Violating Key Separation: On Using One Secret Key for Two Purposes

Rainer Steinwandt

based on joint work with
María Isabel González Vasco and Florian Hess

Textbook RSA again

Naïve encryption with RSA:

$m \rightarrow m^e \bmod n$

Naïve signing with RSA:

$m \rightarrow m^d \bmod n$

… clearly, here a signing oracle helps to decrypt ciphertexts

… but what about real schemes with full padding, like RSA-PSS and RSA-OAEP?

Why violating key separation?

Principle of *key separation* tells us **not** to use the same key for signing and decrypting—so why should we?

- simpler key management: “halve the number of keys”
- protocols using both signatures & asymmetric encryption
- fall-back mechanism if one PKI becomes unavailable

In identity-based cryptography:
option to have one key generation center only

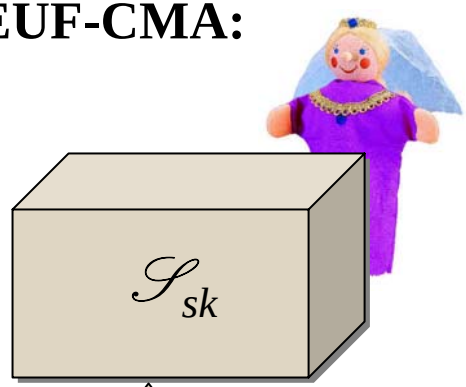
The usual ppt suspects

- **IND-CCA:**

c≠challenge

pk

- **EUF-CMA:**



pk

A combined public key scheme

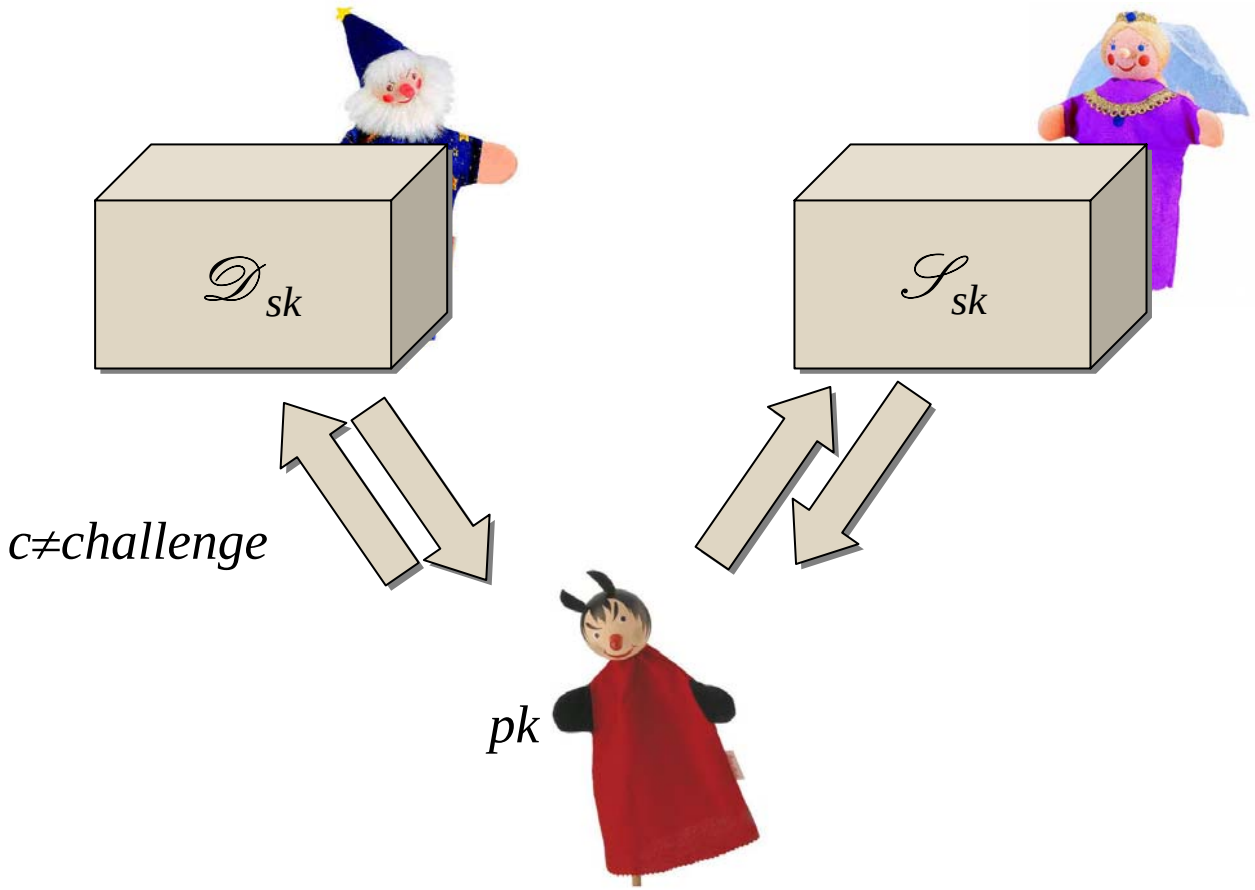
… is a tuple of 5 polynomial time (in the security param. k) algorithms:

