RSAConference2019

San Francisco | March 4–8 | Moscone Center

BETTER.

SESSION ID: CRYPTO-09

ROBUST ENCRYPTION, EXTENDED

Remi Geraud, David Naccache, and Razvan Rosie

ENS Paris, CNRS, INRIA, PSL Research University
University of Luxembourg

Corresponding author: razvan.rosie@ens.fr

#RSAC

Robustness in a nutshell

Robustness: ciphertext can't be decrypted under two different keys.

FSE17: robustness for symmetric primitives.

PKC13: robustness for PKE revisited by Farshim et al.

TCC10: robustness introduced for PKE & IBE by Bellare et al.

Robustness – This Talk

CT-RSA19: robustness for Digital Signatures and Functional Encryption.

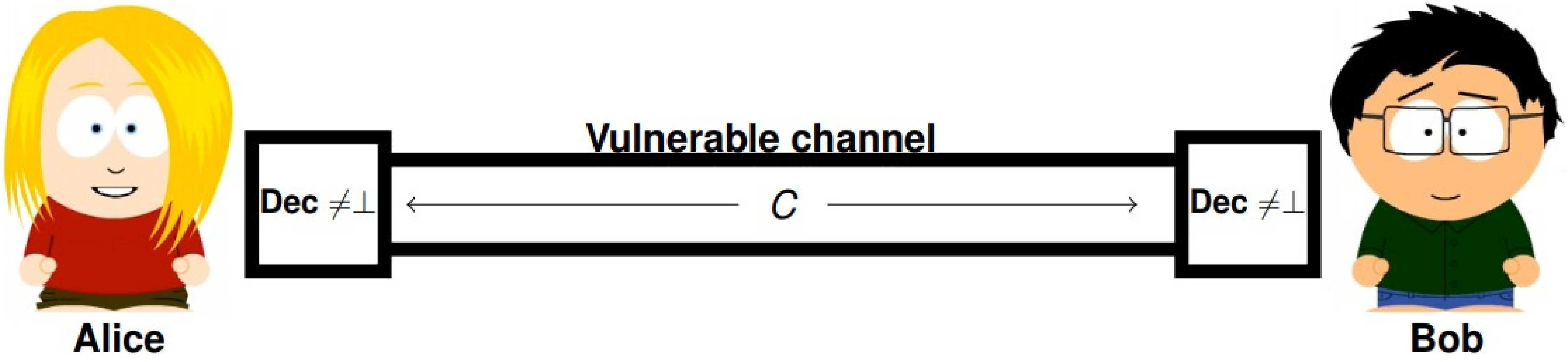
FSE17: robustness for symmetric primitives.

PKC13: robustness for PKE revisited by Farshim et al.

TCC10: robustness introduced for PKE & IBE by Bellare et al.

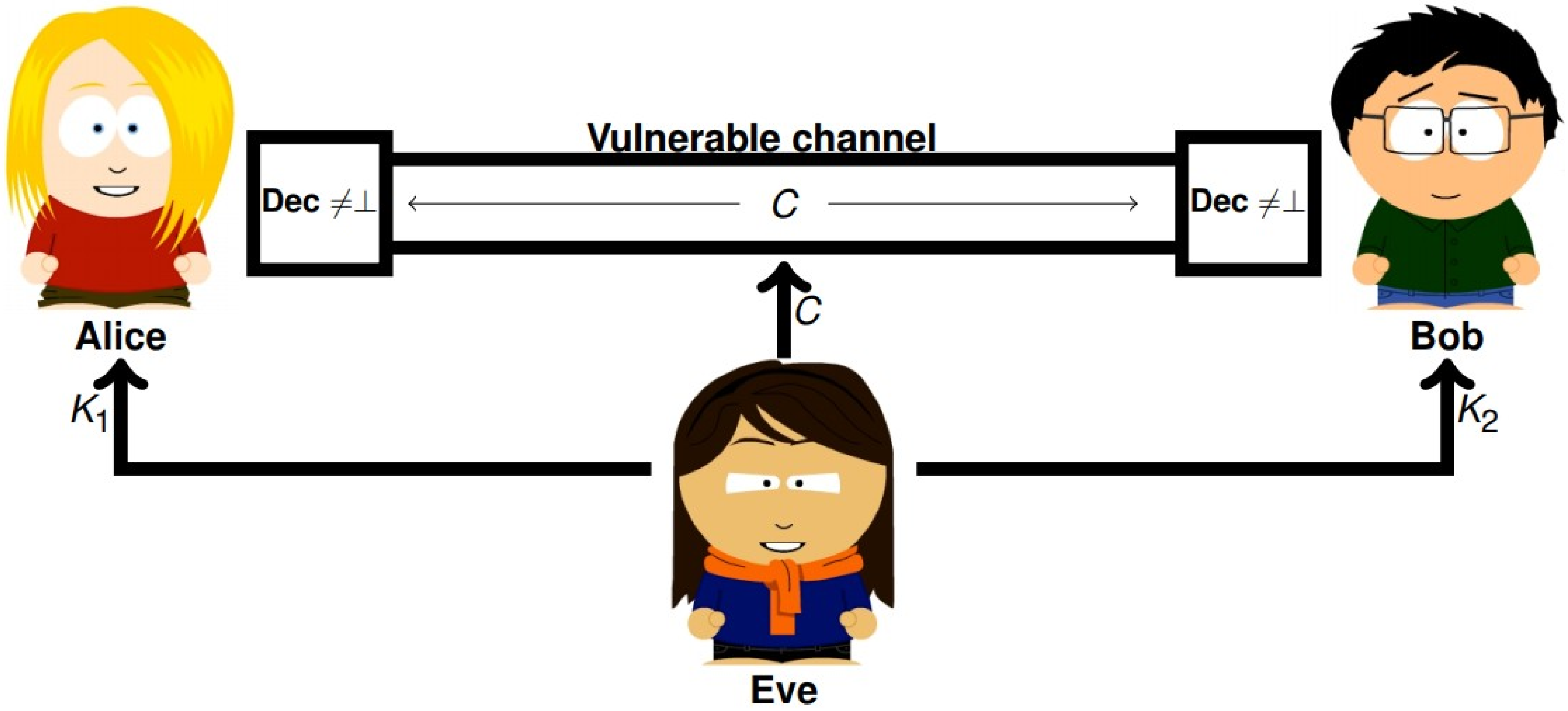
Key-Robustness in a Nutshell

Robustness: ciphertext can't be decrypted under two different keys.



