Malleable Cryptography for Security and Privacy in the Cloud

Daniel Slamanig, Graz University of Technology

ARES-SECODIC 2016
Salzburg, 31 August 2016

Talk Outline

- Brief overview of the PRISMALCOUD project
- Challenges in dynamic cloud environments
- Malleable cryptographic primitives
  - Encryption and signatures
- Applications within PRISMACLOUD
  - Homomorphic proxy re-cryptography
  - Multikey-homomorphic cryptography

PRISMACLOUD Overview

- Improve the security and privacy of cloud based services and make them usable for providers and users
- Enable **end-to-end security** for cloud users and protect their privacy by means cryptography
- Bring novel cryptographic concepts and methods to practical application

- The project is a huge undertaking and produces outcome in many different disciplines and layers
- Today, the focus is on a particular aspect of the project

Project Metadata

Call: H2020-ICT-2014-1

Acronym: PRISMACLOUD

Type of Action: RIA

Number: 644962

Partners: 16

Duration: 42 months

Start Date: 2015-02-01

Estimated Project Cost: approx. 8.5M Euro

Requested EU Contribution: approx. 8M Euro

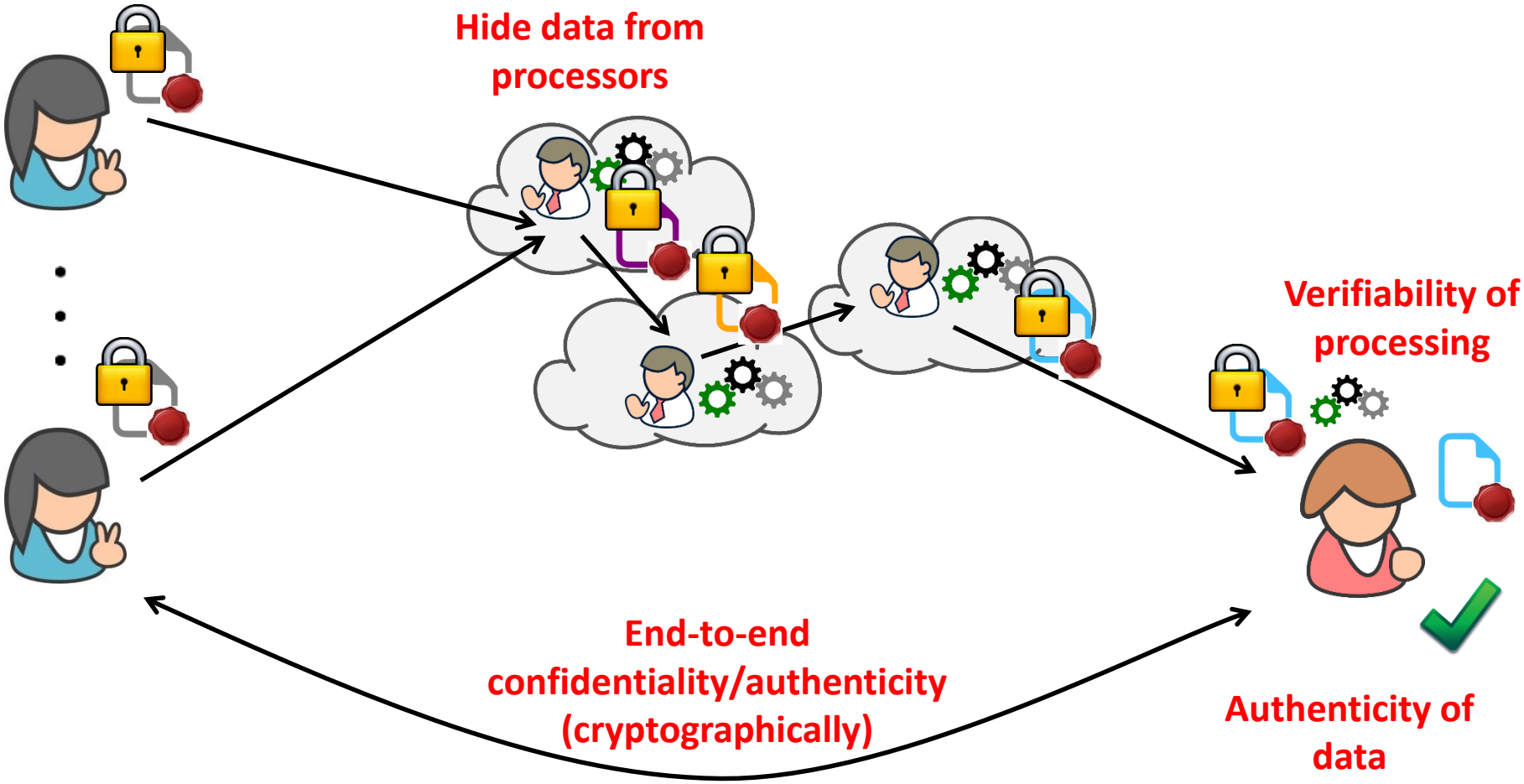
Coordinator: Austrian Institute of Technology GmbH

PRISMACLOUD Partners

The PRISMACLOUD consortium is composed of a balanced team of

6 industrial partners,
2 industrial research centres,
2 institutional research organisations and
6 universities.

