Leakage-Resilient Chosen-Ciphertext Secure Public-Key Encryption from Hash Proof System and One-Time Lossy Filter

Baodong Qin and Shengli Liu

Shanghai Jiao Tong University

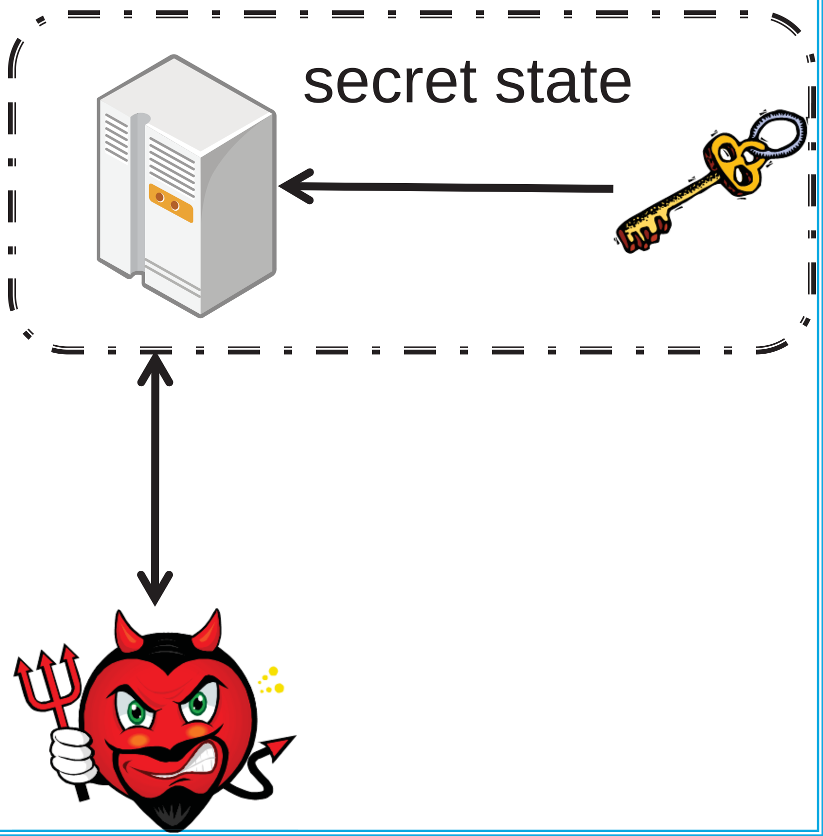
ASIACRYPT 2013

Dec 5, Bangalore, India

Why We Consider Secrets Leak?