Key-Robustness in a Nutshell

Robustness: ciphertext can't be decrypted under two different keys.



Motivating Robustness

Digital Signatures from Symmetric Encryption:

- $sk \leftarrow (K, s)$
- $pk \leftarrow \mathrm{Enc}(K, s)$ — contains the Symm. Enc. of s.
- $\sigma \leftarrow (\mathrm{PRF}(s, M), \pi)$ — PRF evaluation + ZK proof for correctness.

Is the scheme unforgeable?

Motivating Robustness

Digital Signatures from Symmetric Encryption:

- $sk \leftarrow (K, s)$
- $pk \leftarrow \mathrm{Enc}(K, s)$ — contains the Symm. Enc. of s.
- σ $\leftarrow (\mathrm{PRF}(s, M), \pi)$ — PRF evaluation + ZK proof for correctness.

Is the scheme unforgeable?

$\mathrm{Enc}(K, s) = \mathrm{Enc}(K', s') \Rightarrow$ FORGE

Robustness - Covered Primitives

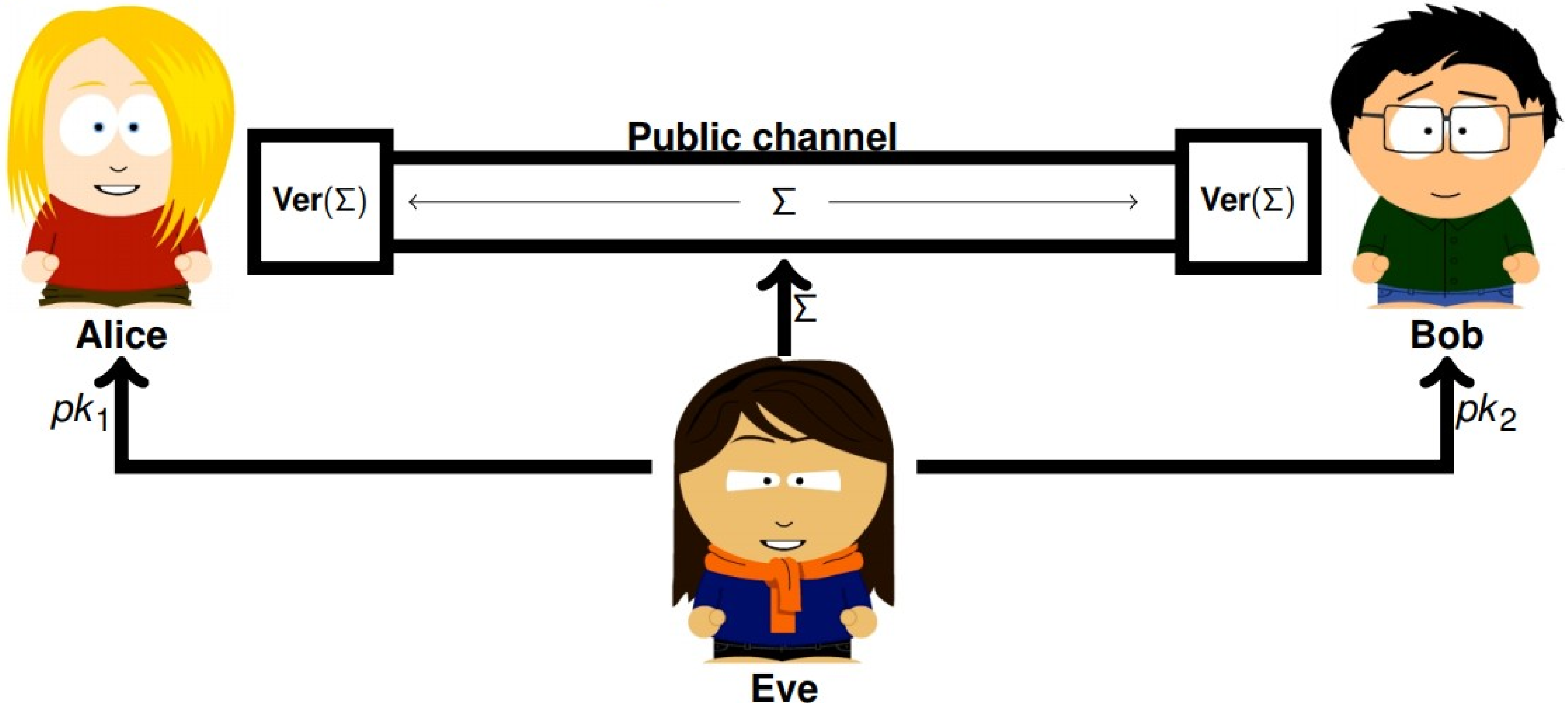
- NEW: Robustness to Signatures & Functional Encryption.
- TCC10, PKC13: Robustness introduced for PKE & IBE.
- AC10: Mohassel extends robustness to Hybrid Encryption.
- FSE17: Robustness for Symmetric primitives.