Dynamic Cloud Environment: Composition of Services



How to Realize Such Features?

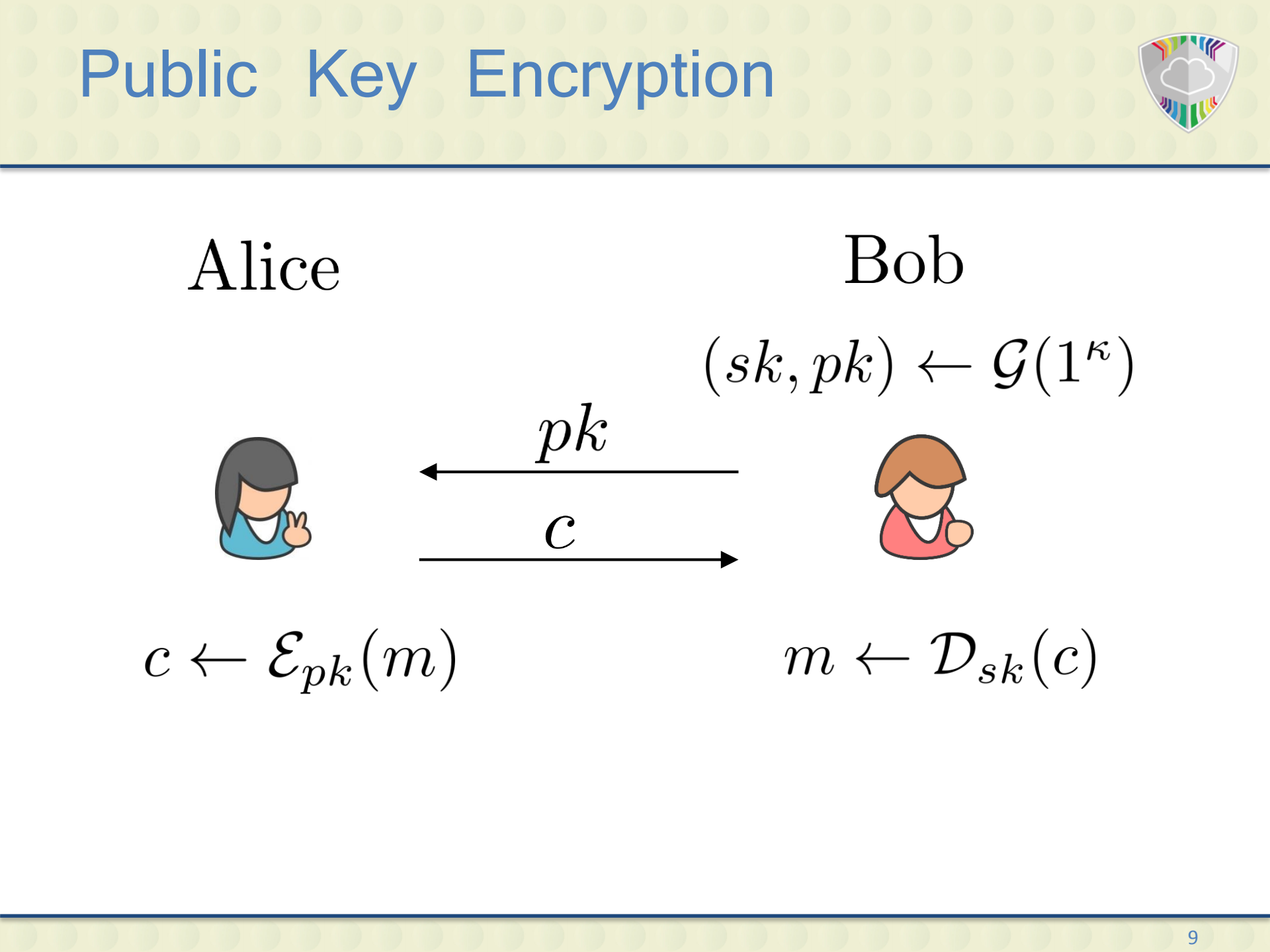
- What is the issue when end-to-end guarantees are required?
  - Never decrypt in transit
  - Never lose the data origin protection (authenticity) in transit

