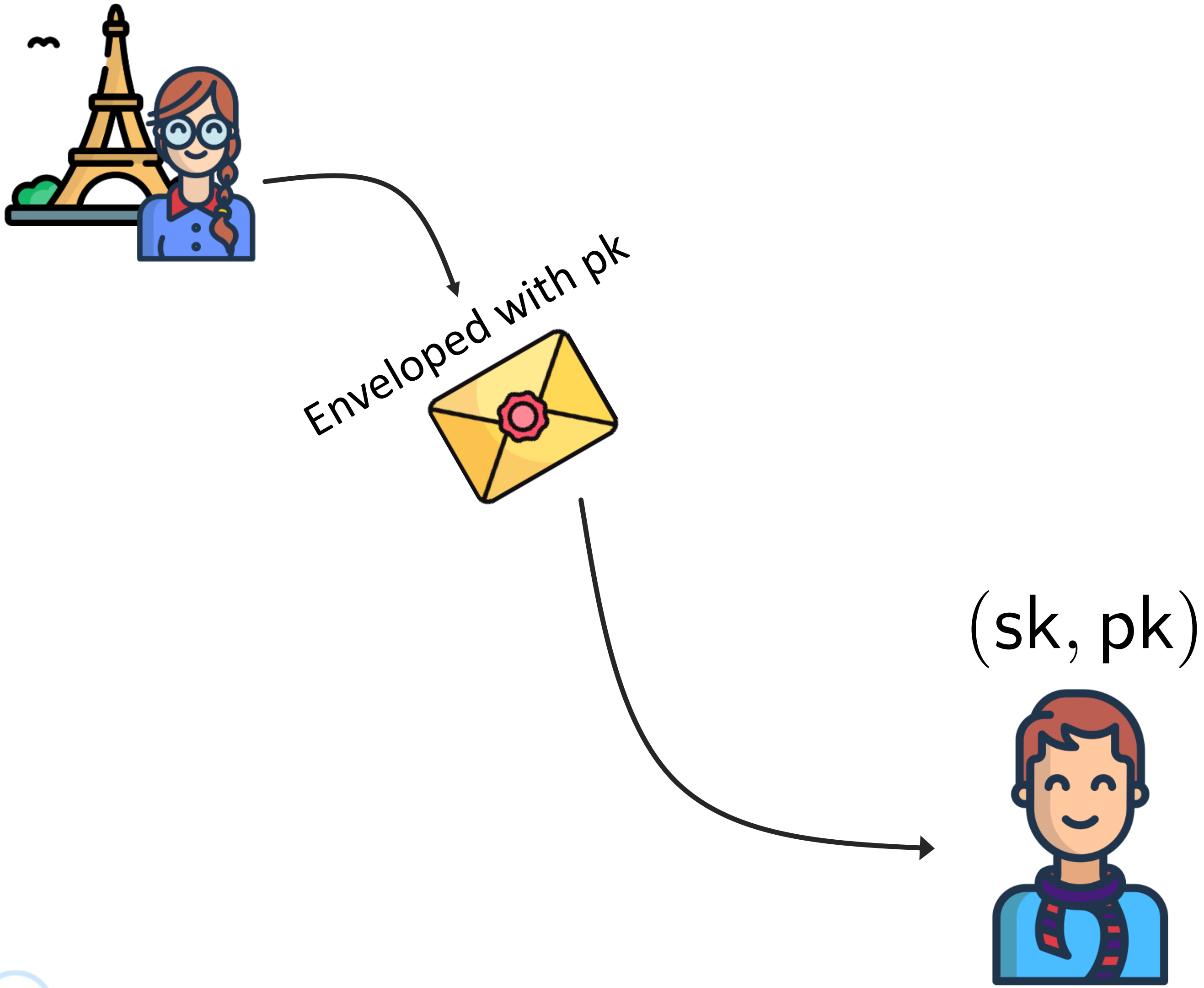
Joint Security of Encryption and Signature in RuCMS: Two Keys are Better, but One is Also Fine

Bozhko A., Babueva A., Kyazhin S.