RSAConference2019

Warm Up: Robustness for Digital Signatures

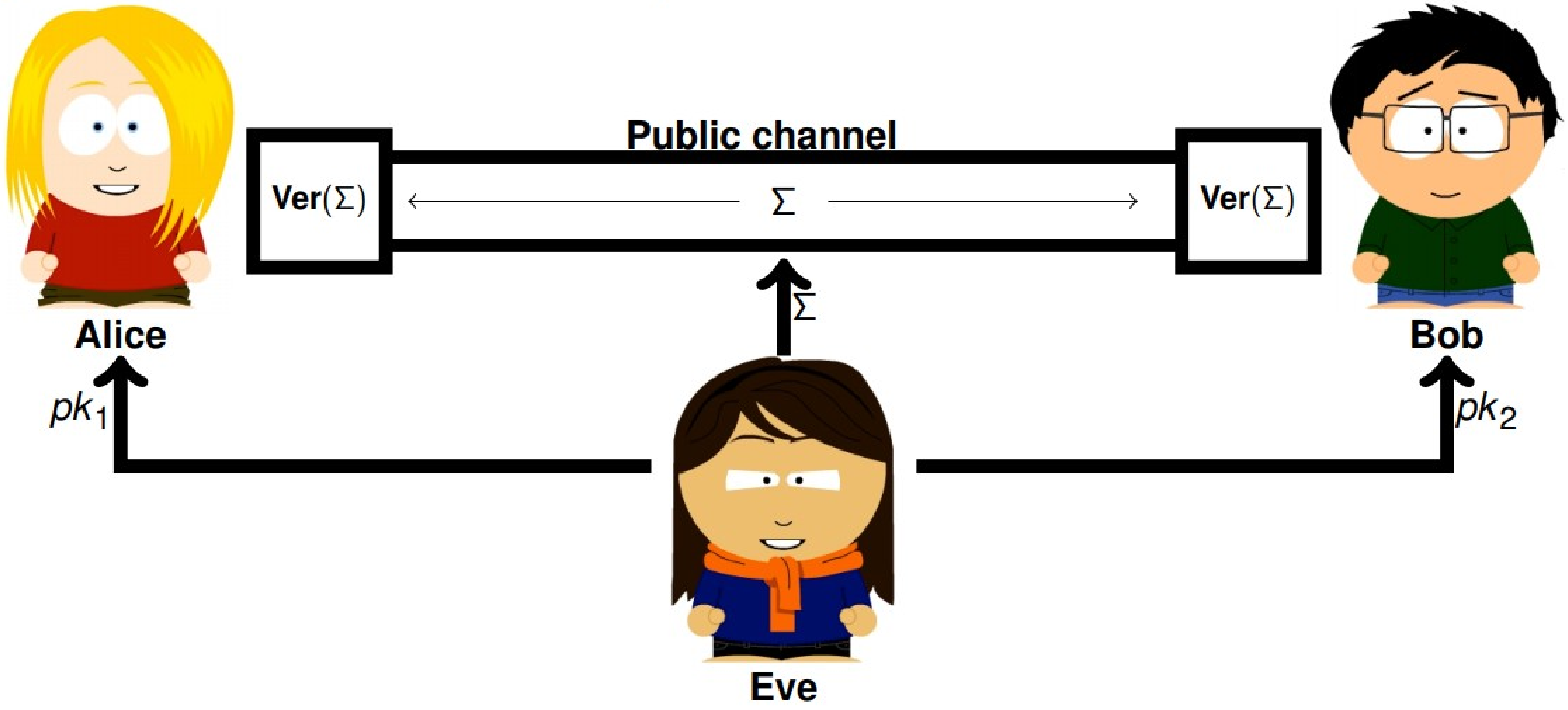
Robust Digital Signatures

Digital Signature - What we expect



Robust Digital Signatures

Digital Signature - What we expect



No signature Σ shall verify under multiple keys.

Robust Digital Signatures - Motivating Example

Consider the Boneh-Boyen signature scheme:

$pk \leftarrow (g_1, g_2, g_2^x, g_2^y, e(g_1, g_2))$.

$sk \leftarrow (x, y)$

To sign M, compute:

$$\sigma \leftarrow (g_1^{1/(x+M+y\cdot r)}, r)$$

To verify:

$$e(\sigma, g_2^x \cdot (g_2^y)^r \cdot g_2^M) = e(g_1, g_2)$$

Robust Digital Signatures - Motivating Example

Adversary *A* can always construct

$$(pk', sk')$$

Having:

$$g'_1 \equiv g_1^t \pmod p$$

Robust Digital Signatures - Motivating Example

Adversary *A* can always construct

$$(pk', sk')$$

Having:

$$g'_1 \equiv g_1^t \pmod p$$

Then, *A* can set *t*, *x'*, *y'* such that:

$$1/(x + M + y \cdot r) = t/(x' + M + y' \cdot r)$$

Robust Digital Signatures - Motivating Example

Adversary *A* can always construct

$$(pk', sk')$$

Having:

$$g'_1 \equiv g_1^t \pmod p$$

Then, *A* can set *t*, *x'*, *y'* such that:

$$1/(x + M + y \cdot r) = t/(x' + M + y' \cdot r)$$

There is always an *M* producing the same σ.

$$\mathbf{Sign}(sk, M) = g_1{}^{1/(x+M+y\cdot r)} = \sigma = g_1'{}^{1/(x'+M+y'\cdot r)} = \mathbf{Sign}(sk', M)$$

Robust Digital Signatures - The Security Model

- Strong Robustness (SROB): honestly generated pk_1 ,pk_2.
- Goal: find (M, σ) verifiable under pk_1 ,pk_2.

1. $(M, \sigma) \leftarrow \mathcal{A}^{\mathbf{Sign}_{sk_1, sk_2}}(pk_1, pk_2)$
2. $\mathbf{Ver}(pk_1, \sigma, M) = 1$
3. $\mathbf{Ver}(pk_2, \sigma, M) = 1$

Robust Digital Signatures - The Security Model

- Complete Robustness (CROB): adversarially generated pk_1 ,pk_2.
- Goal: find (M, σ) verifiable under pk_1 ,pk_2.