- a prob. key generation $\mathscr{K}$ to generate **one** key pair (pk, sk)
- a prob. encryption algorithm $\mathscr{E}_{pk}$
- a det. signature verification $\mathscr{V}_{pk}$

  same pk
- a prob. signature generation $\mathscr{S}_{sk}$
- a det. decryption algorithm $\mathscr{D}_{sk}$

  same sk

IND-C[CM]A *and* EUF-C[CM]A



Note: Submitting the (IND-)challenge to $\mathscr{S}_{sk}$ is okay.

Putting things into context…

- **(insider-)secure signcryption**: similar requirements, but standard construction to derive a (stand-alone) signature or encryption scheme uses two keys per user
- **universal padding schemes**: design padding schemes that can be used to achieve both IND-CCA and EUF-CMA
- **UC with joint state**

Here: focus on combining "given" encryption and signature schemes, not specifically designed to be combined

IND-CCA *does not suffice*

- Take an IND-CCA secure encryption scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ & an EUF-CMA signature scheme $(\mathcal{K}, \mathcal{S}, \mathcal{V})$.
- Now define—contrived, but secure—schemes $(\mathcal{K}^*, \mathcal{E}, \mathcal{D}^*)$ and $(\mathcal{K}^*, \mathcal{S}^*, \mathcal{V}^*)$:
  - $\mathcal{K}^*$ outputs also a secret random string r that cannot occur as ciphertext and adds $H(r)$ to pk
  - $\mathcal{D}^*$ reveals the secret key on input r, otherwise runs $\mathcal{D}$
  - $\mathcal{S}^*$ runs $\mathcal{S}$ to obtain signature σ and returns (σ, r)
  - $\mathcal{V}^*$ checks correctness of r and σ (using $H(r)$ and $\mathcal{V}$)

⇒ **Combined scheme $(\mathcal{K}^*, \mathcal{E}, \mathcal{D}^*, \mathcal{S}^*, \mathcal{V}^*)$ is insecure.**

When is combining two schemes okay?

No general characterization, but the use of a particular proof technique in the security reduction can be of great help to identify a secure combined scheme.

Here: ElGamal encryption+Fujisaki-Okamoto conversion with Pointcheval-Stern variant of **ElGamal signature** yields both IND-C[CM]A and EUF-C[CM]A.

ElGamal encryption+Fujisaki-Okamoto

$\mathscr{K}$: outputs public key (q, g, g^x) with $\mathrm{ord}(g)=q$ and

secret key $x \in_R \{0,\ldots, q-1\}$

$\mathscr{E}$: for plaintext $m \in \{0,1\}^k$ outputs (c_1, c_2, c_3) with

- $c_1 := r\cdot(g^x)^{H_2(r,m)}$
- $c_2 := g^{H_2(r,m)}$
- $c_3 := H_1(r) \oplus m$

… using $r \in_R \langle g \rangle$ and random oracles H_1, H_2

$\mathscr{D}$: recovers r and m and checks consistency with c_1

⇒ **IND-CPA of ElGamal yields IND-CCA (RO model)**

Modified ElGamal signature (MEG)