- Need adequate cryptographic primitives that support
  - agility
  - functionality

Malleable Cryptography

- Let us take a cryptographic primitive, e.g., an encryption or signature scheme
- Informally, a primitive is malleable if we can manipulate its cryptographic objects (ciphertexts, signatures) in a meaningful (controlled) way without knowing the secret
  - Changing the encrypted/signed messages in a controlled way **without** the key
  - Changing the key corresponding to a ciphertext/signature
  - Combining ciphertexts/signatures under **different** keys
- Subsequently, many details are omitted for the sake of presentation

Public Key Encryption



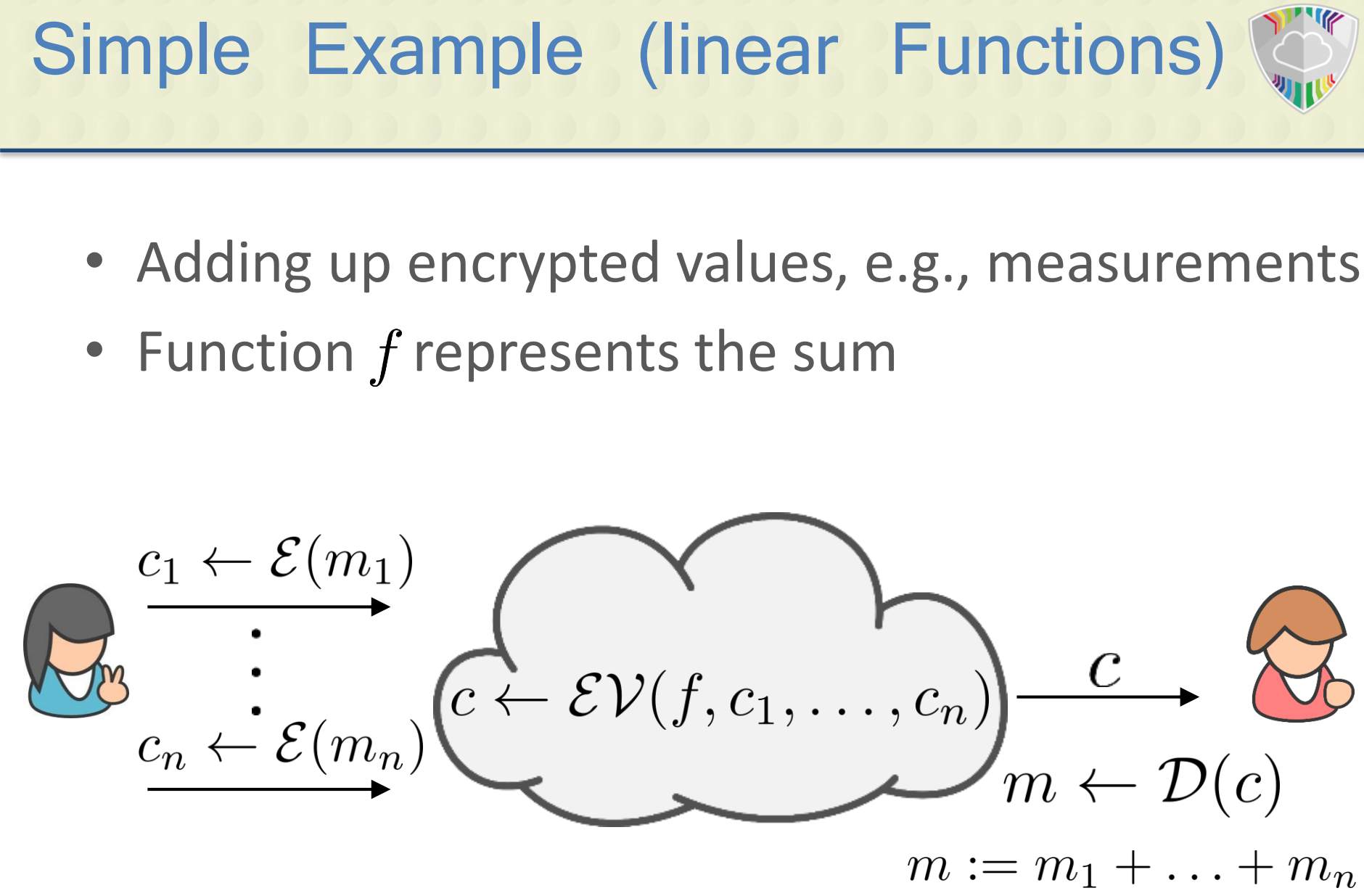
Homomorphic Encryption

- Operations performed on ciphertexts “carry over” to the underlying plaintexts
- Allows to evaluate a class of functions on ciphertexts
- Scheme $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ comes with an additional evaluation algorithm $\mathcal{EV}$ such that for all functions $f \in \mathcal{F}$

$$f(m_1, \ldots, m_n) = \mathcal{D}(\mathcal{EV}(f, \mathcal{E}(m_1), \ldots, \mathcal{E}(m_n)))$$