SROB

1. $(M, \sigma) \leftarrow \mathcal{A}^{\mathbf{Sign}_{sk_1,sk_2}}(pk_1, pk_2)$

CROB

1. $(pk_1, pk_2, \sigma, M) \leftarrow \mathcal{A}$

$\downarrow$ $\downarrow$

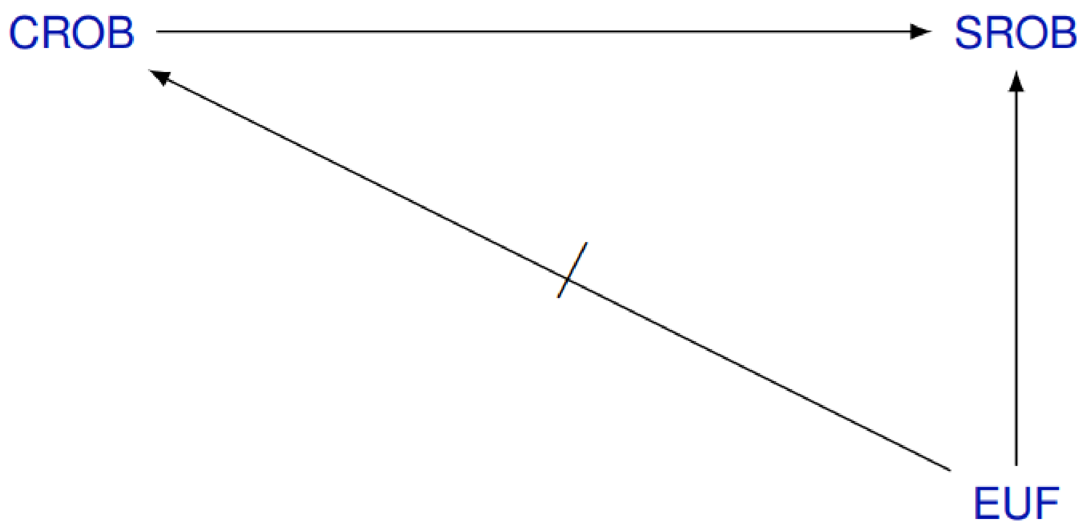
2. $\mathbf{Ver}(pk_1, \sigma, M) = 1$
3. $\mathbf{Ver}(pk_2, \sigma, M) = 1$

Robust Digital Signatures - Definitional Landscape

$\text{CROB} \longrightarrow \text{SROB}$

Robust Digital Signatures - Definitional Landscape

EUF-secure scheme ⇒ SROB-secure.



Robust Digital Signatures - SROB from EUF

Any EUF secure signature scheme achieves SROB security

Proof intuition:

Reduction $\mathcal{R}_{\mathcal{A}}(\lambda, pk_1, \textsc{Sign}_{sk_1}(\cdot))$:

1. $(pk_2, sk_2) \twoheadleftarrow \mathbf{Gen}(1^\lambda)$
2. construct $\textsc{Sign}_{sk_2}(\cdot)$
3. $(M, \sigma) \twoheadleftarrow \mathcal{A}^{pk_1, pk_2, \textsc{Sign}_{sk_1}(\cdot), \textsc{Sign}_{sk_2}(\cdot)}(1^\lambda)$
4. if $M \in \mathbf{Sign}_{sk_1}(\cdot)$.SignedMessages()
5. abort
6. return (M, σ)

$\square$

Robust Digital Signatures - CROB Transform

A CROB digital signature scheme can be achieved *generically*:

- Let H denote a collision resistant hash function (i.e. constructed from claw-free permutations).
- Idea: “commit” to the public-key by hashing it.
- Attach the hash to the signature.

Robust Digital Signatures - CROB Transform

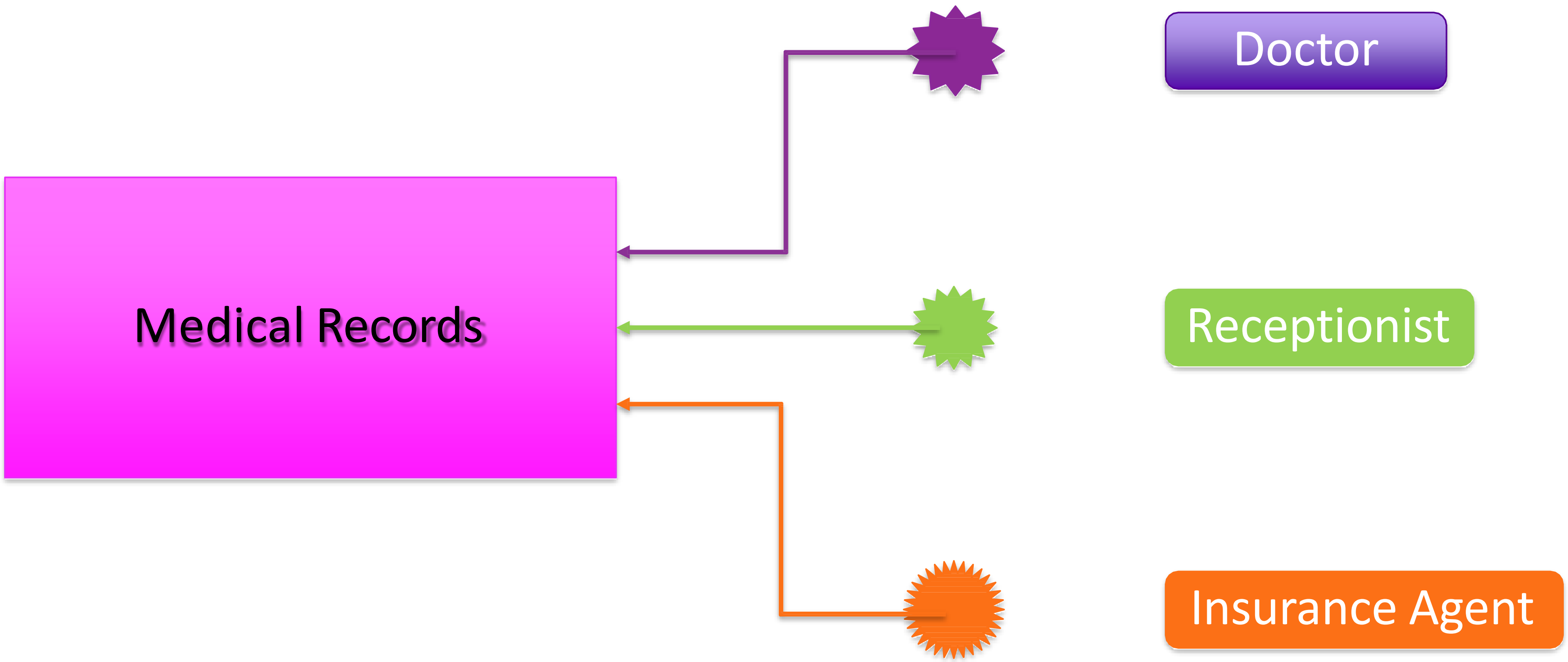
Why is it CROB-secure?