$\mathscr{K}$: outputs public key (q, g, g^x) with g generating a subgroup of $\mathbf{Z}/p\mathbf{Z}^*$ of order q, and secret key $x \in_R \{0,\ldots, q-1\}$

$\mathscr{S}$: for message $m \in \{0,1\}^*$ outputs (r, h, s) with

- $r := g^K$
- $h := F(m,r)$
- s satisfying $h = x \cdot r + K \cdot s$

… using $K \in_R \{1,\ldots, q-1\}$ and random oracle F

$\mathscr{V}$: checks if $g^h = (g^x)^r \cdot r^s$

⇒ **fits pattern of a "generic signature scheme"**

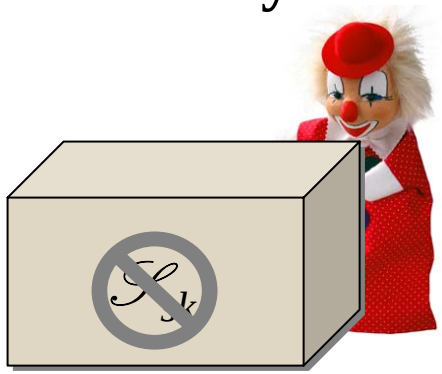
The security proof of MEG helps us

- Forking lemma shows that a passive adversary can be used to obtain two tuples (m, r, h, s), (m, r, h', s') with $h \neq h'$ and $s \neq s'$. With this, $\log_g(g^x)$ can be found.

  ⇒ passive forger enables recovery of secret key

- Signing oracle can be simulated **without secret key** with indistinguishable distribution

  ⇒ EUF-CMA security (with RO)



A secure combination

To prove the combined public key scheme secure, we can build on “key-free simulation” of the signing oracle:

- EUF-C[CM]A adversary is shown to imply discrete logarithm computation with decryption oracle only
- IND-C[CM]A attack is reduced to IND-CCA attack without signing oracle

“Naïve combination of (ElGamal+Fujisaki-Okamoto) & MEG offers strong guarantees.”

Combining IBE and IBS

- **Identity-based encryption (IBE):**

  - Encryption alg.: has message & **identity** $id \in \{0,1\}^*$ as input ($\Longrightarrow$ recipient may not know the secret key yet)
  - Decryption alg.: has candidate ciphertext & **secret key, derived by key generation center**, as input

