(2^{-k} + \varepsilon_c + \varepsilon_h)\right)$$

SAS-based Cryptography

Towards Group Settings

Group implications...

- More than two parties implies that
  - DH cannot be used.
  - Instead, we can use the Burmester-Desmedt (BD).
  - No group-MCA protocol exist?

- Security proofs become more complex:
  - More parties
  - Increase in communication
  - The same message may be received differently by each

LP-SAS-GMA [LP08]

- use commitment to temporarily hide secret keys
- direct authentication (as in MANA IV)

 $\mathcal{P}_{\mathsf{id}_i}$

Input: m_i

pick r_i

$(c_i, d_i) = \mathsf{commit}(\mathsf{crs}, i, r_i)$ $\xrightarrow{i, m_i, c_i}$

d_i

$\forall j : (j, \hat{r}_{ji}) = \mathsf{open}(\mathsf{crs}, \hat{c}_{ji}, \hat{d}_{ji})$

SAS

$\mathsf{SAS}_i = \mathsf{H}\left((\widehat{\mathcal{G}}_i, \widehat{\vec{m}}_i), \widehat{\vec{r}}_i\right)$

check $\mathsf{SAS}_i = \mathsf{SAS}_j$

LP-SAS-GMA [LP08]

Theorem

Assume a (T, ε_b)-binding and
(T, ε_{nm})-non-malleable commitment

a (T, ε_h)-almost strongly universal function

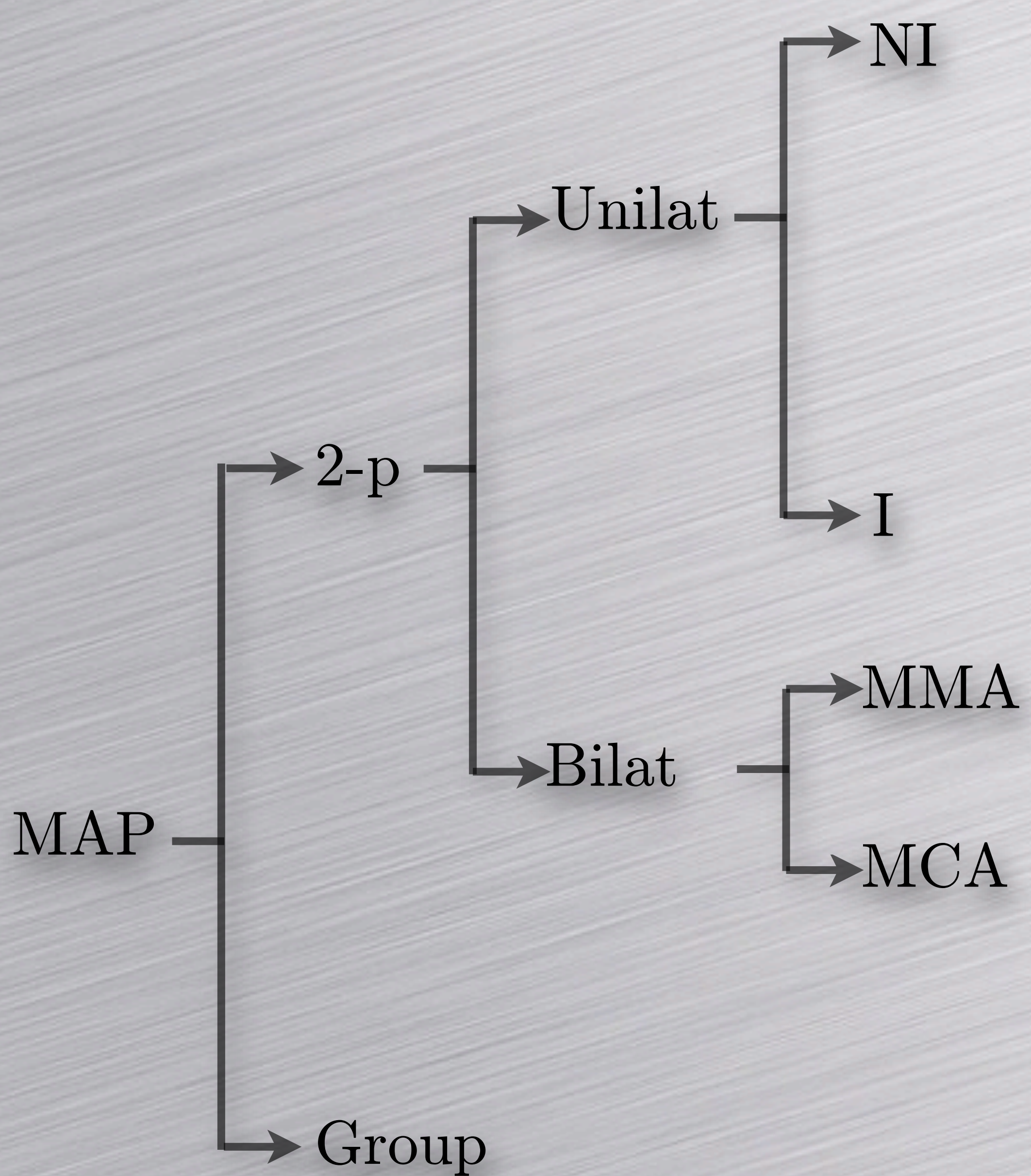
Then, LP-SAS-GMA is

$$\left(T,\ q,\ q(2^{-k} + n\varepsilon_{nm} + \varepsilon_b + \varepsilon_h)\right)$$

SAS-based Cryptography

Summary

Summary



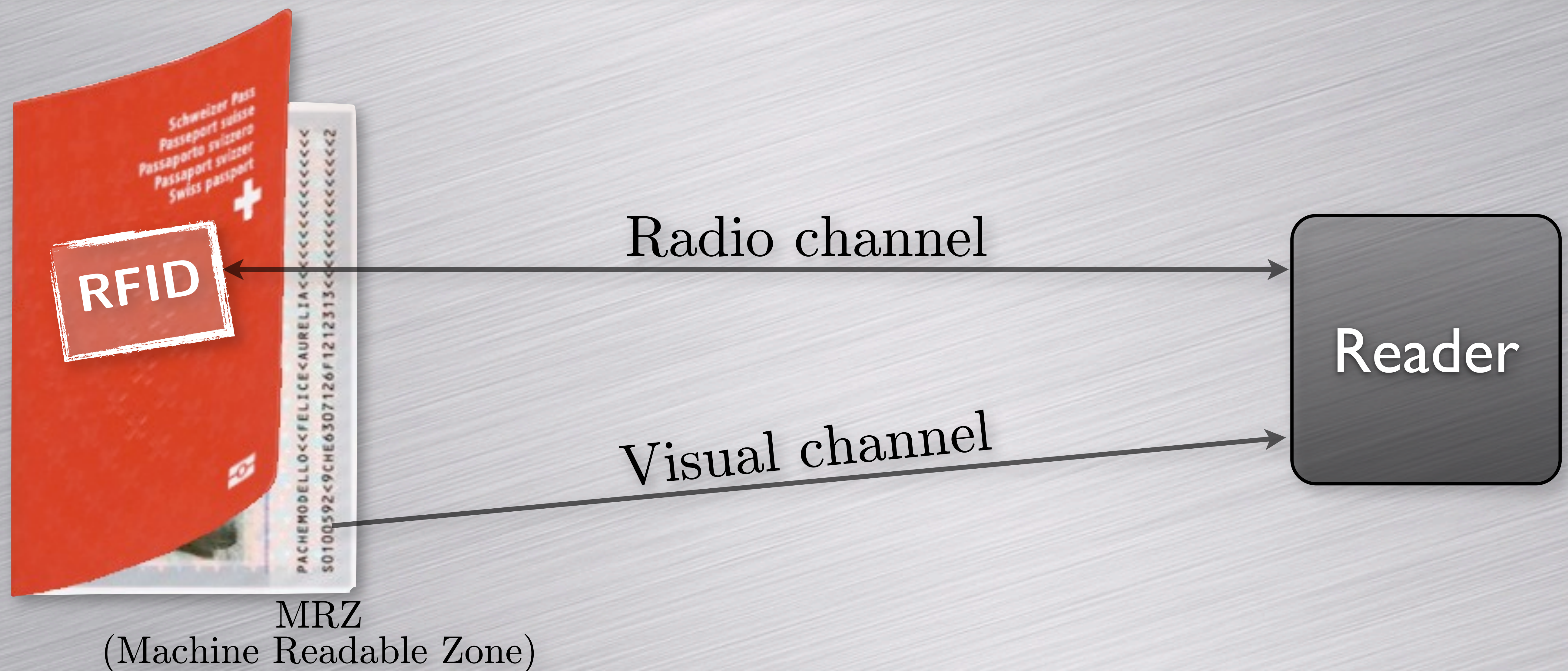
	Auth channel	Optimal	Sec proof
CRHF-based [BSSW02]	weak		y
MANA I [GMN04]	strong		y
PV-NIMAP[PV06a]	weak	y	y
eTCR-based [RWSN07]	weak	y	y
HCR-based [MS07]	weak	?	y
Vau-SAS-IMAP [Vau05]	weak	y	y
ICR-based [MS08]	weak	?	y
MANA III [GMN04]	strong		y
PV-SAS-MMA [PV06b]	weak	y	y
Vau-SAS-MCA [Vau05]	weak		
PV-SAS-MCA [PV06b]	weak	y	y
MANA IV [LN06]	weak	y	y
Group-MANA IV [VAN06]	weak		y
LP-SAS-GMA [LP08]	weak	y	y