- If A comes up with (σ, M, pk_1 ,pk_2).
- Such that verification passes under both , pk_1 ,pk_2.
- It must be the case that: H(pk_1) = H(pk_2) (assumed to be hard).

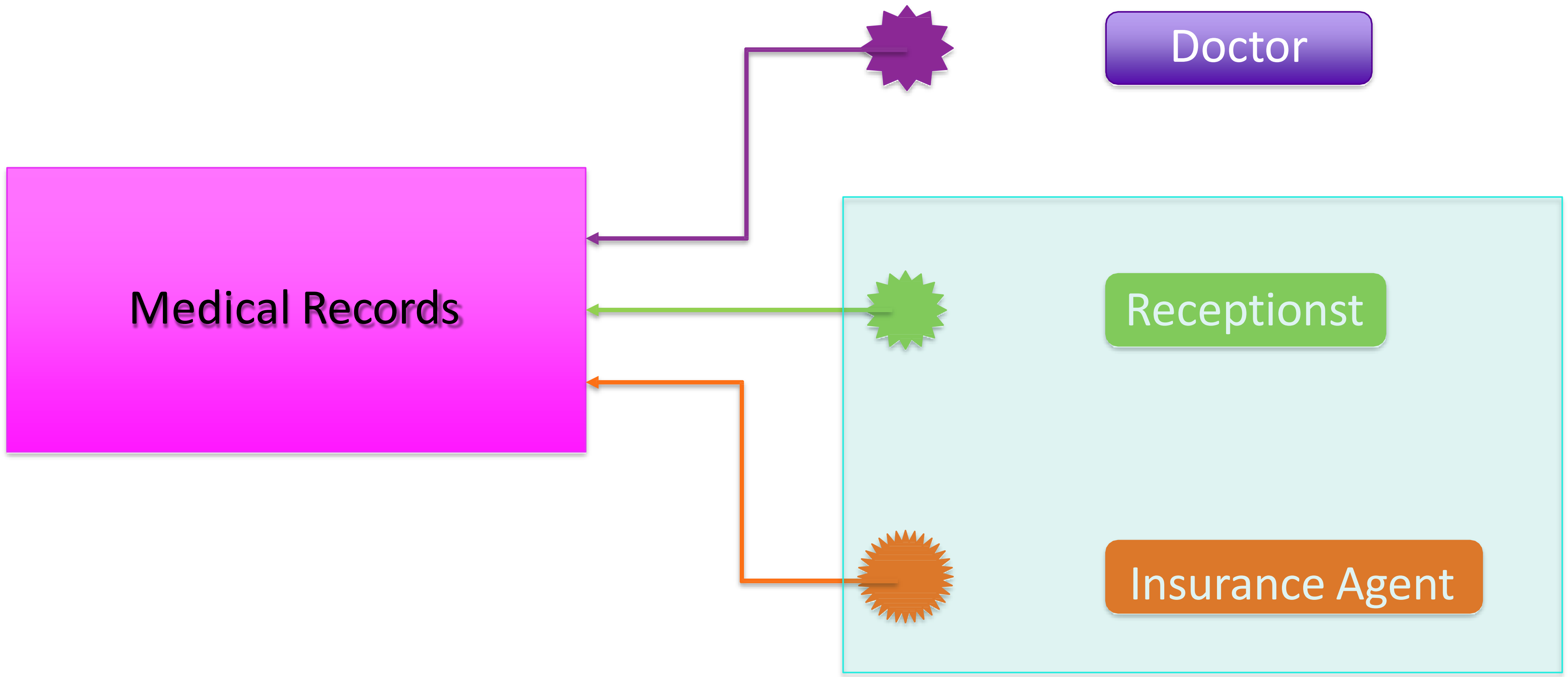
RSAConference2019

Robustness for Functional Encryption

Functional Encryption - An example



Functional Encryption - An example



Functional Encryption - What is it

- Want to compute *f* over some ciphertext and recover *f(M)*.
- Ideally, no other information on *M* is leaked.
- A primitive with many potential applications.

Functional Encryption - Syntax

$prms \twoheadleftarrow Params(1^\lambda)$: produces public parameters.

$(\mathrm{msk}, \mathrm{mpk}) \twoheadleftarrow \textbf{Setup}(1^\lambda)$: outputs the master secret/public keys.