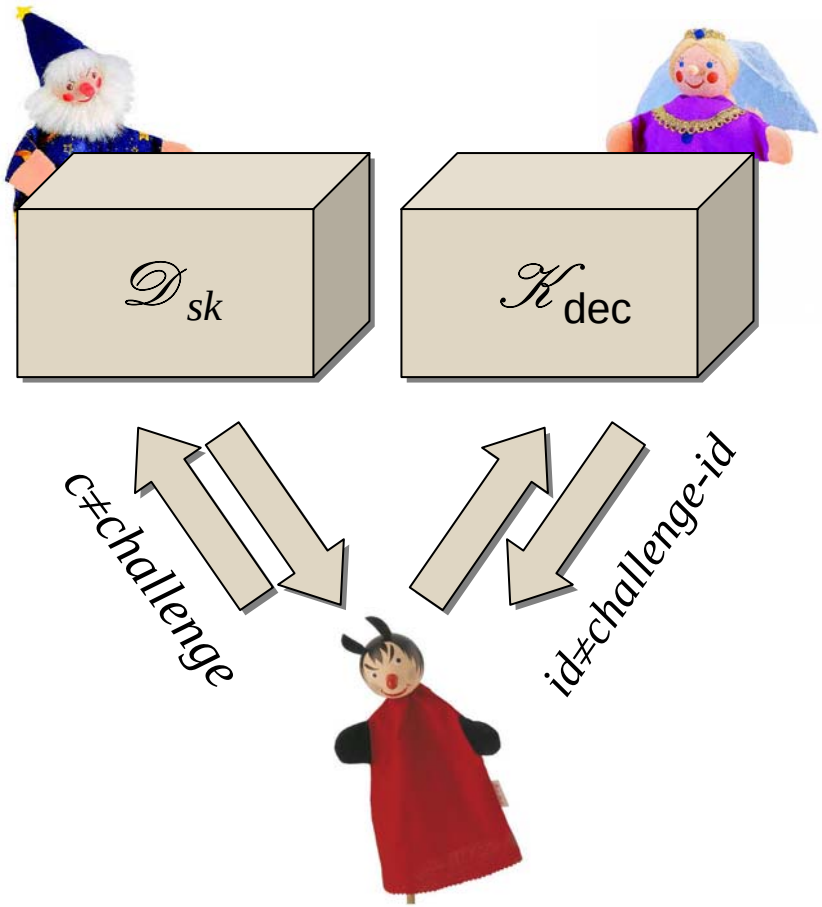
- **Identity-based signature (IBS):**

  - Signing alg.: has message & **secret key, derived by key generation center**, as input
  - Verification alg.: has message & candidate signature as input

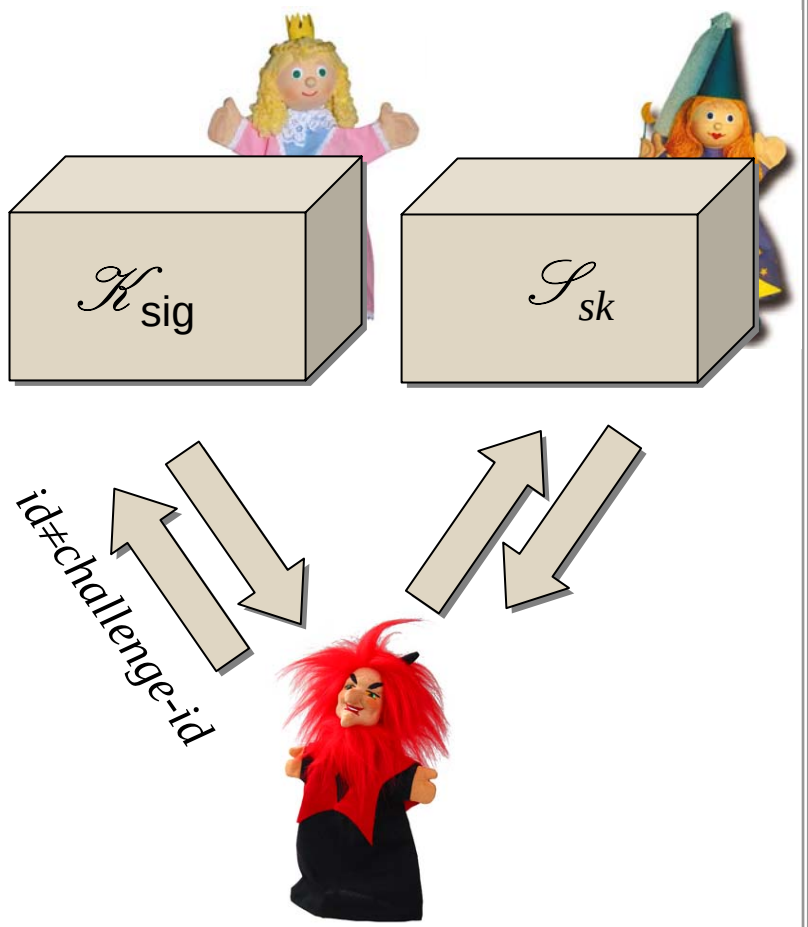
Hope: simpler key management, avoiding certificates