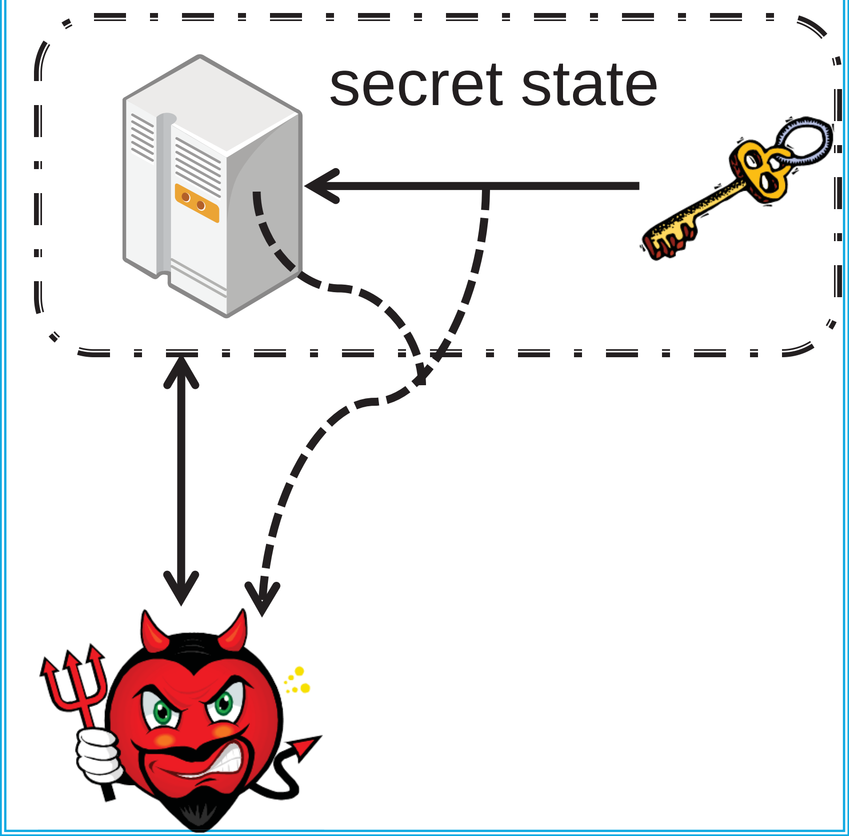
THEORY

- Ideal setting
- Private internal secret state



REAL LIFE

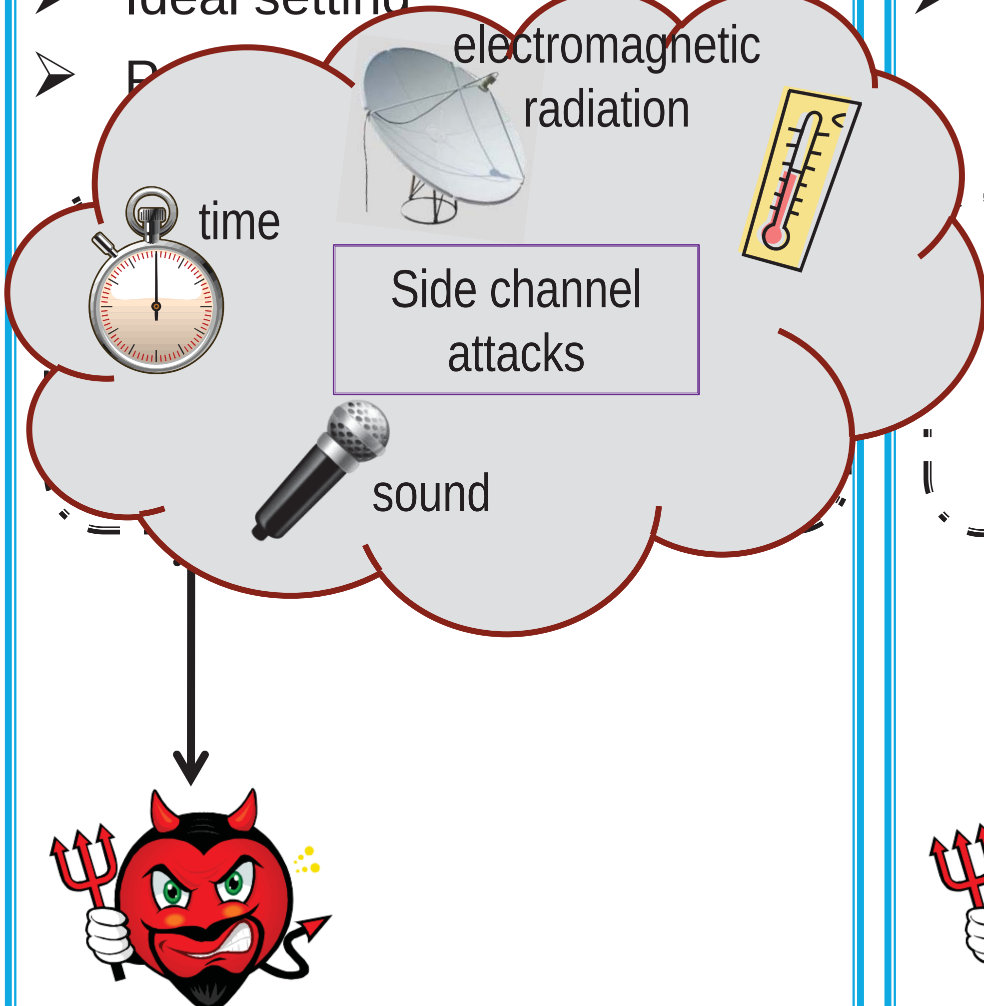
- Physical implementation leaks information
- e.g.: secret key/ randomness



Why We Consider Secrets Leak?