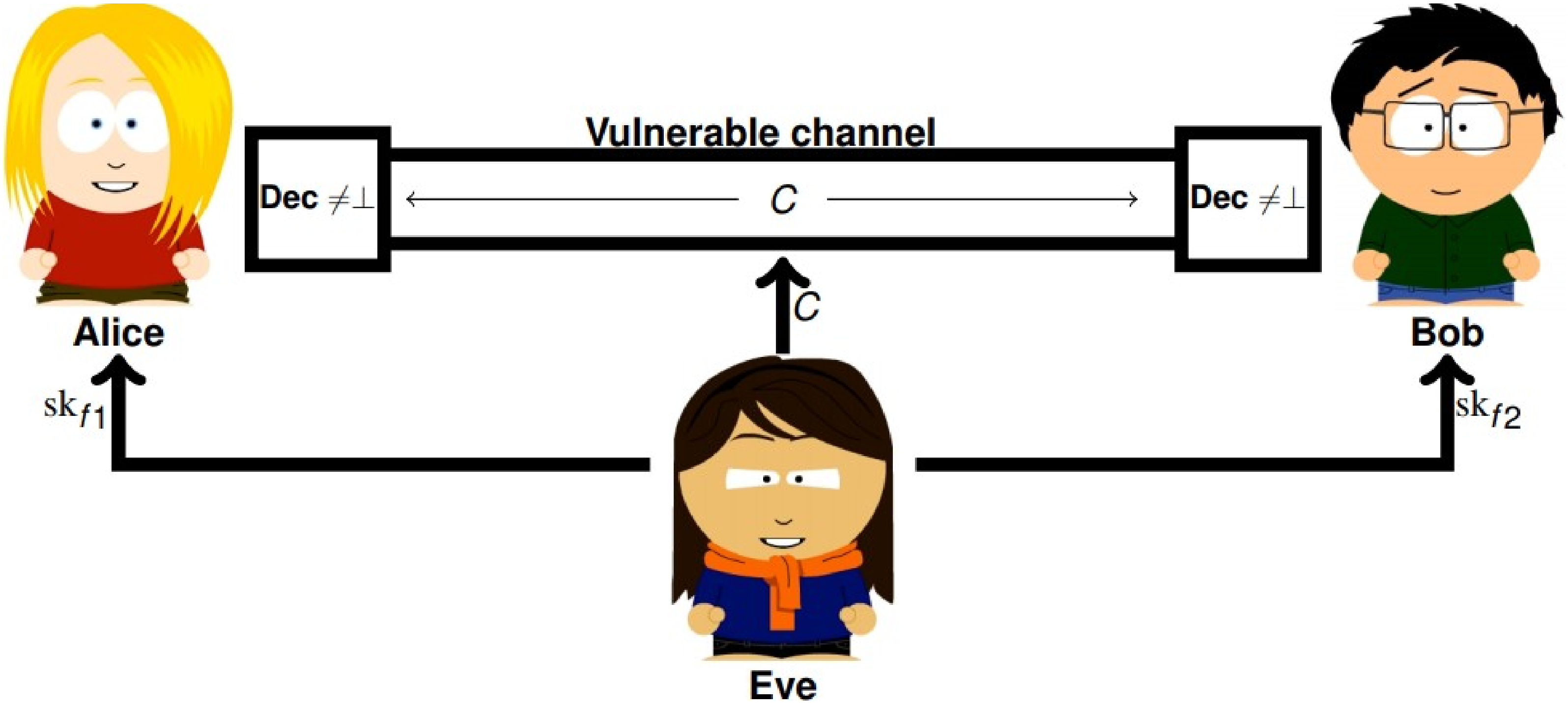
$\mathrm{sk}_f \twoheadleftarrow \textbf{Gen}(\mathrm{msk}, f)$: given the master secret key and a function f, outputs a corresponding sk_f.

$C \twoheadleftarrow \textbf{Enc}(\mathrm{mpk}, M)$: encrypts the plaintext M with respect to mpk.

$\textbf{Dec}(C, \mathrm{sk}_f)$: decrypts the ciphertext C using the functional key sk_f.