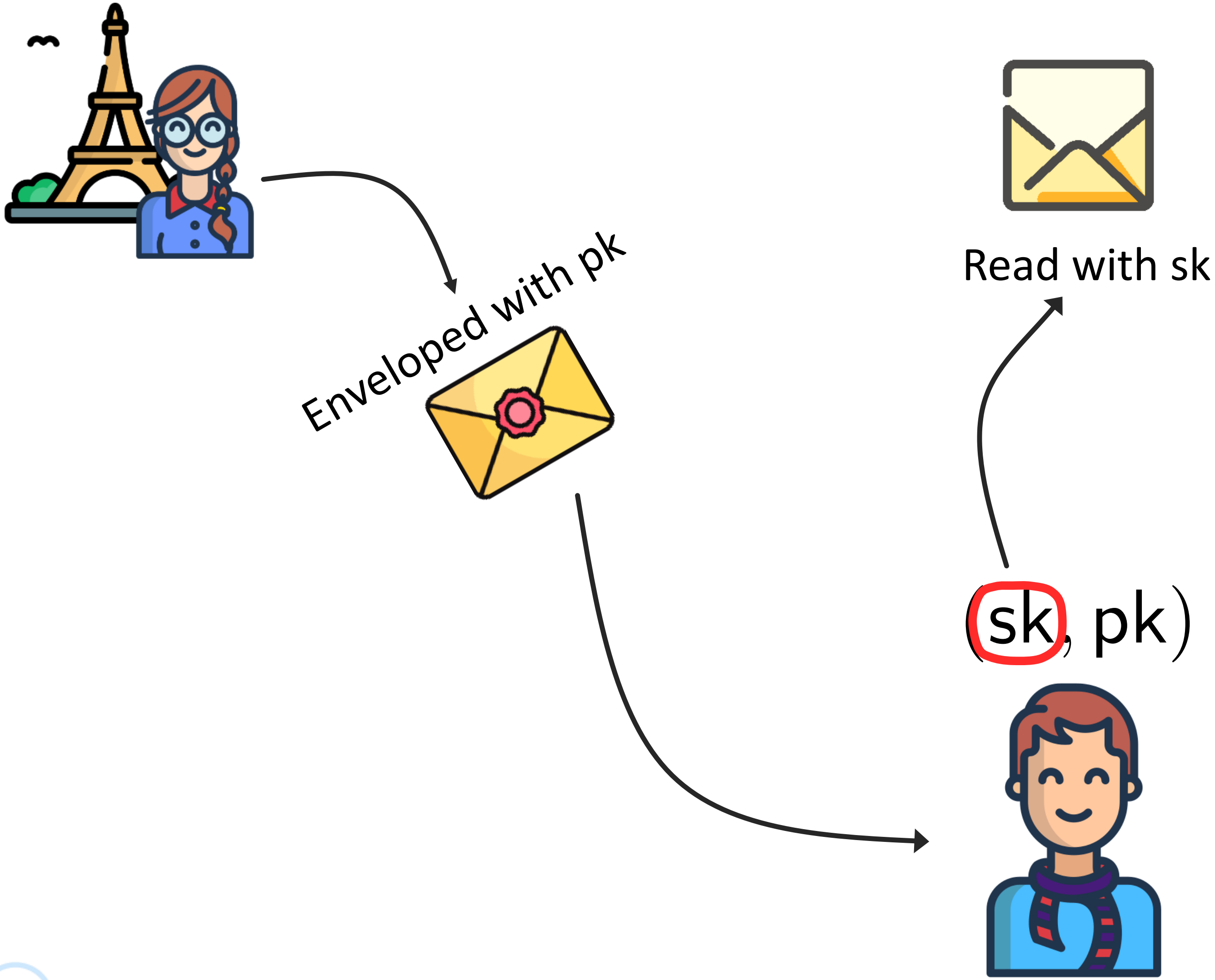
CryptoPro LLC

CTCrypt’2024

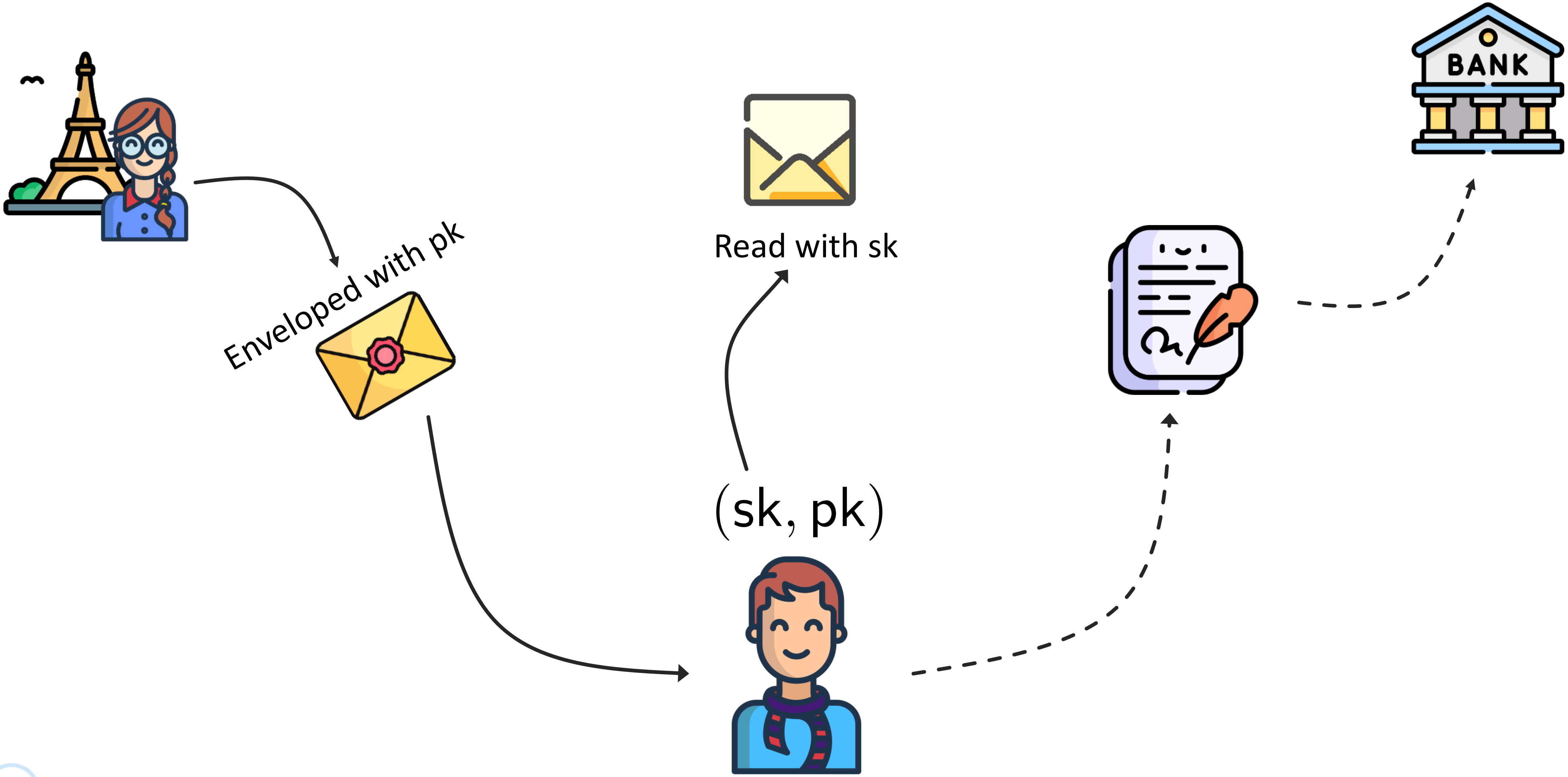
Enveloped Data vs. Signed Data in CMS



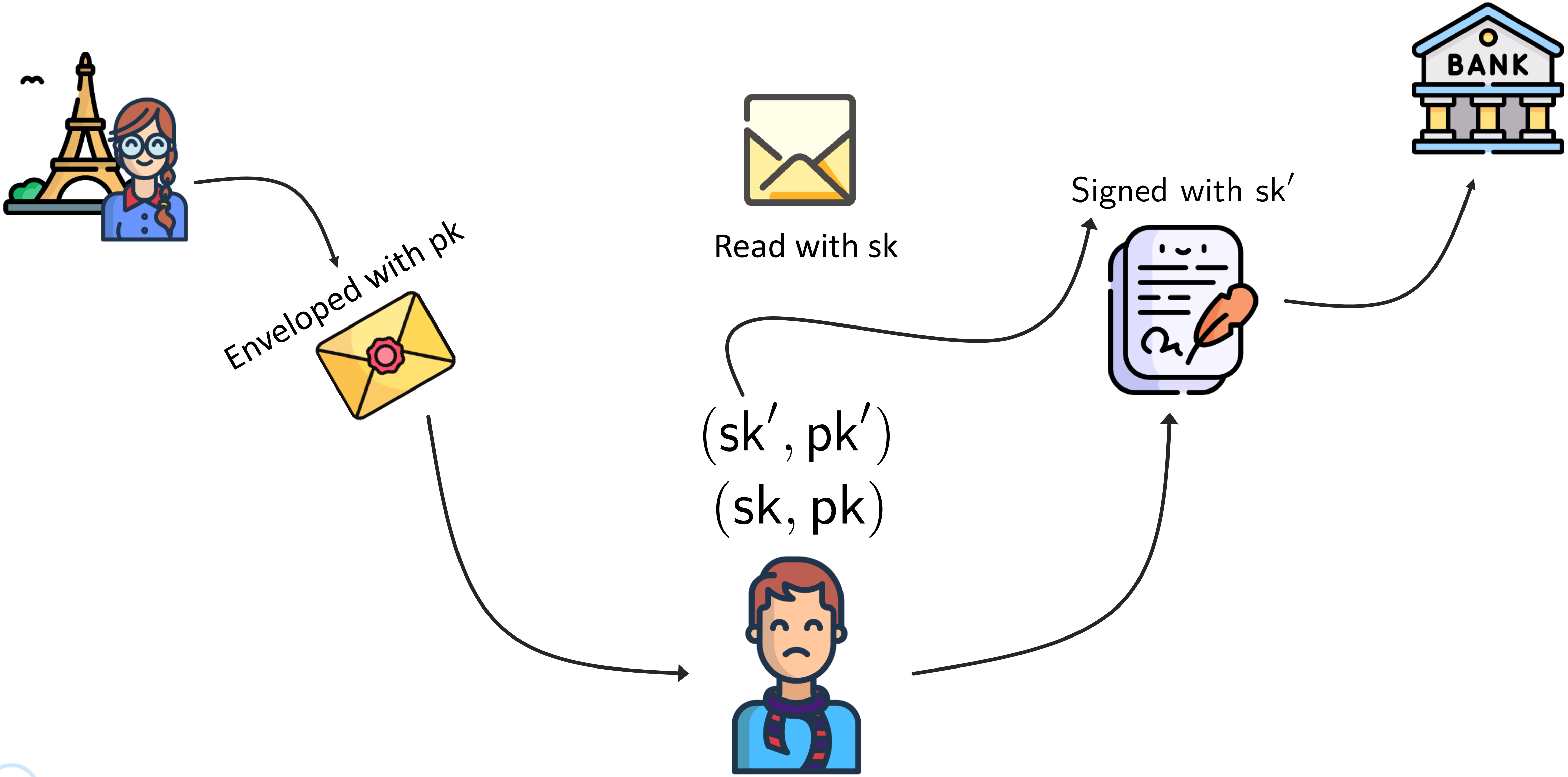
Enveloped Data vs. Signed Data in CMS



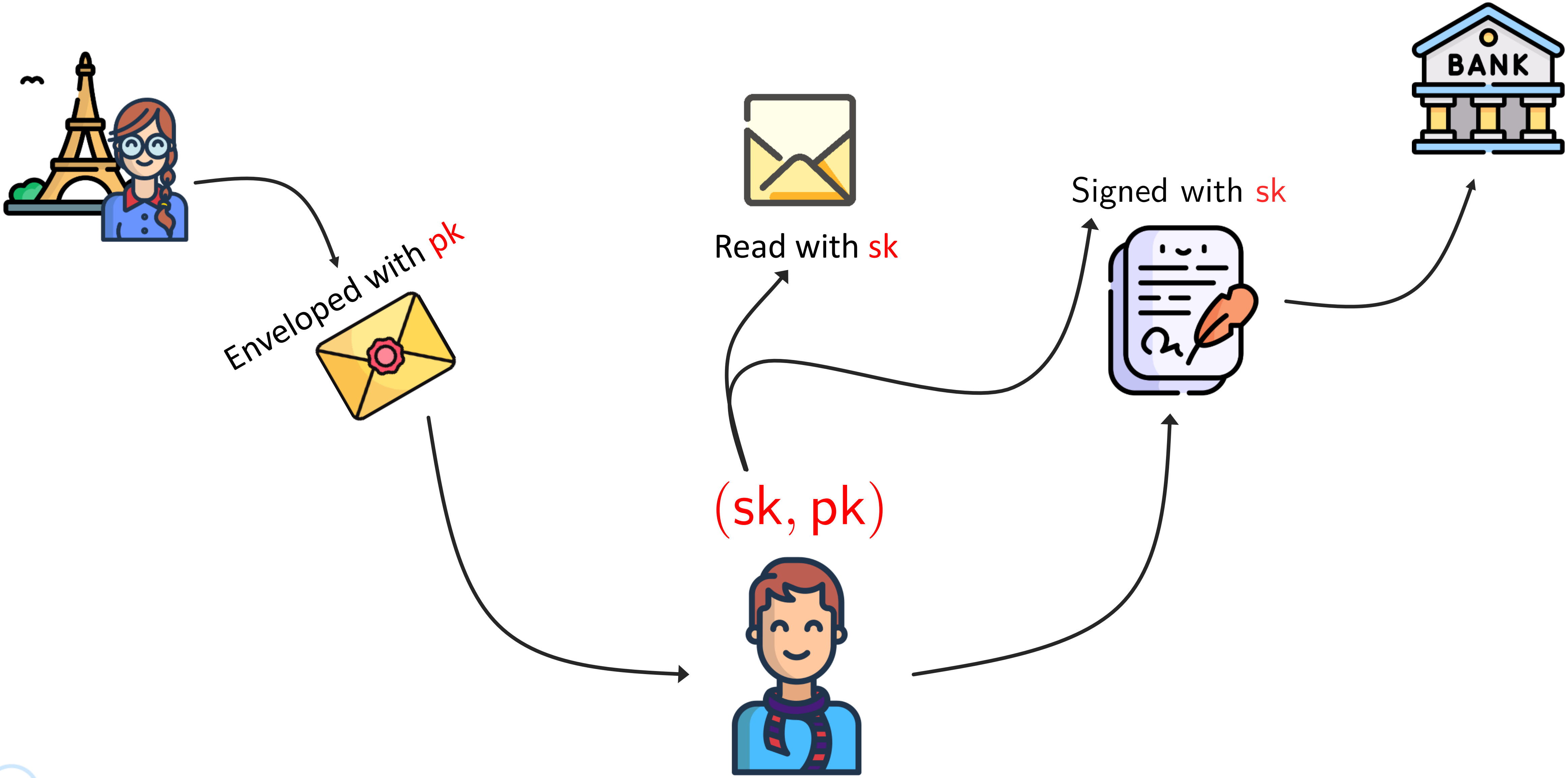
Enveloped Data vs. Signed Data in CMS



Enveloped Data vs. Signed Data in CMS



What if I Use a Single Key?



What if I Use a Single Key?

Pros:

1. Very convenient;

2. ~twice cheaper;

3. Some applications do actually require this, such as proof of possession in a PKCS#10 certificate request for a PKE key.

What if I Use a Single Key?

Cons:

That's not secure in general. For example:

- textbook RSA;
- generic counterexample from [1];
- RSA-based protocols in EMV (Europay + MasterCard + VISA) standards [2].

[1] Paterson, Schuldt, Stam, Thomson. (2011). On the Joint Security of Encryption and Signature, Revisited.