THEORY

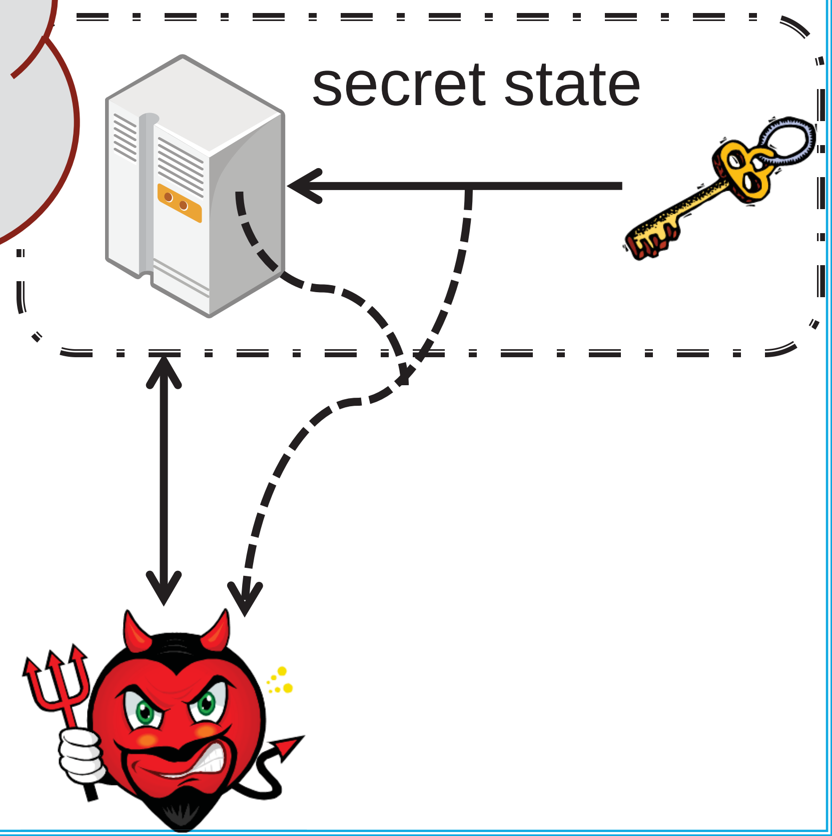
- Ideal setting



REAL LIFE

- Physical implementation leaks information

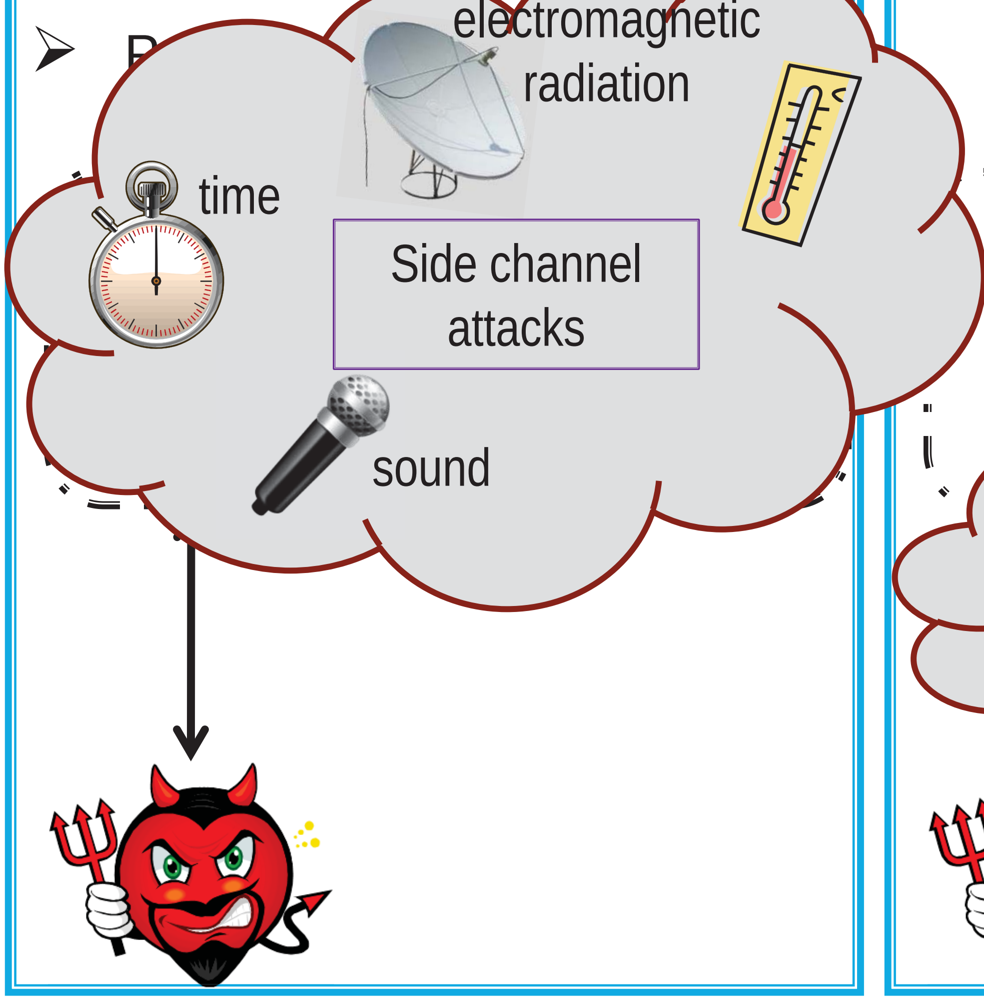
  e.g.: secret key/ randomness



Why We Consider Secrets Leak?