- If $\mathcal{F}$ covers all computable functions: FHE

Simple Example (linear Functions)



- Adding up encrypted values, e.g., measurements
- Function f represents the sum

$c_1 \leftarrow \mathcal{E}(m_1)$

$\vdots$

$c_n \leftarrow \mathcal{E}(m_n)$

$c \leftarrow \mathcal{EV}(f, c_1, \ldots, c_n)$

c

$m \leftarrow \mathcal{D}(c)$

$m := m_1 + \ldots + m_n$

Multikey-Homomorphic Encryption

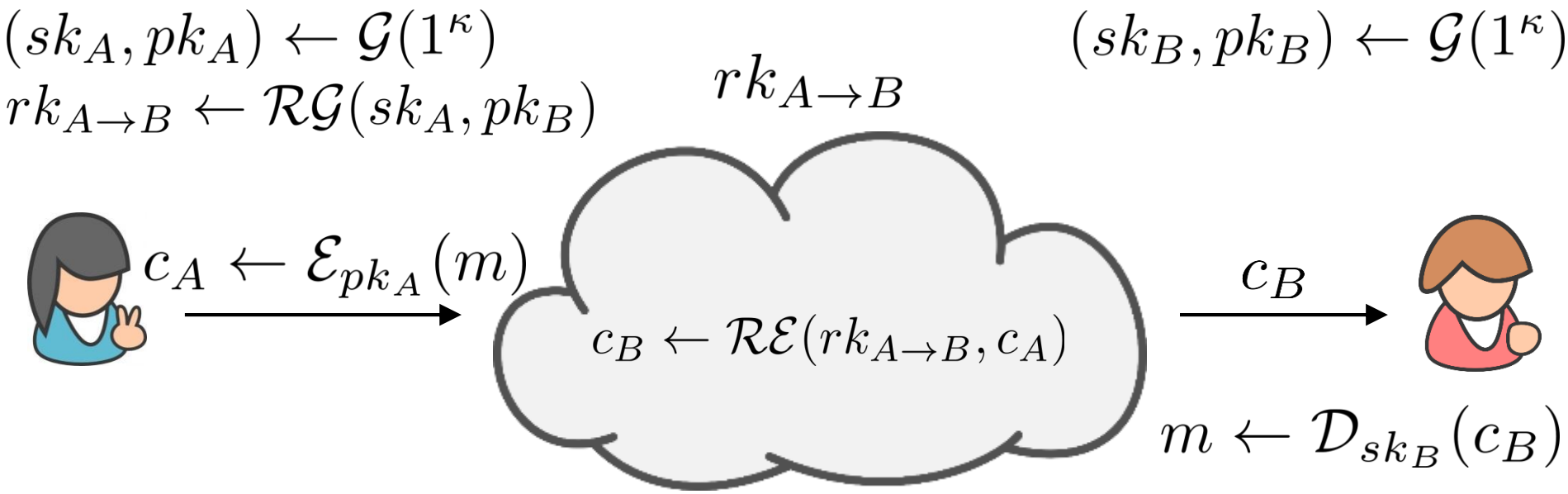
- Homomorphic encryption scheme that can evaluate functions on ciphertexts under **different** public keys
- Decryption requires **all** respective private keys

$$f(m_1, \ldots, m_n) = \mathcal{D}_{sk_1,\ldots,sk_n}(\mathcal{EV}(f, \mathcal{E}_{pk_1}(m_1), \ldots, \mathcal{E}_{pk_n}(m_n)))$$

- We investigate a related concept based on homomorphic proxy re-encryption

Proxy Re-Encryption

- An encryption scheme $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ with additional algorithms
  - re-key generation $rk_{A\to B} \leftarrow \mathcal{RG}(sk_A, pk_B)$
  - re-encryption $c_B \leftarrow \mathcal{RE}(rk_{A\to B}, c_A)$