[2] Degabriele, Lehmann, Paterson, Smart, Strefler. (2011). On the Joint Security of Encryption and Signature in EMV.

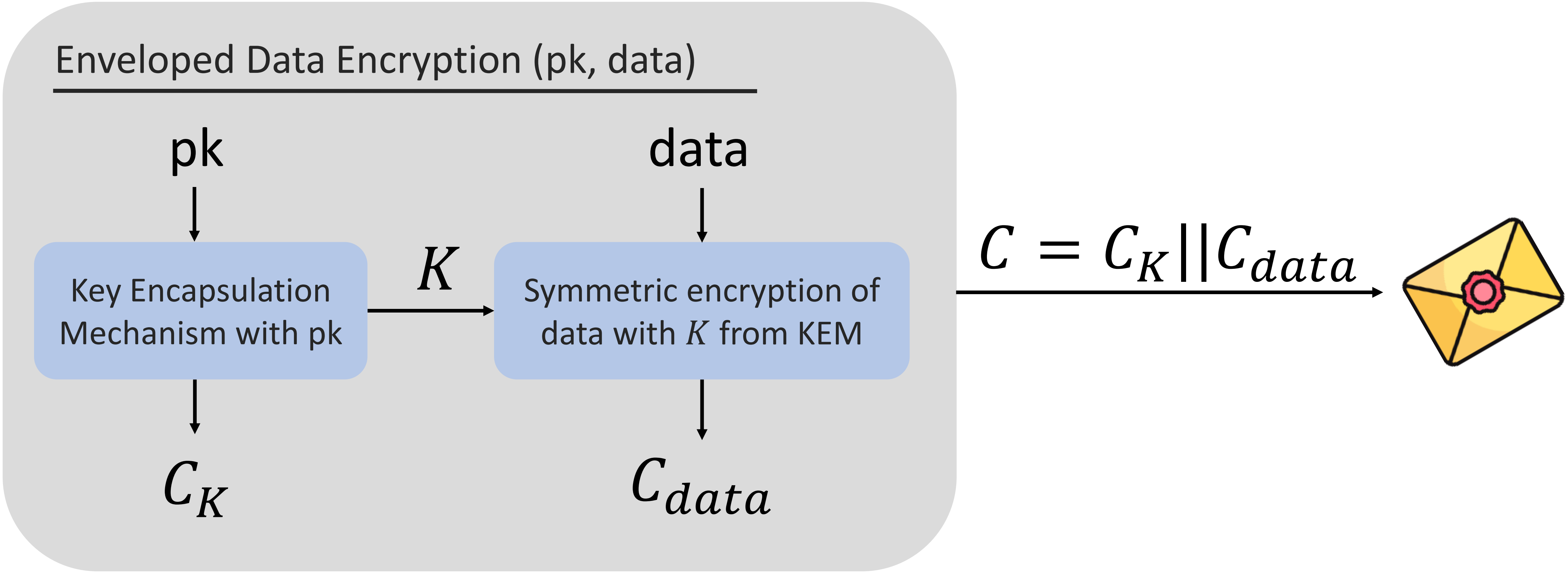
What if I Use a Single Key?

➡️ Some but not all constructions might be secure.

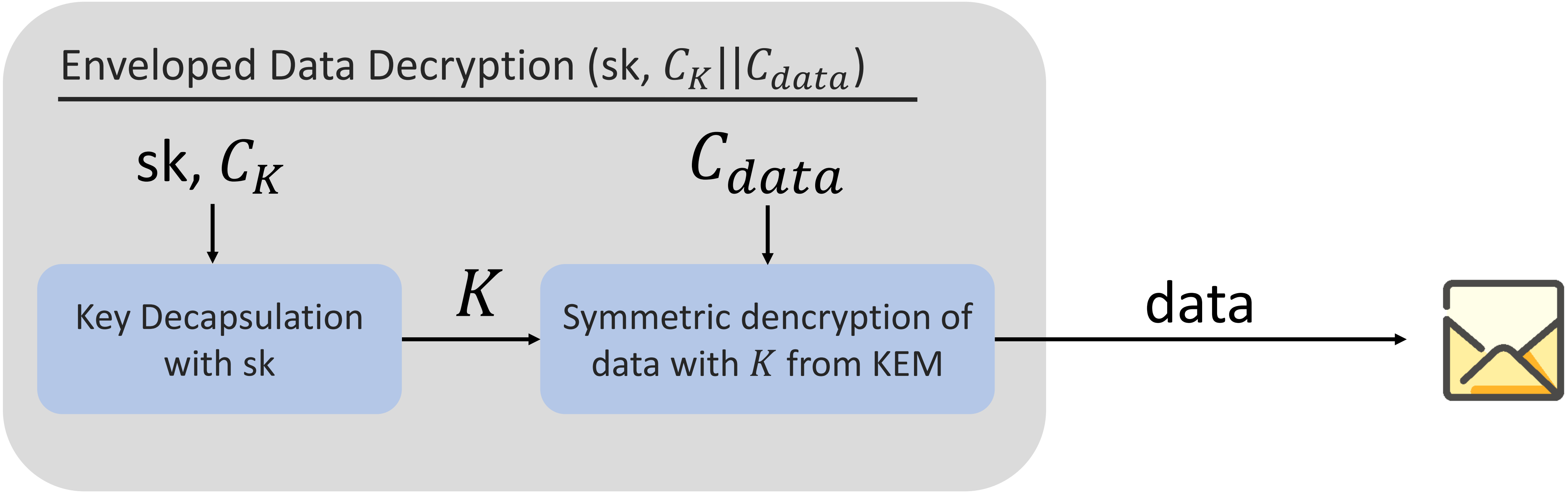
What if I Use a Single Key?

- Some but not all constructions might be secure.
- What about CMS with GOST algorithms?

CMS with GOST Algorithms



CMS with GOST Algorithms



CMS with GOST Algorithms

Signed Data (sk, data)

Basically the GOST signature:

1. pick k at random
2. compute $R = k \cdot P$
3. compute $r = R.x$
4. compute $e = H(data)$
5. compute $s = ke + dr$
6. output (r, s)

(r, s)

CMS with GOST Algorithms and a Single Key

TL;DR

- Enveloped data is a Public Key Encryption scheme constructed from Key Encapsulation Mechanism (KEM) and symmetric encryption (following the PKE = KEM + DEM paradigm).
- Signed data is the GOST signature scheme.

CMS with GOST Algorithms and a Single Key

TL;DR

- Enveloped data is a Public Key Encryption scheme constructed from Key Encapsulation Mechanism (KEM) and symmetric encryption (following the PKE = KEM + DEM paradigm).
- Signed data is the GOST signature scheme.

!! It was shown in [1] that to prove joint security of a PKE scheme based on KEM+DEM paradigm and a signature it is suffice to prove joint security of the KEM and the signature scheme

[1] Degabriele, Lehmann, Paterson, Smart, Strefler. (2011). On the Joint Security of Encryption and Signature in EMV.