THEORY

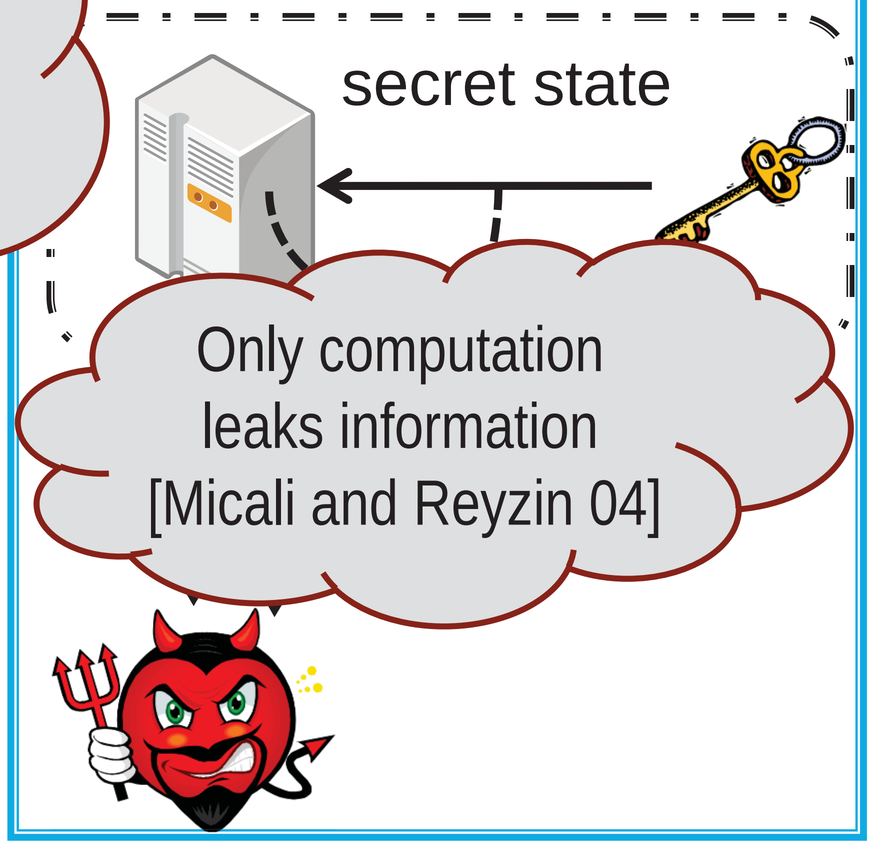
- Ideal setting



REAL LIFE

- Physical implementation leaks information

  e.g.: secret key/ randomness



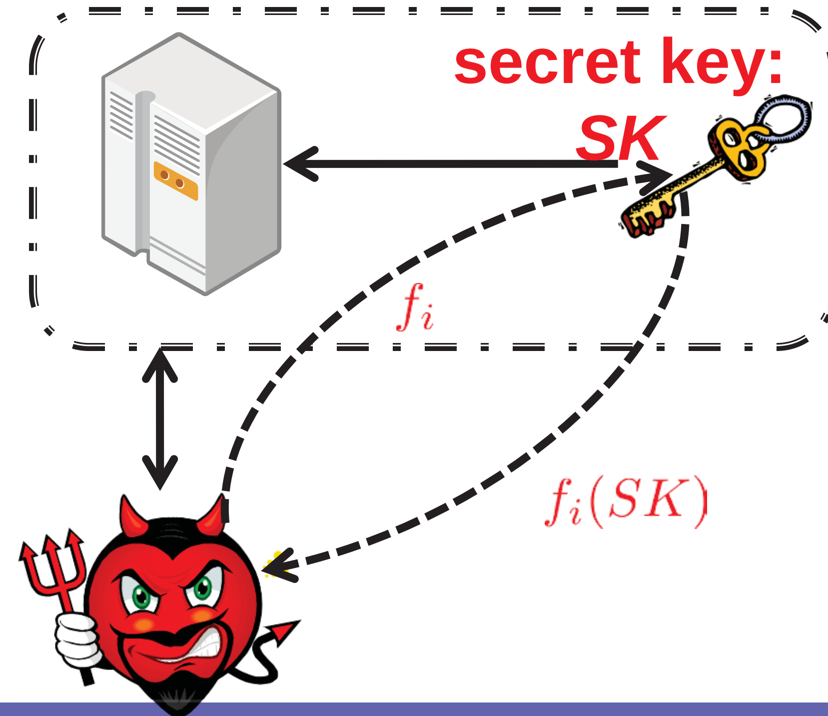
Bounded Leakage Model

- Inspired by "cold-boot" attack/memory attack [Halderman et al.08]
  - Not only computation leaks information

- Model: leakage oracle

$\mathcal{O}_{SK}^{\lambda,\kappa}$:

- $f_i : \{0,1\}^* \rightarrow \{0,1\}^{\lambda_i}$
- $\sum_i \lambda_i \leq \lambda$
- Leakage rate: $\lambda/|SK|$



Public-Key Encryption

Semantic security against key leakage and CCA [NS09]

$$\mathsf{PKE.Enc}(PK, M_0) \overset{c}{\approx} \mathsf{PKE.Enc}(PK, M_1)$$

Adversary

$(PK, SK) \leftarrow \mathsf{PKE.Gen}(1^\kappa)$

PK

CT

$M \leftarrow \mathsf{PKE.Dec}(SK, CT)$

Decryption queries

f

$f(SK)$

Leakage queries

M_0, M_1

$b \leftarrow \{0, 1\}$

CT^*

$CT^* \leftarrow \mathsf{PKE.Enc}(PK, M_b)$

$CT \neq CT^*$

$M \leftarrow \mathsf{PKE.Dec}(SK, CT)$

output b'

Public-Key Encryption

Semantic security against key leakage and CCA [NS09]

$$\mathsf{PKE.Enc}(PK, M_0) \overset{c}{\approx} \mathsf{PKE.Enc}(PK, M_1)$$