LP-SAS-AKA [PV06b]

LP-SAS-GKA [LP08]

Efficient Deniable Authentication for Signatures

Reading an E-passport



Implementation, use, and security mandated by the International Civil Aviation Organization (ICAO).

Access Control

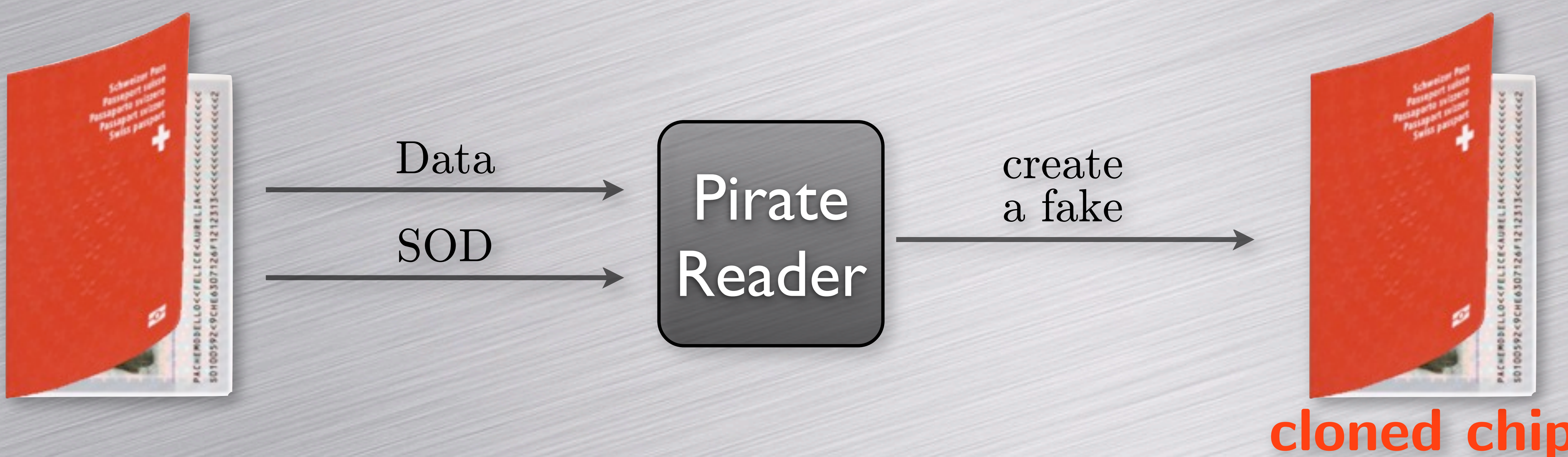
- By default, no access control
- Basic Access Control (BAC)
  - Prove to the e-passport that you have visual access
  - Use an encryption key $sk=f(MRZ)$
- Extended Access Control (EAC)
  - Terminal authentication, PKI for border patrols
  - EU standard (not an ICAO standard)
  - Basic data must remain accessible

Usually used

Passive Authentication

- Aims to prove that the **data is genuine**
- The chip has a Security Object Document (SOD)
- Basically, the national authority signed the data

$$\mathrm{SOD} = \mathsf{sign}_{K_{s,NA}}(\mathrm{data})$$



Active Authentication

- Aims to prove that the **chip is genuine**
  - no cloning and no substitution possible
- The e-passport contains a pair of keys: Kp and Ks
  - Ks stored in a secure memory
  - Kp is a standard data (authenticated by SOD)

Reader

$\xrightarrow{\text{data and SOD}}$ Verify Kp

$\xleftarrow{\text{challenge } c}$ Pick c

$\sigma = \mathsf{sign}_{Ks}(c)$ $\xrightarrow{\text{response } \sigma}$ Verify σ

The passport is able to sign...

Privacy Issue

- Anyone having a reader (50$) can obtain all data and the SOD
- Publishing the data only:
  - the owner can still claim that it is incorrect
- But, publishing the SOD too:
  - SOD is an evidence of the authenticity of DGs

Goal

Protect the SOD...