KEM in CMS with GOST Algorithms

Key Encapsulation (pk)

1. pick K at random
2. pick ephemeral secret u at random
3. compute ephemeral public $U = uP$
4. compute export key $K_{exp} = F(u \cdot \text{pk})$
5. compute encapsulation
   $$IV||C_{enc} = AE.Enc(K_{exp}, K)$$
6. set $C_K = U||IV||C_{enc}$
7. output K, C_K

Key Decapsulation (pk, C_K)

1. parse C_K as $U||IV||C_{enc}$
2. compute export key $K_{exp} = F(\text{sk} \cdot U)$
3. finally compute $K = AE.Dec(K_{exp}, IV, C_{enc})$
4. output K

P – a generator point of a cyclic subgroup $\mathbb{G}$ of points of an elliptic curve $\mathcal{E}$ of a prime order q;
AE – an Authenticated Encryption scheme;
F – a key derivation function.

What Is a Secure KEM?

IND-CCA

1. pick a key pair (sk, pk) at random
2. pick bit b at random
3. compute $(K, C) = \mathrm{KEM.Enc}(\mathrm{pk})$
4. if $b = 1$ re-pick K at random

Finalize: return $b == b'$

Oracle Dec

Compute $\widetilde{K} = KEM.DEC(\mathrm{sk}, \tilde{C})$

pk, (K, C)

guess b

$\tilde{C} \neq C$

$\widetilde{K}$

What Is a Secure KEM?

IND-CCA

1. pick a key pair (sk, pk) at random
2. pick bit b at random
3. compute $(K, C) = \mathrm{KEM.Enc}(\mathrm{pk})$
4. if $b = 1$ re-pick K at random

Finalize: return $b == b'$

Oracle Dec

Compute $\tilde{K} = KEM.DEC(\mathrm{sk}, \tilde{C})$

✅ Must not be able to guess bit b with a probability much higher than 0.5; i.e., must not be able to distinguish the real K from a randomly chosen one.

Is KEM in GOST CMS Secure?

It can be seen that KEM in GOST is based on the DHIES public key encryption. By adjusting the proof for DHIES, we have proved the following theorem.

Theorem 1. Let $\mathcal{A}$ be an IND-CCA adversary for KEM. Then there exist an AE-CCA adversary $\mathcal{B}$ for AE and an adversary $\mathcal{D}$ solving ODH problem, such that

$$Adv_{KEM}^{INDCCA}(\mathcal{A}) \leq Adv_{AE}^{AECCA}(\mathcal{A}) + Adv_{\mathbb{G},F}^{ODH}(\mathcal{B}) + \frac{N_d}{q-1},$$

where $\mathcal{A}$ makes no more than N_d queries to its Dec oracle and AE-CCA security notion is an IND-CCA2 for authenticated encryption.

Is KEM in GOST CMS Secure?

Wait a second! WHAT is that?!

Theorem 1. Let $\mathcal{A}$ be an IND-CCA adversary for KEM. Then there exist an AE-CCA adversary $\mathcal{B}$ for AE and an adversary $\mathcal{D}$ solving ODH problem, such that

$$Adv_{KEM}^{INDCCA}(\mathcal{A}) \leq Adv_{AE}^{AECCA}(\mathcal{A}) + Adv_{\mathbb{G},F}^{ODH}(\mathcal{B}) + \frac{N_d}{q-1},$$

where $\mathcal{A}$ makes no more than N_d queries to its Dec oracle and AE-CCA security notion is an IND-CCA2 for authenticated encryption.

Oracle Diffie-Hellman Problem

TL;DR

- A modification of the Decisional Diffie-Hellman problem
- An adversary has to distinguish a key derived from a DH from a random key
- An adversary is allowed to derive keys from DH of one of the secrets and arbitrary point

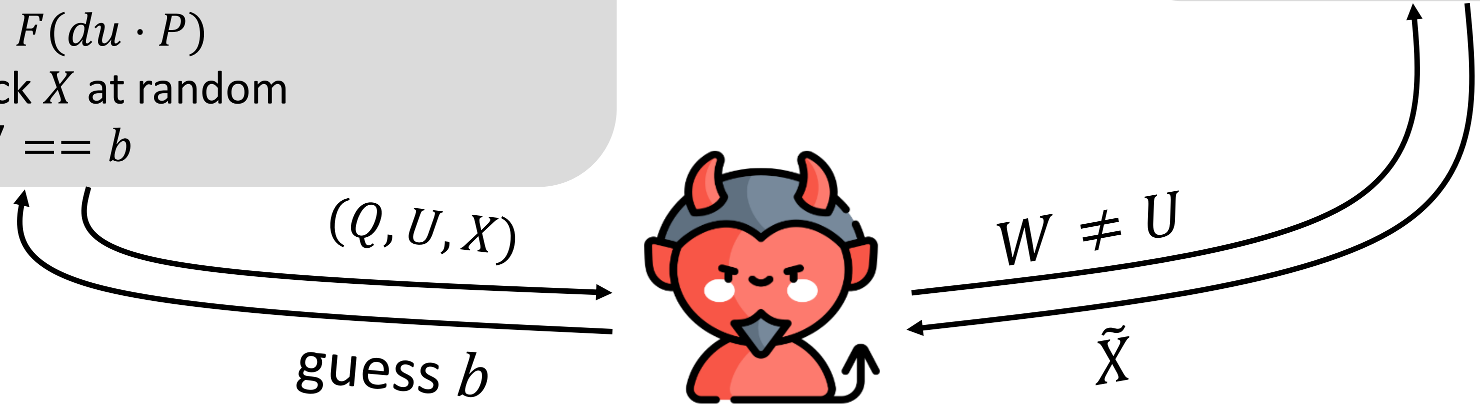
$ODH_{\mathbb{G},F}$

1. pick d at random, compute $Q = d \cdot P$
2. pick u at random, compute $U = u \cdot P$
3. pick bit b at random
4. compute $X = F(du \cdot P)$
5. if $b = 1$ re-pick X at random

Finalize: return $b' == b$

Oracle $fCDH$

Compute $\tilde{X} = F(d \cdot W)$



✅ Must not be able to guess bit b with a probability much higher than 0.5;
i.e., must not be able to distinguish the real X from a randomly chosen one.

What Is a Secure Signature?