Adversary

Decryption queries

Leakage queries

PK

$(PK, SK) \leftarrow \mathsf{PKE.Gen}(1^\kappa)$

CT

$M \leftarrow \mathsf{PKE.Dec}(SK, CT)$

The adversary succeeds if b=b'

Advantage: Pr[b=b']-1/2

$b \leftarrow \{0,1\}$

$CT^* \leftarrow \mathsf{PKE.Enc}(PK, M_b)$

$CT \neq CT^*$

$M \leftarrow \mathsf{PKE.Dec}(SK, CT)$

output b'

Previous Works

- High leakage-rate (e.g. 1-o(1), using NIZK) but
  - either no efficient instantiations [NS09] or
  - over a pairing-friendly group (efficient, but the ciphertext size is a little bit large) [Dodis et al.10, Galindo et al.12]

Previous Works

- High leakage-rate (e.g. 1-o(1), using NIZK) but
  - either no efficient instantiations [NS09] or
  - over a pairing-friendly group (efficient, but the ciphertext size is a little bit large) [Dodis et al.10, Galindo et al.12]
- Low leakage rate (e.g. 1/4-o(1)), but
  - very practical construction via hash proof system [NS09,Li et al.12, Liu et al.13]
  - has short ciphertext size (for reasonable leakage rate)
  - Instantiations under DDH, DCR etc. (without pairing)