Homomorphic Proxy Re-Encryption

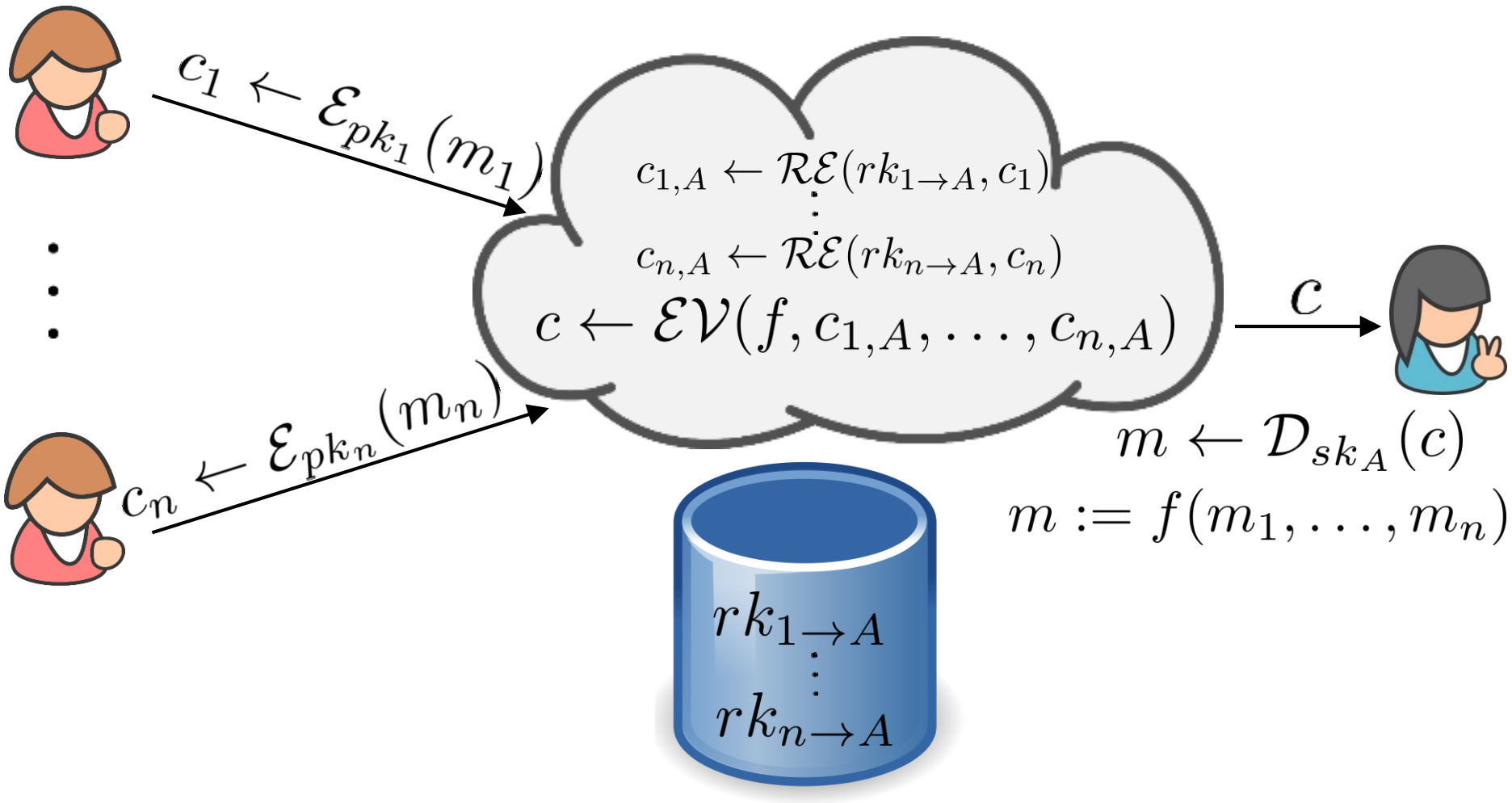
- A PRE scheme $(\mathcal{G}, \mathcal{E}, \mathcal{D}, \mathcal{RG}, \mathcal{RE})$ that is homomorphic with respect to a class of functions
- Scheme comes with evaluation algorithm $\mathcal{EV}$

$$f(m_1, \ldots, m_n) = \mathcal{D}(sk_A, \mathcal{EV}(f, \mathcal{E}_{pk_A}(m_1), \ldots, \mathcal{E}_{pk_A}(m_n)))$$

- Under the correctness of the PRE, if re-keys $rk_{i \to A}$ for ciphertexts $c_i \leftarrow \mathcal{E}_{pk_i}(m_i)$ are available

$$f(m_1, \ldots, m_n) = \mathcal{D}(sk_A, \mathcal{EV}(f, \mathcal{RE}(rk_{1 \to A}, c_1)), \ldots, \mathcal{RE}(rk_{n \to A}, c_n)))$$