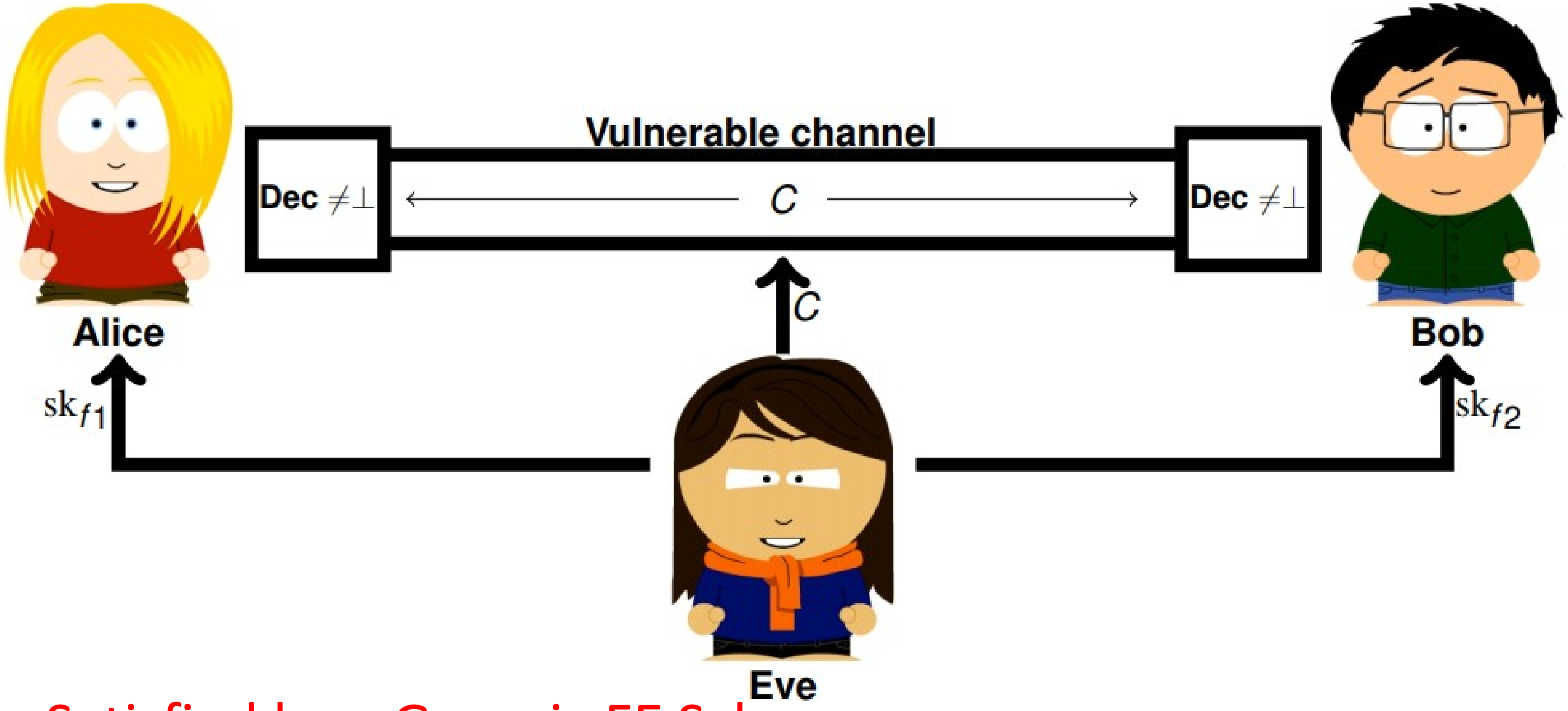
Robust Functional Encryption

Functional Encryption - Defining robustness



Robust Functional Encryption

Functional Encryption - Defining robustness



Issue: Trivially Satisfied by a Generic FE Scheme.

Robust Functional Encryption

- What is the intuition of robustness for FE?
- Why are we defining this notion?
- Any real attacks?

Robust Functional Encryption

- What is the intuition of robustness for FE?
- Why are we defining this notion?
- Any real attacks?

Consider the following inner-product FE scheme

$$\mathrm{msk} \twoheadleftarrow \vec{s}$$
$$\mathrm{mpk} \twoheadleftarrow g^{\vec{s}}$$
$$C_{\vec{x}} \twoheadleftarrow (g^{-r}, g^{r \cdot \vec{s} + \vec{x}})$$
$$sk_{\vec{y}} \leftarrow \vec{s}^{\top} \cdot \vec{y}$$
$$Dec(C_{\vec{x}}, sk_{\vec{y}}) = \vec{x}^{\top} \cdot \vec{y}$$

Robust Functional Encryption

- What is the intuition of robustness for FE?
- Why are we defining this notion?
- Any real attacks?

Consider the following inner-product FE scheme

$$\mathrm{msk} \twoheadleftarrow \vec{s}$$
$$\mathrm{mpk} \twoheadleftarrow g^{\vec{s}}$$
$$C_{\vec{x}} \twoheadleftarrow (g^{-r}, g^{r\cdot\vec{s}+\vec{x}})$$
$$sk_{\vec{y}} \leftarrow \vec{s}^{\top} \cdot \vec{y}$$
$$Dec(C_{\vec{x}}, sk_{\vec{y}}) = \vec{x}^{\top} \cdot \vec{y}$$

$$\mathrm{msk} \twoheadleftarrow \vec{s'}$$
$$\mathrm{mpk} \twoheadleftarrow g^{\vec{s'}}$$
$$C_{\vec{x}} \twoheadleftarrow (g^{-r}, g^{r\cdot\vec{s'}+\vec{x'}})$$
$$sk_{\vec{y}} \leftarrow \vec{s'}^{\top} \cdot \vec{y}$$
$$Dec(C_{\vec{x}}, sk_{\vec{y}}) = \vec{x'}^{\top} \cdot \vec{y}$$

Robust Functional Encryption

- What is the intuition of robustness for FE?
- Why are we defining this notion?
- Any real attacks?

Consider the following inner-product FE scheme

$$\mathrm{msk} \twoheadleftarrow \vec{s}$$
$$\mathrm{mpk} \twoheadleftarrow g^{\vec{s}}$$
$$C_{\vec{x}} \twoheadleftarrow (g^{-r}, g^{r\cdot\vec{s}+\vec{x}})$$
$$sk_{\vec{y}} \leftarrow \vec{s}^{\top} \cdot \vec{y}$$
$$Dec(C_{\vec{x}}, sk_{\vec{y}}) = \vec{x}^{\top} \cdot \vec{y}$$

$$\mathrm{msk} \twoheadleftarrow \vec{s'}$$
$$\mathrm{mpk} \twoheadleftarrow g^{\vec{s'}}$$
$$C_{\vec{x}} \twoheadleftarrow (g^{-r}, g^{r\cdot\vec{s'}+\vec{x'}})$$
$$sk_{\vec{y}} \leftarrow \vec{s'}^{\top} \cdot \vec{y}$$
$$Dec(C_{\vec{x}}, sk_{\vec{y}}) = \vec{x'}^{\top} \cdot \vec{y}$$

Issue: same ciphertext decrypts under two different keys!

Robust Functional Encryption

A possible definition:

Robustness: ciphertext can't be decrypted under two different keys.

Robust Functional Encryption

A possible definition:

Robustness: ciphertext can't be decrypted under two different keys.

For FE: keys are issued via different master secret keys.