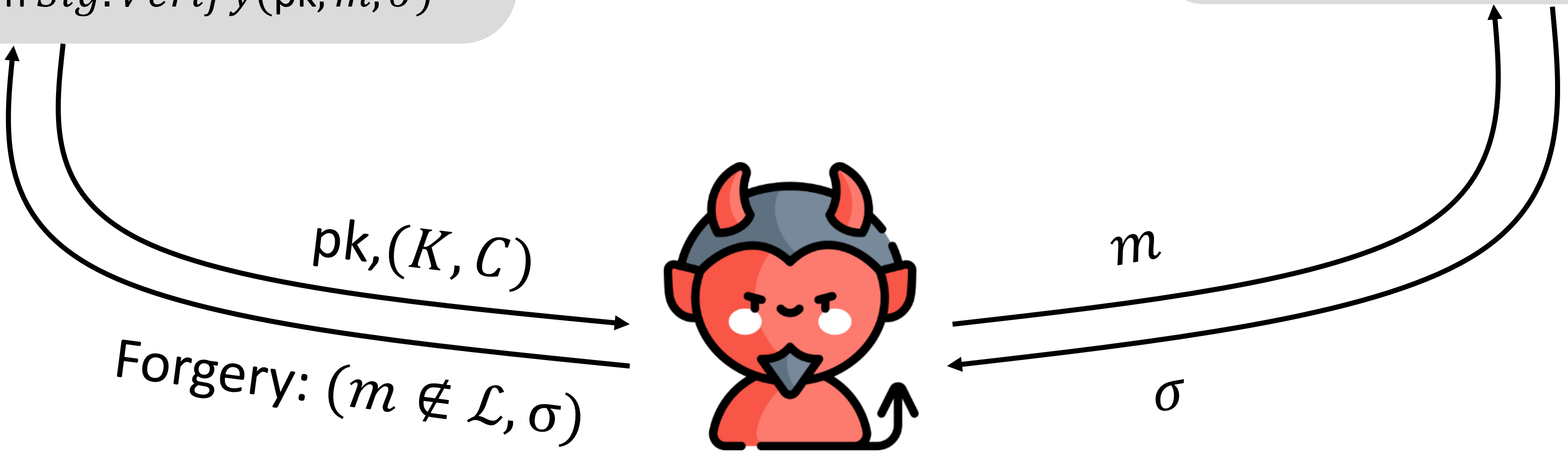
UF-CMA

1. pick a key pair (sk, pk) at random
2. init a table $\mathcal{L}$

Finalize: return $Sig.Verify(\text{pk}, m, \sigma)$

Oracle Sign

1. compute $\sigma = Sig.Sign(\text{sk}, m)$
2. update $\mathcal{L} = \mathcal{L} \cup m$



✅ Must not be able to come up with a valid forgery

What about GOST signature?

It was shown in [1] that generalized ElGamal signatures are secure in the UF-CMA notion in the Bijective Random Oracle (BRO) model under the DLP hardness assumption and two collision-resistance assumptions on a hash function.
This result also provides a security bound for the GOST signature.

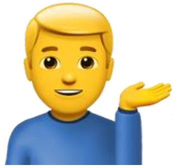
Theorem 1. Let $\mathcal{A}$ be an UF-CMA adversary for GOST scheme. Then there exists an adversary $\mathcal{D}_1$ and an adversary $\mathcal{D}_2$ that solve the DLP problem for $\mathbb{G}$, an adversaries $\mathcal{C}$ and $\mathcal{M}$ that break properties of H, such that:

$$Adv_{GOST}^{UFCMA}(\mathcal{A}) \le \sqrt{2N_\Pi^2 Adv_H(\mathcal{M}) + 2N_\Pi Adv_{\mathbb{G}}^{DLP}(\mathcal{D}_1) + Adv_{\mathbb{G}}^{DLP}(\mathcal{D}_2) + N_s Adv_H(\mathcal{C})} + \frac{N_\Pi^2}{2^l} + \frac{N_\Pi N_s}{2^l - N} + \frac{3N_s N}{q-1},$$

where $\mathcal{A}$ makes no more than N_s queries to its signature oracle, N_Π queries to BRO, $N = N_\Pi + N_s$, and $l = \lceil \log_2 q \rceil$.

[1] Fersch M. (2018). The provable security of Elgamal-type signature schemes.

Bijective Random Oracle



The bijective random oracle is an idealization of a conversion function $R \to R.x = r$ and r is in $\mathbb{Z}_q$. Such a conversion is intended to disrupt the algebraic structure of the cyclic group $\mathbb{G}$. An idealization for such a disruption is a random permutation.

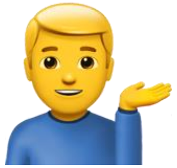
Signed Data (sk, data)

Basically the GOST signature:

1. pick k at random
2. compute $R = k \cdot P$
3. compute $r = R.x$
4. compute $e = H(data)$
5. compute $s = ke + dr$
6. output (r, s)

(r, s) →

Bijective Random Oracle



It was shown in [1] that ECDSA cannot be proved (under DLP-type assumptions) secure without the BRO.

[1] Hartmann D., Kiltz E. (2023). Limits in the Provable Security of ECDSA Signatures.

Signed Data (sk, data)

Basically the GOST signature:

1. pick k at random
2. compute $R = k \cdot P$
3. compute $r = R.x$
4. compute $e = H(data)$
5. compute $s = ke + dr$
6. output (r, s)

(r, s) →

What Is Joint Security?

We say that a KEM scheme and a signature scheme are jointly secure iff:

- The **KEM** scheme **remains secure** even if an adversary can **obtain signatures** of chosen messages signed with the secret key ➡️ IND-CCA-sig notion = IND-CCA + Signature oracle

- The **signature** scheme **remains secure** even if an adversary can **obtain decryption results** of chosen ciphertexts with a secret key ➡️ UF-CMA-dec notion = UF-CMA + Decapsulation Oracle

Is KEM Secure with a Signature Oracle?

The KEM in question in IND-CCA with a signature oracle is almost as secure as in conventional IND-CCA.

Theorem 1. Let $\mathcal{A}$ be an IND-CCA-sig adversary for KEM and GOST in the bijective random oracle model. Then there exist an IND-CCA adversary $\mathcal{B}$ for KEM, such that

$$Adv_{KEM,GOST}^{INDCCA-sig}(\mathcal{A}) \leq Adv_{KEM}^{INDCCA}(\mathcal{B}) + \frac{3N_s(N_s + N_\Pi)}{q-1},$$

where $\mathcal{A}$ makes no more than N_s queries to its signing oracle and no more than N_Π queries to BRO. $\mathcal{B}$ makes the same number of queries to the Dec oracle as $\mathcal{A}$.

Is KEM Secure with a Signature Oracle?

The KEM in question in IND-CCA with a signature oracle is almost as secure as in conventional IND-CCA.