Homomorphic PRE Example

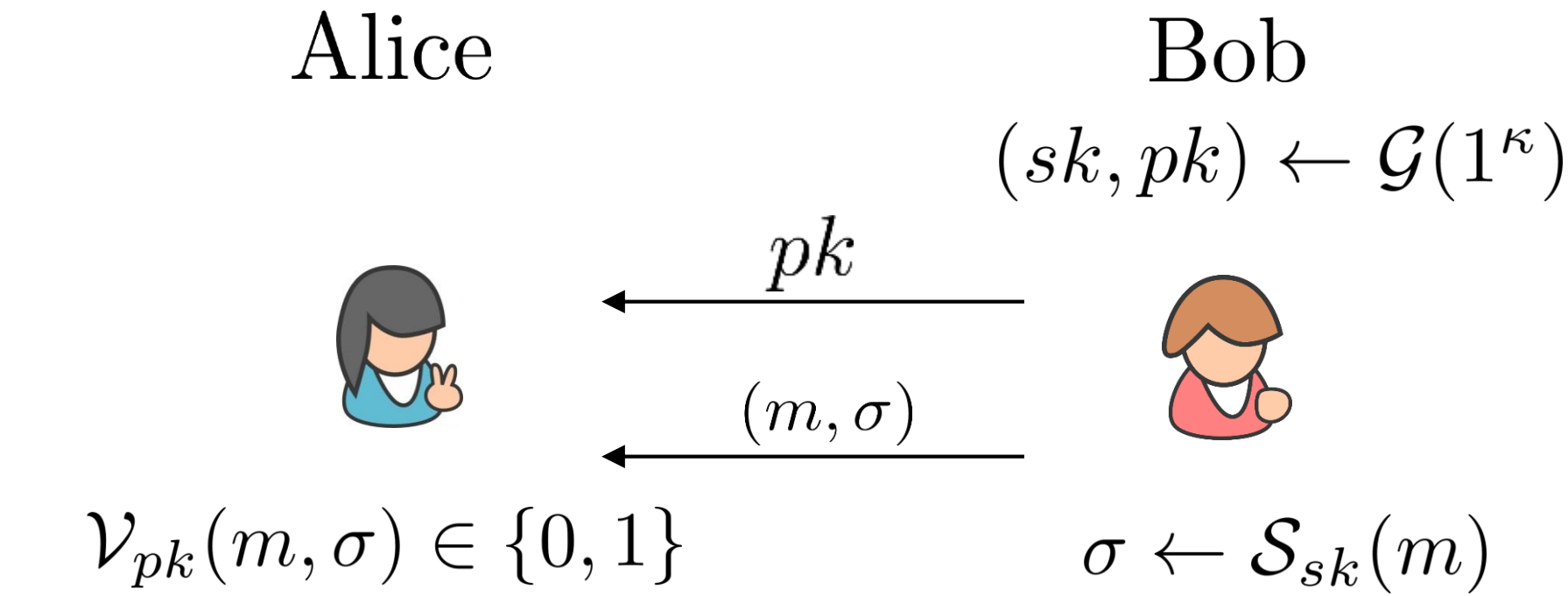


What do we achieve so far?

- Only confidentiality guarantees, but no means to preserve verifiability of operations/computations on data in transit
- Neither do we get authenticity guarantees

- Now we consider the same “stack” for signatures (authenticators)
- And then briefly discuss how to combine them and what this will give us