(Remember that data should be accessible.)

Solution: The Main Idea

Reader

X

$K_{p,NA}, K_{s,NA}$

$K_{p,NA}$

data

σ

data, SOD

data

$\cancel{SOD}$

data, SOD

data is valid

Privacy issue

interactive proof
"I know a valid σ"

ONTAP Overview

Signer **Prover** **Verifier**

$(K_p, K_s) = \mathsf{setup}()$ $\xrightarrow{K_p}$ (to Prover) $\xrightarrow{K_p}$ (to Verifier)

$(\sigma_p, \sigma_s) = \mathsf{sign}(K_s, m)$ $\xrightarrow{\sigma_p, \sigma_s}$ $\xrightarrow{\sigma_p}$

$\mathsf{iProof}(K_p, m, \sigma_p, \sigma_s) \longleftrightarrow \mathsf{iProof}(K_p, m, \sigma_p)$

$\hookrightarrow$ accept/reject

- Properties:
  - Completeness
  - Unforgeability (sign) + soundness (iProof)
  - Non-transferability (offline)

ONTAP Construction

Theorem

An ONTAP can be build with

- a secure signature scheme such as
  - the signature is splittable in two parts: σ_p and σ_s
  - σ_p is simulatable
- a zero-knowledge proof for witness σ_s

An e-passport uses RSA, DSA, or ECDSA.

The Guillou-Quisquater Protocol

RSA params: $N{=}pq$, $ed \equiv 1 \pmod{\varphi(N)}$

P		**V**
Input: w	*Input:* N,e,x	
pick y		
$Y = y^e \bmod N$	$\xrightarrow{\quad Y \quad}$	
	$\xleftarrow{\quad r \quad}$	pick r
$z = yw^r \bmod N$	$\xrightarrow{\quad z \quad}$	check $z^e \stackrel{?}{=} Yx^r \pmod{N}$

P proves to **V**:
I know w

GQ is a proof knowledge with:

- ☑ Efficiency
- ☑ Completeness
- ☑ Soundness

V is convinced because P replied to the challenge r.

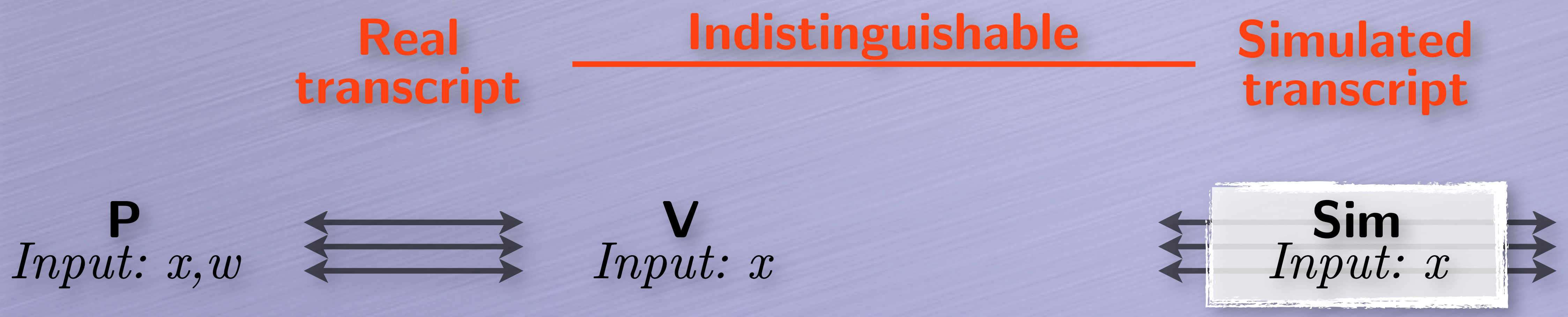
Zero-Knowledge

No information leaks to **V** (except the statement)

Zero-knowledge

For any x, there exists **Sim** able to generate the transcript without w.

$$\left\{\mathsf{View}_{\mathsf{V}}\left(\mathsf{proof}_{\mathsf{P}(w),\mathsf{V}(z)}(x)\right)\right\}_{z\in\{0,1\}^*,x\in L_R,w\in R(x)} \qquad \{\mathsf{Sim}(x,z)\}_{z\in\{0,1\}^*,x\in L_R}$$



Zero-Knowledge: GQ protocol

P		**V**
Input: w		*Input:* N, e, x
pick y		
$Y = y^e \bmod N$	$\xrightarrow{Y}$	
	$\xleftarrow{r}$	pick r
$z = yw^r \bmod N$	$\xrightarrow{z}$	check $z^e \stackrel{?}{=} Yx^r \pmod N$

Simulated transcript (without w):

given $N, e, x,$ and r
pick z
$Y = z^e / x^r$
output (Y, r, z)

Everybody is able to generate this transcript, this is not a proof of interaction.