Question

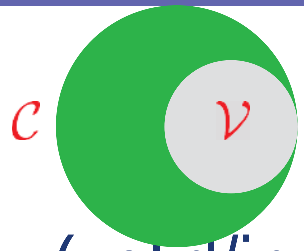
From [Dodis et al. Asiacrypt 2010]

…, it seems that the hash proof system approach to building CCA encryption is inherently limited to leakage-rates below 1/2: this is because the secret-key consists of two components (one for verifying that the ciphertext is well-formed and one for decrypting it) and the proofs break down if either of the components is individually leaked in its entirety.

However, no HPS-based PKEs are known achieving leakage-rate 1/2-o(1), especially under DDH or DCR assumptions.

Question: can we find a new way to construct LR-CCA secure PKEs which are as practical as HPS with reasonable high leakage-rates, like 1/2-o(1)?

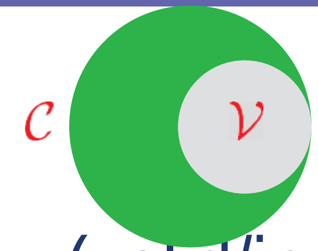
Hash Proof System[CS02]



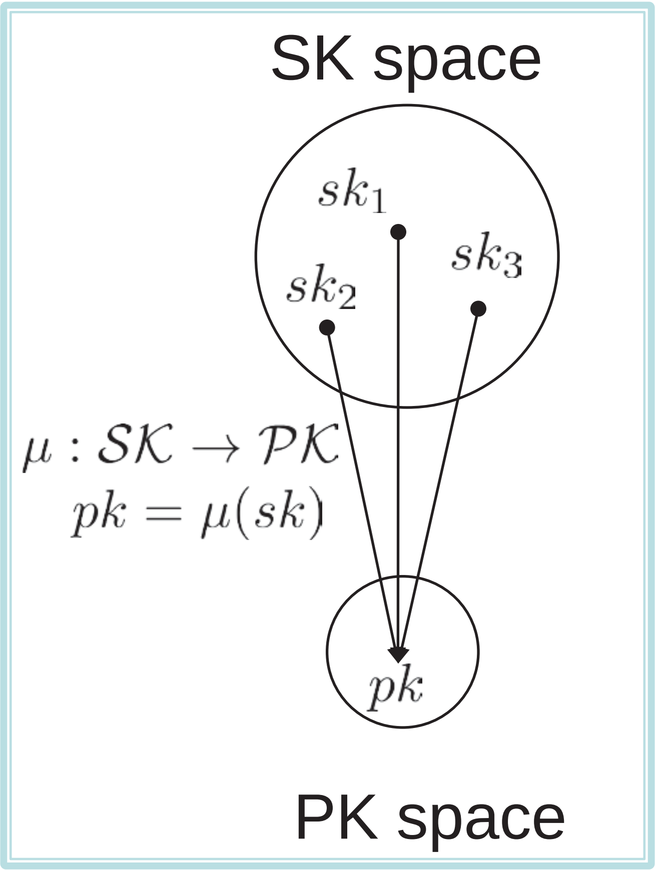
- Family of projective hash functions
- Subset membership problem: $\mathcal{V} \overset{c}{\approx} \mathcal{C} \setminus \mathcal{V}$ (valid/invalid)

Hash Proof System[CS02]

- Family of projective hash functions

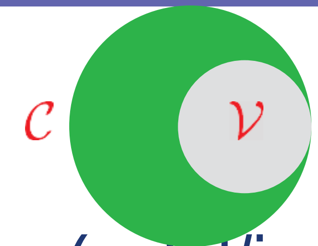


- Subset membership problem: $\mathcal{V} \overset{c}{\approx} \mathcal{C} \setminus \mathcal{V}$ (valid/invalid)