Digital Signatures



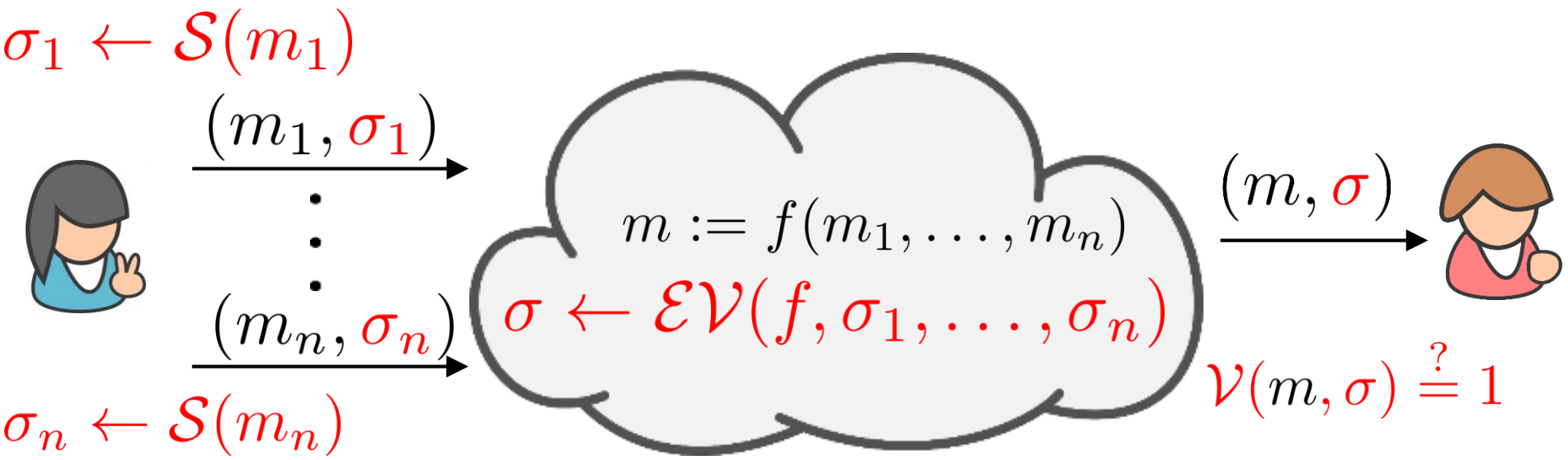
- In a message authentication code (MAC) there is a secret key shared by Alice and Bob and verification also requires the secret key

Homomorphic Signatures

- Operations performed on signatures “carry over” to the signed messages
- Allows to evaluate a class of functions on signatures
- Scheme $(\mathcal{G}, \mathcal{S}, \mathcal{V})$ comes with an additional evaluation algorithm $\mathcal{EV}$ such that for all functions $f \in \mathcal{F}$

$$\mathcal{V}(f(m_1, \ldots, m_n), \mathcal{EV}(f, \mathcal{S}(m_1), \ldots, \mathcal{S}(m_n))) = 1$$

Homomorphic Signatures Example



Multikey-Homomorphic Signatures

- Homomorphic signature scheme that can evaluate functions on signatures under **different** public keys
- Verification requires **all** respective public keys (or a compact representation of a key set) denoted $\mathbf{pk}$

$$\mathcal{V}_{\mathbf{pk}}(f(m_1,\ldots,m_n),\mathcal{EV}(f,\mathcal{S}_{pk_1}(m_1),\ldots,\mathcal{S}_{pk_n}(m_n))) = 1$$

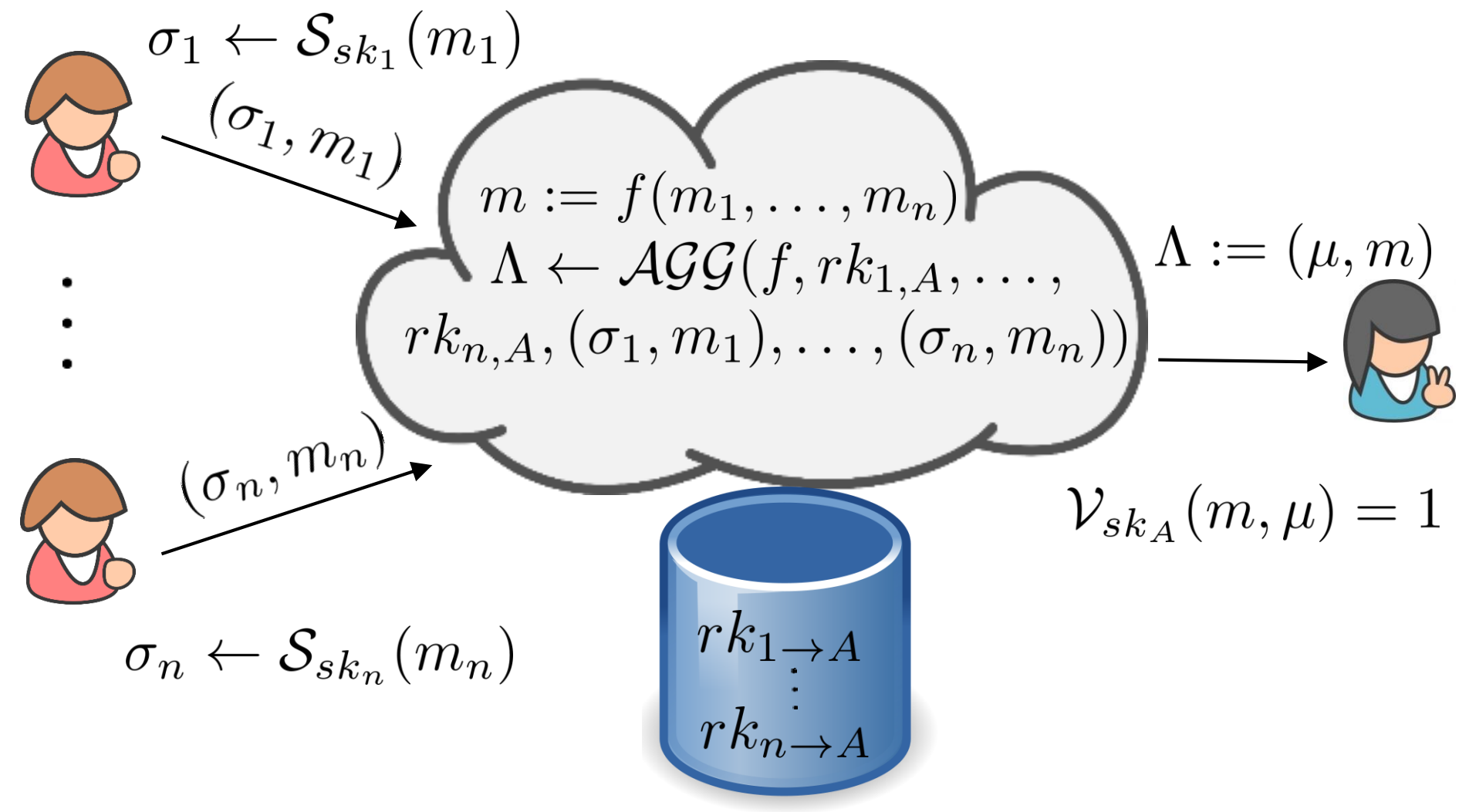
- We investigate a related concept based on homomorphic proxy re-authenticators