Fiat-Shamir Transform

The GQ protocol is only Honest-Verifier ZK.

P		**V***
Input: w	*Input:* N, e, x	
pick y $Y = y^e \bmod N$	$\xrightarrow{\quad Y \quad}$	
	$\xleftarrow{\quad r \quad}$	~~pick~~ r $\quad r = f(Y)$
$z = yw^r \bmod N$	$\xrightarrow{\quad z \quad}$	check $z^e \stackrel{?}{=} Yx^r \pmod N$

Considering malicious verifiers:

- the proof is not simulatable (without w),
- the proof becomes transferable.

Protocol Transform

To obtain a **full** zero-knowledge protocol,
ensure that r is chosen independently from Y.

P		**V**
Input: w	*Input:* N, e, x	
		pick r
	$\xleftarrow{c}$	$(c,d) = \mathsf{commit}(r)$
pick y		
$Y = y^e \bmod N$	$\xrightarrow{Y}$	
$r = \mathsf{open}(c,d)$	$\xleftarrow{\cancel{r}\ d}$	~~pick r~~
$z = yw^r \bmod N$	$\xrightarrow{z}$	check $z^e \stackrel{?}{=} Yx^r \pmod N$

Require the CRS Model...

P		**V**
Input: w	*Input:* N, e, x	
(Kp,Ks) = keygen	$\xrightarrow{Kp}$	pick r
pick y	$\xleftarrow{c}$	$(c,d) = \text{commit}(r)$ **Kp**,
$Y = y^e \bmod N$	$\xrightarrow{Y}$	
$r = \text{commit}(c,d)$ **Kp**	$\xleftarrow{d}$	
$z = yw^r \bmod N$	$\xrightarrow{z\ ,Ks}$	check $z^e \stackrel{?}{=} Yx^r \pmod N$

To prove the soundness, we should add a trapdoor.

- in the plain model, we add a move.
- in the CRS/RO model, Kp is a global setup.

Deniable Zero-Knowledge

Honest-Verifier ZK:

P *Input:* x, w $\Longleftrightarrow$ **V** *Input:* x — **Sim** *Input:* x

ZK:

P *Input:* x, w $\Longleftrightarrow$ **V*** *Input:* x — **Sim** *Input:* x

ZK in the CRS model:

P *Input:* x, w, crs $\Longleftrightarrow$ **V*** *Input:* x, crs — **Sim** *Input:* x

Deniable ZK in the CRS model:

P *Input:* x, w, crs $\Longleftrightarrow$ **V*** *Input:* x, crs — **Sim** *Input:* x, crs

Back to ONTAP: RSA example

Signer | **Prover** | **Verifier**

RSA : p, q, N, e, d
$K_p = (N, e)$
$K_s = d$

Signer $\xrightarrow{K_p}$ Prover $\xrightarrow{K_p}$ Verifier

$\sigma_p = \mathsf{H}_{\mathsf{seed}}(m)$
$\sigma_s = \sigma_p^d \bmod N$

Signer $\xrightarrow{m, \sigma_p, \sigma_s}$ Prover $\xrightarrow{m}$ Verifier

Verifier: pick r

Verifier: $(c, d) = \mathsf{commit}(\mathsf{crs}, r)$ $\xrightarrow{c}$ Prover: pick y

Prover: $Y = y^e \bmod N$ $\xrightarrow{Y, \sigma_p}$ Verifier: check $\mathsf{V}(\sigma_p, m) \stackrel{?}{=} 1$

Verifier $\xrightarrow{d}$ Prover: $r = \mathsf{open}(\mathsf{crs}, c, d)$

Prover: $z = y\sigma_s^r \bmod N$ $\xrightarrow{z}$ Verifier: check $z^e \stackrel{?}{=} Y\sigma_p^r \bmod N$

Read

Conclusion

Conclusion

- SAS-based cryptography:
  - dedicated network and adversarial model
  - generic security analysis (notion of optimality)
  - optimal NIMAP, MMA, MCA, and GMA
  - optimal AKA and GKA
- Offline Non-Transferable Authentication Protocol
  - solve privacy issue in a three-party setting (e-passport)
- (Hash-and-sign-based signatures)
  - pre-processing strengthening actual implementations

Thank you
for
your attention!