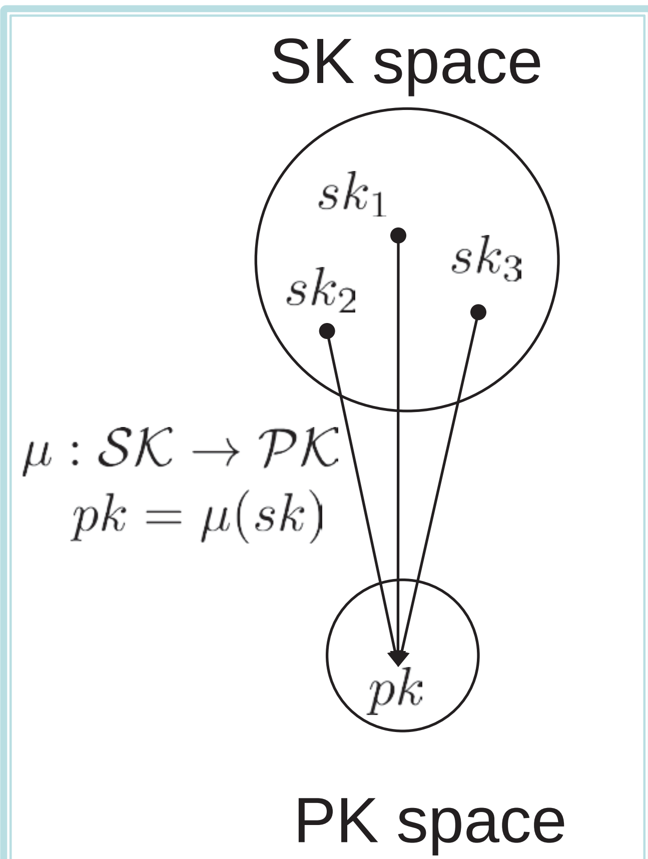


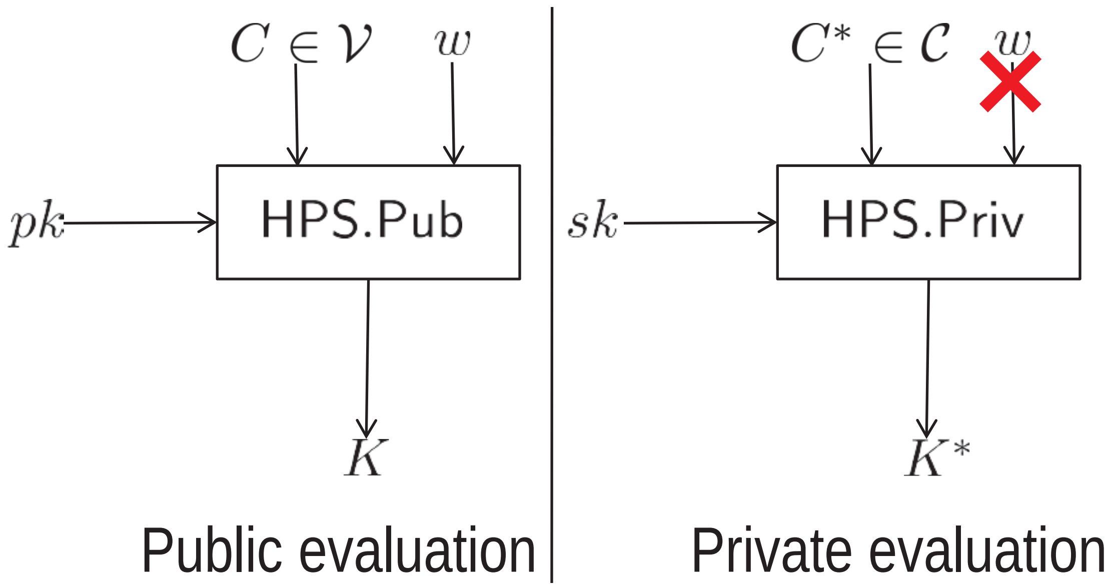
Hash Proof System[CS02]

- Family of projective hash functions



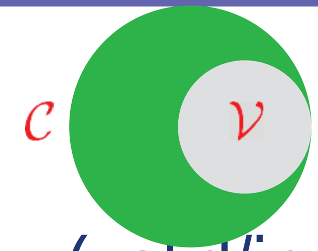
- Subset membership problem: $\mathcal{V} \stackrel{c}{\approx} \mathcal{C} \setminus \mathcal{V}$ (valid/invalid)