Proxy Re-Authenticators

- A combination of a signature scheme and a message authentication code (MAC)
- "Signature to MAC keys" given to a proxy
- Proxy can perform re-operations from signatures to MACs
- The final result is privately verifiable

- There are also proxy re-signatures
  - No approach/construction is compatible with homomorphic properties

Homomorphic Proxy Re-Authenticators



Combining Both Tools

- Homomorphic PRAs do neither hide inputs nor outputs from proxy
- So far no multikey-homomorphic concept that combines encryption and authentication
- We can combine homomorphic PRAs with homomorphic PREs
  - Obtain a general framework of homomorphic proxy re-cryptography
- Prior work
  - Verifiable homomorphic encryption [LDPW14] (only under a single key)
  - Private and unforgeable data aggregation [LEÖM15] (only sum of inputs; common setup – shared key)

Input and Output Privacy of Homomorphic PRAs

- We must not send (σ_1, m_1) directly
  - Using homomorphic PRE for messages not sufficient
  - Signatures may still leak messages
- Need some technical tricks
  - Use homomorphic PRE to encrypt extended message (additional random message)
  - Use PRA to sign extended message and do not leak randomness
  - Constructions not fully black-box

Conclusions

- Homomorphic proxy re-cryptography
  - Homomorphic proxy re-encryption
  - Homomorphic proxy re-authenticators
- Allows us to support confidentiality and verifiability in a multi-key (-source) setting
- In contrast to multikey-homomorphic primitives, we have a single independent key for decryption/verification
- Many interesting open questions

References

[DMNP16] D. Fiore and A. Mitrokotsa and L. Nizzardo and E. Pagnin: **Multi-Key Homomorphic Authenticators, AsiaCrypt 2016 (ePrint 2016/804)**

[DRS16] D. Derler, S. Ramacher, D. Slamanig: **Homomorphic Proxy Re-Cryptography for Secure And Verifiable Multi-Source Data Aggregation, Manuscript, 2016.**

[DS16] D. Derler, D. Slamanig: **Key-Homomorphic Signatures and Applications to Multiparty Signatures, Manuscript (ePrint 2016/792), 2016**

[LATV12] A. López-Alt, E. Tromer, and V. Vaikuntanathan. **On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption, STOC, 2012**

[LDPW14] J. Lai, R. H. Deng, and H. Pang, J. Weng: **Verifiable Computation on Outsourced Encrypted Data, ESORICS 2014**

[LEÖM15] I. Leontiadis, K. Elkhiyaoui, M. Önen, R. Molva: **PUDA - Privacy and Unforgeability for Data Aggregation, CANS 2015**

Contact

Website:

https://www.prismacloud.eu

Coordinator Contact:

Thomas Lorünser

thomas.loruenser@ait.ac.at

PRISMACLOUD is also on:

LinkedIn: https://in.linkedin.com/in/prismacloud

Twitter: https://twitter.com/prismacloud (@prismacloud)