IBE & IBS: the usual ppt suspects

- **IND-ID-CCA:**



- **EUF-ID-CMA:**



Combined id-based public key scheme

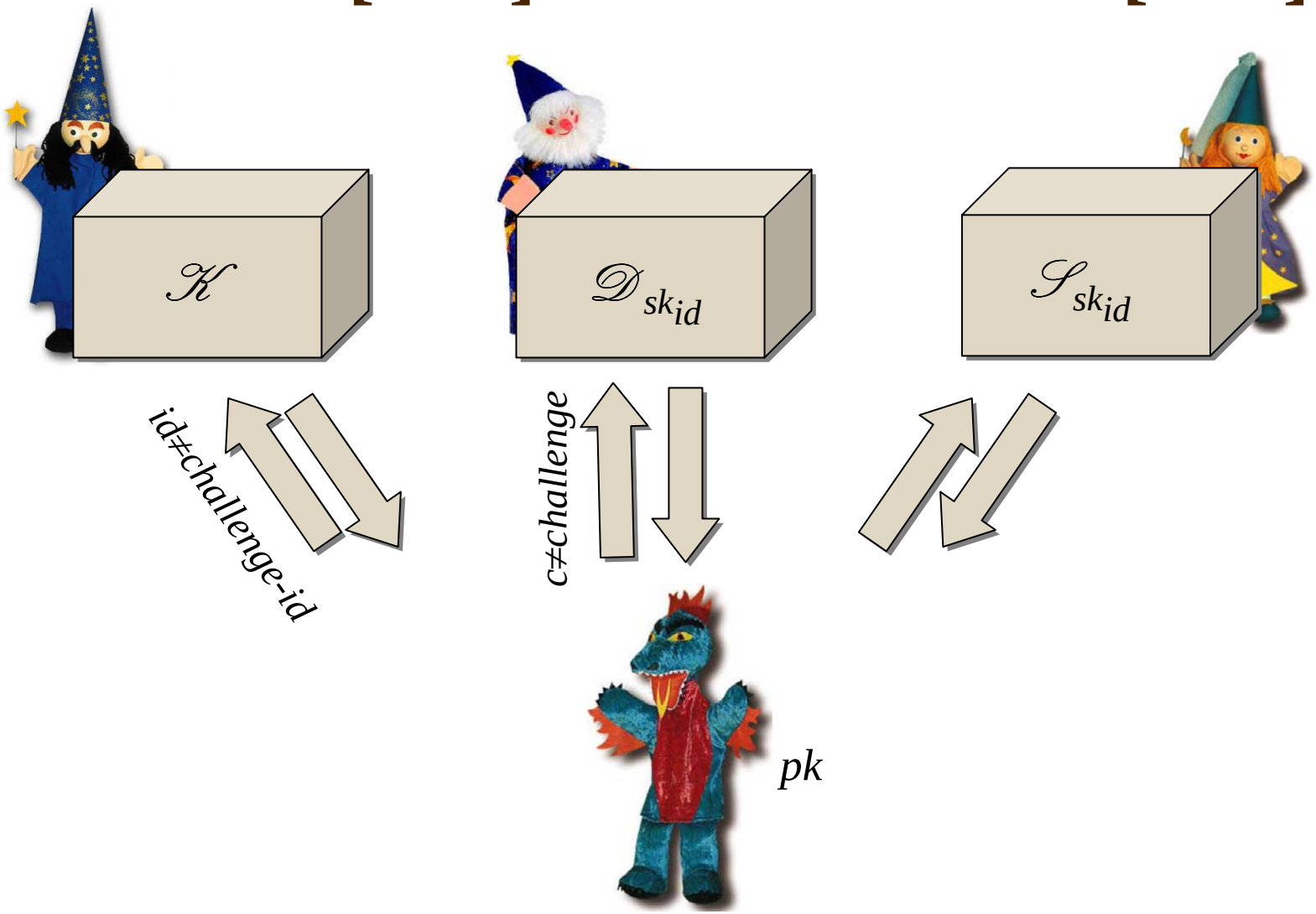
… aims at a setting where we want to have both IBE & IBS:

- $\mathcal{I}$: setup algorithm to derive public system parameter *pk* and a master secret *sk*
- $\mathcal{K}$: on input a user identity *id* extracts **one** secret signing/decryption key sk_{id}
- $\mathcal{E}$: encrypt for identity *id* (as usual in IBE)
- $\mathcal{D}$: decrypt with sk_{id}
- $\mathcal{S}$: sign with sk_{id}
- $\mathcal{V}$: verify for identity *id* (as usual in IBS)

same sk_{id}

one key from a common key generation center

IND-ID-C[CM]A *and* EUF-ID-C[CM]A



Finding secure examples

- No general characterization when IND-ID-CCA combined with EUF-ID-CMA yields a secure combined scheme

Here: combine two pairing-based schemes in the RO model; security proof for combined scheme builds on existence of certain "simulators" for comprising schemes

- Underlying hardness assumption: Bilinear Diffie-Hellman ("Given (P, aP, bP, cP), any ppt $\mathscr{A}$ finds $e(P, P)^{abc}$ with negl. probability only.")