Robust Functional Encryption - Security Model

$\underline{\mathrm{FEROB}_{\mathsf{FE}}^{\mathcal{A}}(\lambda)}$:

$(\mathrm{mpk}_1, \mathrm{msk}_1, R_1, M_1, f_1, R_{f_1},$
$\quad \mathrm{mpk}_2, \mathrm{msk}_2, R_2, M_2, f_2, R_{f_2}) \twoheadleftarrow \mathcal{A}(1^\lambda)$

$C_1 \twoheadleftarrow \mathbf{Enc}(\mathrm{mpk}_1, M_1; R_1)$
$C_2 \twoheadleftarrow \mathbf{Enc}(\mathrm{mpk}_2, M_2; R_2)$

if $C_1 = C_2 \wedge \mathrm{mpk}_1 \neq \mathrm{mpk}_2$:
$\quad \mathrm{sk}_{f_1} \twoheadleftarrow \mathbf{KDer}(\mathrm{msk}_1, f_1; R_{f_1})$
$\quad \mathrm{sk}_{f_2} \twoheadleftarrow \mathbf{KDer}(\mathrm{msk}_2, f_2; R_{f_2})$

$\quad$ if $\mathbf{Dec}(C, \mathrm{sk}_{f_1}) \neq \perp \ \wedge\ \mathbf{Dec}(C, \mathrm{sk}_{f_2}) \neq \perp$:
$\quad\quad$ return 1
return 0

Robust Functional Encryption - Security Model

Winning conditions:

- No ciphertext obtained for Authority 1 can be decrypted under a functional key obtained under Authority 2.

- SROB for FE: adversary finds C, sk_{f1}, sk_{f2} such that

$$\mathsf{FE.Dec}(\mathrm{sk}_{f_1}, C) \neq \perp \wedge \mathsf{FE.Dec}(\mathrm{sk}_{f_2}, C) \neq \perp .$$

- We have that FEROB $\Rightarrow$ SROB.

Robust Public-Key FE - Generic Transform

$\underline{\overline{\mathsf{Gen}}(1^\lambda):}$
$(\mathrm{mpk}, \mathrm{msk}) \twoheadleftarrow \mathsf{FE.Gen}(1^\lambda)$
$\overline{\mathrm{mpk}} \leftarrow \mathrm{mpk}$
$\overline{\mathrm{msk}} \leftarrow \mathrm{msk}$
return $(\overline{\mathrm{msk}}, \overline{\mathrm{mpk}})$

$\underline{\overline{\mathsf{KDer}}(\overline{\mathrm{msk}}, f):}$
$\mathrm{msk}, \leftarrow \overline{\mathrm{msk}}$
$\mathrm{sk}_f \twoheadleftarrow \mathsf{FE.KDer}(\mathrm{msk}, f)$
$\overline{\mathrm{sk}_f} \leftarrow \mathrm{sk}_f$
return $\overline{\mathrm{sk}_f}$

$\underline{\overline{\mathsf{Enc}}(\overline{\mathrm{mpk}}, M):}$
$\mathrm{mpk} \leftarrow \overline{\mathrm{mpk}}$
$C_1 \twoheadleftarrow \mathsf{FE.Enc}(\mathrm{mpk}, M)$
$C_2 \twoheadleftarrow \mathbf{H}(\mathrm{mpk}||C_1)$
$\overline{C} \leftarrow (C_1, C_2)$
return $\overline{C}$

$\underline{\overline{\mathsf{Dec}}(\overline{\mathrm{sk}_f}, C):}$
$\mathrm{sk}_f \leftarrow \overline{\mathrm{sk}_f}$
$(C_1, C_2) \leftarrow \overline{C}$
if $\mathbf{H}(\mathrm{mpk}||C_1) \neq C_2$:
    return $\perp$
return $\mathsf{FE.Dec}(\mathrm{sk}_f, C_1)$

Robust Public-Key FE - Generic Transform

$\overline{\mathsf{Gen}}(1^\lambda)$:

$(\mathrm{mpk}, \mathrm{msk}) \twoheadleftarrow \mathsf{FE.Gen}(1^\lambda)$
$\overline{\mathrm{mpk}} \leftarrow \mathrm{mpk}$
$\overline{\mathrm{msk}} \leftarrow \mathrm{msk}$
return $(\overline{\mathrm{msk}}, \overline{\mathrm{mpk}})$

$\overline{\mathsf{KDer}}(\overline{\mathrm{msk}}, f)$:

$\mathrm{msk}, \leftarrow \overline{\mathrm{msk}}$
$\mathrm{sk}_f \twoheadleftarrow \mathsf{FE.KDer}(\mathrm{msk}, f)$
$\overline{\mathrm{sk}_f} \leftarrow \mathrm{sk}_f$
return $\overline{\mathrm{sk}_f}$

$\overline{\mathsf{Enc}}(\overline{\mathrm{mpk}}, M)$:

$\mathrm{mpk} \leftarrow \overline{\mathrm{mpk}}$
$C_1 \twoheadleftarrow \mathsf{FE.Enc}(\mathrm{mpk}, M)$
$C_2 \twoheadleftarrow \mathbf{H}(\mathrm{mpk}||C_1)$
$\overline{C} \leftarrow (C_1, C_2)$
return $\overline{C}$

$\overline{\mathsf{Dec}}(\overline{\mathrm{sk}_f}, C)$:

$\mathrm{sk}_f \leftarrow \overline{\mathrm{sk}_f}$
$(C_1, C_2) \leftarrow \overline{C}$
if $\mathbf{H}(\mathrm{mpk}||C_1) \neq C_2$:
    return $\perp$
return $\mathsf{FE.Dec}(\mathrm{sk}_f, C_1)$

FEROB : follows from $\mathbf{H}(\mathrm{mpk}_1||C_1) = \mathbf{H}(\mathrm{mpk}_2||C_1)$

Summary - Robust DS and FE

DS: signature can't be verified w.r.t. multiple keys.

FE: ciphertext can't be decrypted w.r.t. keys issued by different *msk.*

- Under correct key-generation any unforgeable DS scheme is SROB-secure.
- Generic constructions based on collision-resistant hashes and collision-resistant PRFs.
- FEROB: harder to achieve for Private-Key Functional Encryption.

Summary - Robust DS and FE

DS: signature can't be verified w.r.t. multiple keys.

FE: ciphertext can't be decrypted w.r.t. keys issued by different *msk*.

- Under correct key-generation any unforgeable DS scheme is SROB-secure.
- Generic constructions based on collision-resistant hashes and collision-resistant PRFs.
- FEROB: harder to achieve for Private-Key Functional Encryption.