Theorem 1. Let $\mathcal{A}$ be an IND-CCA-sig adversary for KEM and GOST in the bijective random oracle model. Then there exist an IND-CCA adversary $\mathcal{B}$ for KEM, such that

$$Adv_{KEM,GOST}^{INDCCA-sig}(\mathcal{A}) \leq Adv_{KEM}^{INDCCA}(\mathcal{B}) + \frac{3N_s(N_s + N_\Pi)}{q-1},$$

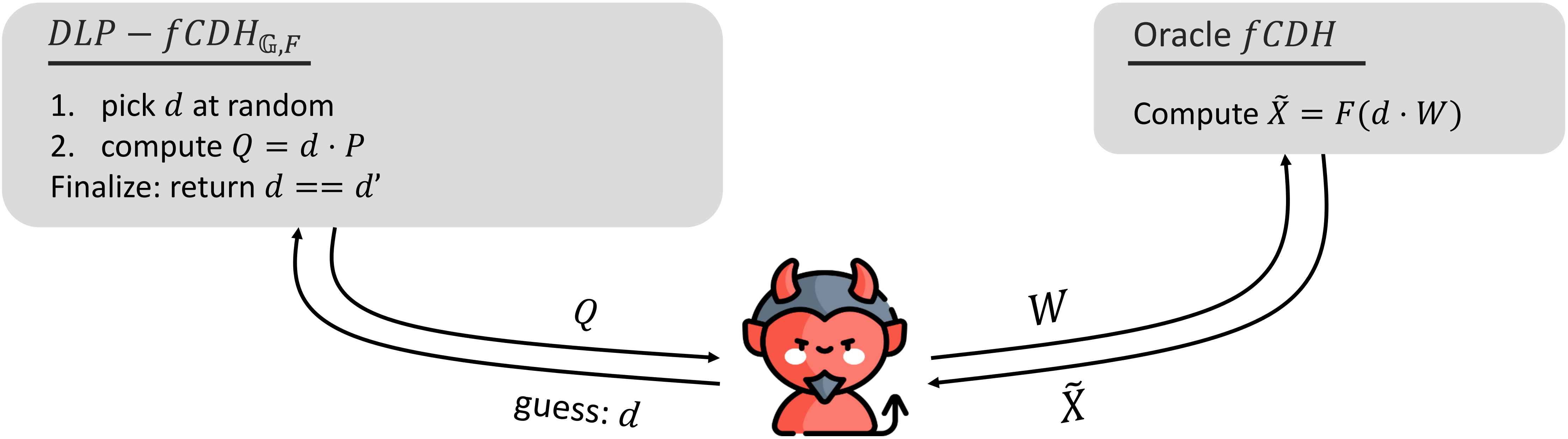
where $\mathcal{A}$ makes no more than N_s queries to its signing oracle and no more than N_Π queries to BRO. $\mathcal{B}$ makes the same number of queries to the Dec oracle as $\mathcal{A}$.

The proof is in the bijective random oracle model!
BRO is necessary to simulate the signature oracle answers, just like in the UF-CMA security proof of GOST.

Before We Go Further – DLP-fCDH Assumption

TL;DR

- A modification of the discrete logarithm problem similar to ODH
- The DLP-fCDH problem is harder than ODH problem, as proved in the paper



✅ Must not be able to find the secret d (i.e., solve DLP) even if it can obtain the results of F applied to DH values of the secret key and chosen points

Before We Go Further – DLP-fCDH Assumption

TL;DR

- A modification of the discrete logarithm problem similar to ODH
- The DLP-fCDH problem is harder than ODH problem, as proved in the paper

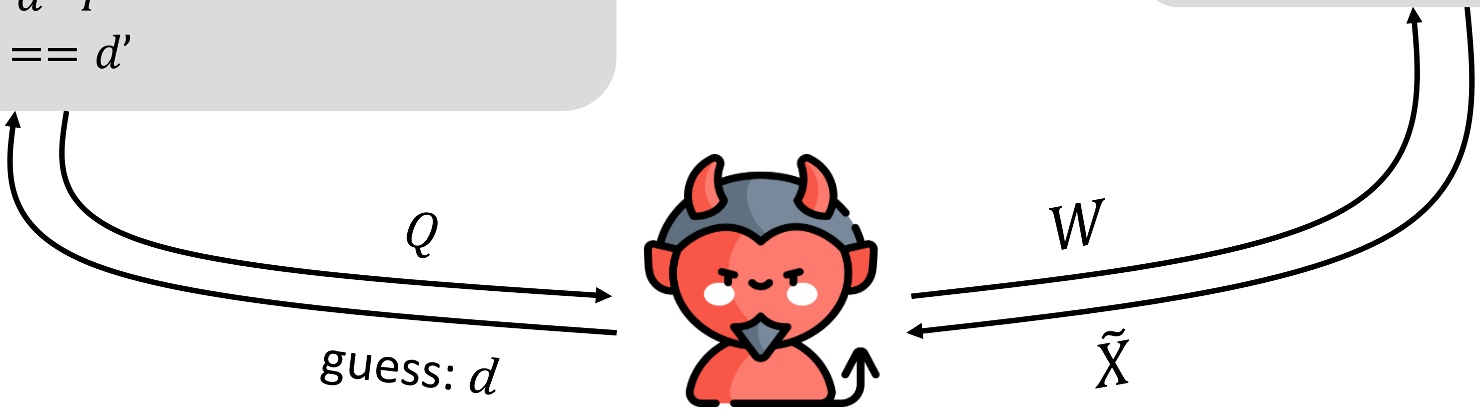
$DLP - fCDH_{\mathbb{G},F}$

1. pick d at random
2. compute $Q = d \cdot P$

Finalize: return $d == d'$

Oracle $fCDH$

Compute $\tilde{X} = F(d \cdot W)$



Must not be able to find the secret d (i.e., solve DLP) even if it can obtain the results of F applied to DH values of the secret key and chosen points

Is the GOST Scheme Secure with a Decapsulation Oracle?

For the GOST signature in UF-CMA in the presence of a decapsulation oracle, a bound similar to the conventional UF-CMA bound can be obtained.

Theorem 1. Let $\mathcal{A}$ be an UF-CMA adversary for GOST scheme. Then there exists an adversary $\mathcal{D}_1$ and an adversary $\mathcal{D}_2$ that solve the DLP-fCDH problem for $\mathbb{G}$, an adversaries $\mathcal{C}$ and $\mathcal{M}$ that break properties of H, such that:

$$Adv_{GOST}^{UFCMA}(\mathcal{A}) \leq \sqrt{2N_\Pi^2 Adv_H(\mathcal{M}) + 2N_\Pi Adv_{\mathbb{G}}^{DLP-fCDH}(\mathcal{D}_1)} + Adv_{\mathbb{G}}^{DLP-fCDH}(\mathcal{D}_2) + N_s Adv_H(\mathcal{C}) + \frac{N_\Pi^2}{2^l} + \frac{N_\Pi N_s}{2^l - N} + \frac{3N_s N}{q-1},$$

where $\mathcal{A}$ makes no more than N_s queries to its signature oracle, N_Π queries to BRO, $N = N_\Pi + N_s$, and $l = \lceil \log_2 q \rceil$.