IBE: Boneh and Franklin's FullIdent

Setup algorithm $\mathcal{I}$: chooses a random generator $P \in_R G$. Moreover, $Y_{\text{master}} := sk \cdot P$ is published, where $sk \in_R \left(\frac{\mathbb{Z}}{l\mathbb{Z}}\right)^{\times}$ is the uniformly at random chosen master key.

Key extraction $\mathcal{K}_{\text{dec}}$: for an identity $id \in \{0,1\}^*$, the secret key is $sk_{id} := sk \cdot Q_{id}$, where $Q_{id} := H_1(id) \in G^*$.

Encryption algorithm $\mathcal{E}$: to encrypt a message $m \in \{0,1\}^n$ under the identity id, the following steps are performed:

- compute $Q_{id} := H_1(id) \in G^*$
- choose a random $\sigma \in_R \{0,1\}^n$ and set $r := H_3(\sigma, m) \in \left(\frac{\mathbb{Z}}{l\mathbb{Z}}\right)^{\times}$
- compute $g_{id} := e(Q_{id}, Y_{\text{master}}) \in V$

The ciphertext is $c := (r \cdot P, \sigma \oplus H_2(g_{id}^r), m \oplus H_4(\sigma))$.

Decryption algorithm $\mathcal{D}$: To decrypt a candidate ciphertext $c = (U, v, w)$ with secret key sk_{id}, the subsequent steps are performed:

- if $U \notin G^*$, the error symbol $\perp$ is returned
- compute $\sigma := v \oplus H_2(e(sk_{id}, U))$
- let $m := w \oplus H_4(\sigma)$ and $r := H_3(\sigma, m)$
- if $U \neq r \cdot P$ return the error symbol $\perp$, otherwise output m as decryption of c

IBS: Hess' signature scheme

Setup algorithm $\mathcal{I}$: chooses a random generator $P \in_R G$. Moreover, $Y_{\text{master}} := sk \cdot P$ is published, where $sk \in_R \left(\frac{\mathbb{Z}}{l\mathbb{Z}}\right)^{\times}$ is the uniformly at random chosen master secret key.

Key extraction $\mathcal{K}_{\text{sig}}$: for an identity $id \in \{0,1\}^*$, the secret key corresponding to this identity is computed from the secret master key by the issuing authority as $sk_{id} := sk \cdot H(id)$ and forwarded to the signer.

Signing algorithm $\mathcal{S}$: to sign a message m with sk_{id}, the signer chooses arbitrary $P_1 \in G^*$, picks a random integer $k \in \left(\frac{\mathbb{Z}}{l\mathbb{Z}}\right)^{\times}$ and computes:

- $r := e(P_1, P)^k$
- $v := h(m, r)$
- $u := v \cdot sk_{id} + k \cdot P_1$

The signature on m under sk_{id} then is $(u, v) \in G \times \left(\frac{\mathbb{Z}}{l\mathbb{Z}}\right)^{\times}$.

Verification algorithm $\mathcal{V}$: returns **true** if and only if a candidate signature (u, v) for a message m satisfies the equation $v = h(m, r)$, where r is computed as $r = e(u, P) \cdot e(H(id), -Y_{\text{master}})^v$.

Basic idea for security proof

- IND-ID-C[CM]A: derive ordinary IND-ID-CCA adversary against Boneh-Franklin scheme by simulating signing oracle without using secret key (& suitable RO simulation)

- EUF-ID-C[CM]A: derive ordinary EUF-ID-CMA attack against Hess' signature scheme by simulating decryption oracle without using secret key (& suitable RO simulation)

Boneh-Franklin's FullIdent + Hess' signature scheme: EUF-ID-C[CM]A and IND-ID-C[CM]A

Conclusions

- Naïvely combining (identity-based) encryption and signature schemes **can** result in schemes with strong provable guarantees

- No general characterization of secure combinations: security analysis based on rather specific properties of comprising schemes

More details: `http://eprint.iacr.org/2008/466/`