- The DLP hardness assumption is replaced with the DLP-fCDH assumption, which is used to simulate the decapsulation oracle.
- The proof of the theorem requires surgical work with the original UF-CMA proof.

So, What Do We Have?

➡️ The theorems obtained demonstrate that the KEM and signature schemes in GOST CMS are jointly secure.

➡️ However, the bounds do degrade. Specifically, for the signature scheme, a different assumption is required – DLP-fCDH instead of the conventional DLP.

So, What Do We Have?

➡️ The theorems obtained demonstrate that the KEM and signature schemes in GOST CMS are jointly secure.

➡️ However, the bounds do degrade. Specifically, for the signature scheme, a different assumption is required – DLP-fCDH instead of the conventional DLP.

❓ The only remaining question is:

What does that degradation indicate?

Let's Compare

- Consider "joint" security of KEM and signature when 2 independent keys are used:

$$Adv^{2keys} = Adv_{CMS}^{INDCCA}(\mathcal{A}) + Adv_{GOST}^{UFCMA}(\mathcal{A}) \leq Adv_{AE}^{AECCA}(\mathcal{B}) + Adv_{\mathbb{G},F}^{ODH} + \frac{2N_d}{q-1} + \sqrt{2N_\Pi^2 Adv_H(\mathcal{M}) + 2N_\Pi Adv_{\mathbb{G}}^{DLP}(\mathcal{D}_1)} + Adv_{\mathbb{G}}^{DLP}(\mathcal{D}_2) + N_s Adv_H(\mathcal{C}) + \frac{N_\Pi^2}{2^l} + \frac{N_\Pi N_s}{2^l - N} + \frac{3N_s N}{q-1},$$

- Consider joint security of KEM and signature when a single key is used:

$$Adv^{1keys} = Adv_{CMS}^{INDCCA-sig}(\mathcal{A}) + Adv_{GOST}^{UFCMA-dec}(\mathcal{A}) \leq Adv_{AE}^{AECCA}(\mathcal{B}) + Adv_{\mathbb{G},F}^{ODH} + \frac{2N_d}{q-1} + \sqrt{2N_\Pi^2 Adv_H(\mathcal{M}) + 2N_\Pi Adv_{\mathbb{G}}^{DLPfCDH}(\mathcal{D}_1)} + Adv_{\mathbb{G}}^{DLPfCDH}(\mathcal{D}_2) + N_s Adv_H(\mathcal{C}) + \frac{N_\Pi^2}{2^l} + \frac{N_\Pi N_s}{2^l - N} + \frac{6N_s N}{q-1},$$

Let's Compare

- Consider "joint" security of KEM and signature when 2 independent keys are used:

$$Adv^{2keys} = Adv_{CMS}^{INDCCA}(\mathcal{A}) + Adv_{GOST}^{UFCMA}(\mathcal{A}) \leq Adv_{AE}^{AECCA}(\mathcal{B}) + Adv_{\mathbb{G},F}^{ODH} + \frac{2N_d}{q-1} + \sqrt{2N_\Pi^2 Adv_H(\mathcal{M}) + 2N_\Pi} \left(Adv_{\mathbb{G}}^{DLP}(\mathcal{D}_1) + Adv_{\mathbb{G}}^{DLP}(\mathcal{D}_2) \right) + N_s Adv_H(\mathcal{C}) + \frac{N_\Pi^2}{2^l} + \frac{N_\Pi N_s}{2^l - N} + \frac{3N_s N}{q-1},$$

- Consider joint security of KEM and signature when a single key is used:

$$Adv^{1keys} = Adv_{CMS}^{INDCCA-sig}(\mathcal{A}) + Adv_{GOST}^{UFCMA-dec}(\mathcal{A}) \leq Adv_{AE}^{AECCA}(\mathcal{B}) + Adv_{\mathbb{G},F}^{ODH} + \frac{2N_d}{q-1} + \sqrt{2N_\Pi^2 Adv_H(\mathcal{M}) + 2N_\Pi} \left(Adv_{\mathbb{G}}^{DLPfCDH}(\mathcal{D}_1) + Adv_{\mathbb{G}}^{DLPfCDH}(\mathcal{D}_2) \right) + N_s Adv_H(\mathcal{C}) + \frac{N_\Pi^2}{2^l} + \frac{N_\Pi N_s}{2^l - N} + \frac{6N_s N}{q-1},$$

Let's Compare

- Consider "joint" security of KEM and signature when 2 independent keys are used:

$$Adv^{2keys} = Adv_{CMS}^{INDCCA}(\mathcal{A}) + Adv_{GOST}^{UFCMA}(\mathcal{A}) \le Adv_{AE}^{AECCA}(\mathcal{B}) + Adv_{\mathbb{G},F}^{ODH} + \frac{2N_d}{q-1} + \sqrt{2N_\Pi^2 Adv_H(\mathcal{M}) + 2N_\Pi Adv_{\mathbb{G}}^{DLP}(\mathcal{D}_1) + Adv_{\mathbb{G}}^{DLP}(\mathcal{D}_2)} + N_s Adv_H(\mathcal{C}) + \frac{N_\Pi^2}{2^l} + \frac{N_\Pi N_s}{2^l - N} + \frac{3N_s N}{q-1},$$

- Consider joint security of KEM and signature when a single key is used:

$$Adv^{1keys} = Adv_{CMS}^{INDCCA-sig}(\mathcal{A}) + Adv_{GOST}^{UFCMA-dec}(\mathcal{A}) \le Adv_{AE}^{AECCA}(\mathcal{B}) + Adv_{\mathbb{G},F}^{ODH} + \frac{2N_d}{q-1} + \sqrt{2N_\Pi^2 Adv_H(\mathcal{M}) + 2N_\Pi Adv_{\mathbb{G}}^{DLPfCDH}(\mathcal{D}_1) + Adv_{\mathbb{G}}^{DLPfCDH}(\mathcal{D}_2)} + N_s Adv_H(\mathcal{C}) + \frac{N_\Pi^2}{2^l} + \frac{N_\Pi N_s}{2^l - N} + \frac{6N_s N}{q-1},$$

Is harder than ODH. And we already have ODH at home!

Conclusion

- The theorems obtained demonstrate that the KEM and signature schemes in GOST CMS are jointly secure.

- However, the bounds do degrade. Specifically, for the signature scheme, a different assumption is required – DLP-fCDH instead of the conventional DLP.

- Nevertheless, the obtained bound suggests an absence of new classes of attacks arising from the use of the same key.

We Did a Little Bit More

- The KEM in GOST CMS might utilize a randomization value, UKM, in Diffie-Hellman. Such a KEM requires a different security notion and corresponding modifications in the proof. We have addressed that case as well.
- We have demonstrated joint security not only for GOST but also for generalized ElGamal.

Questions?

Contacts:

bozhko@cryptopro.ru