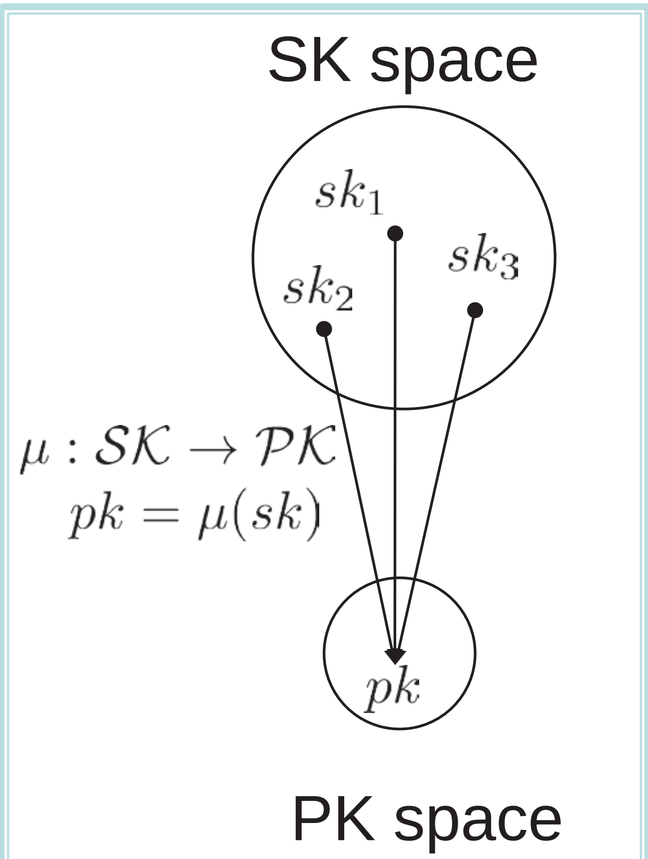


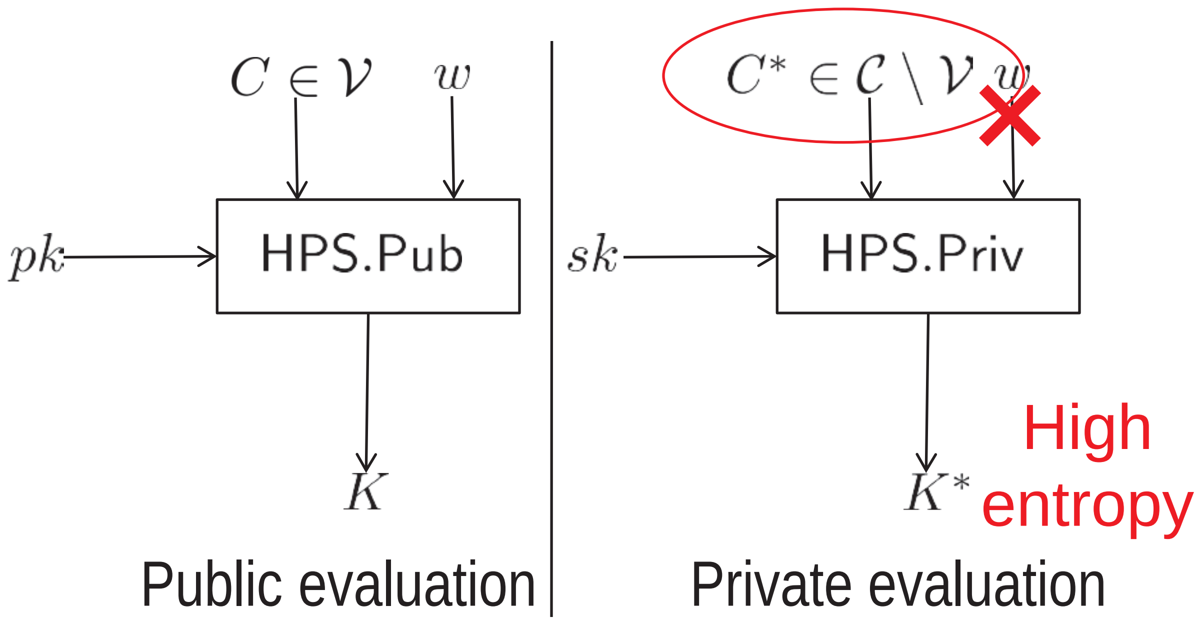
Hash Proof System[CS02]

- Family of projective hash functions



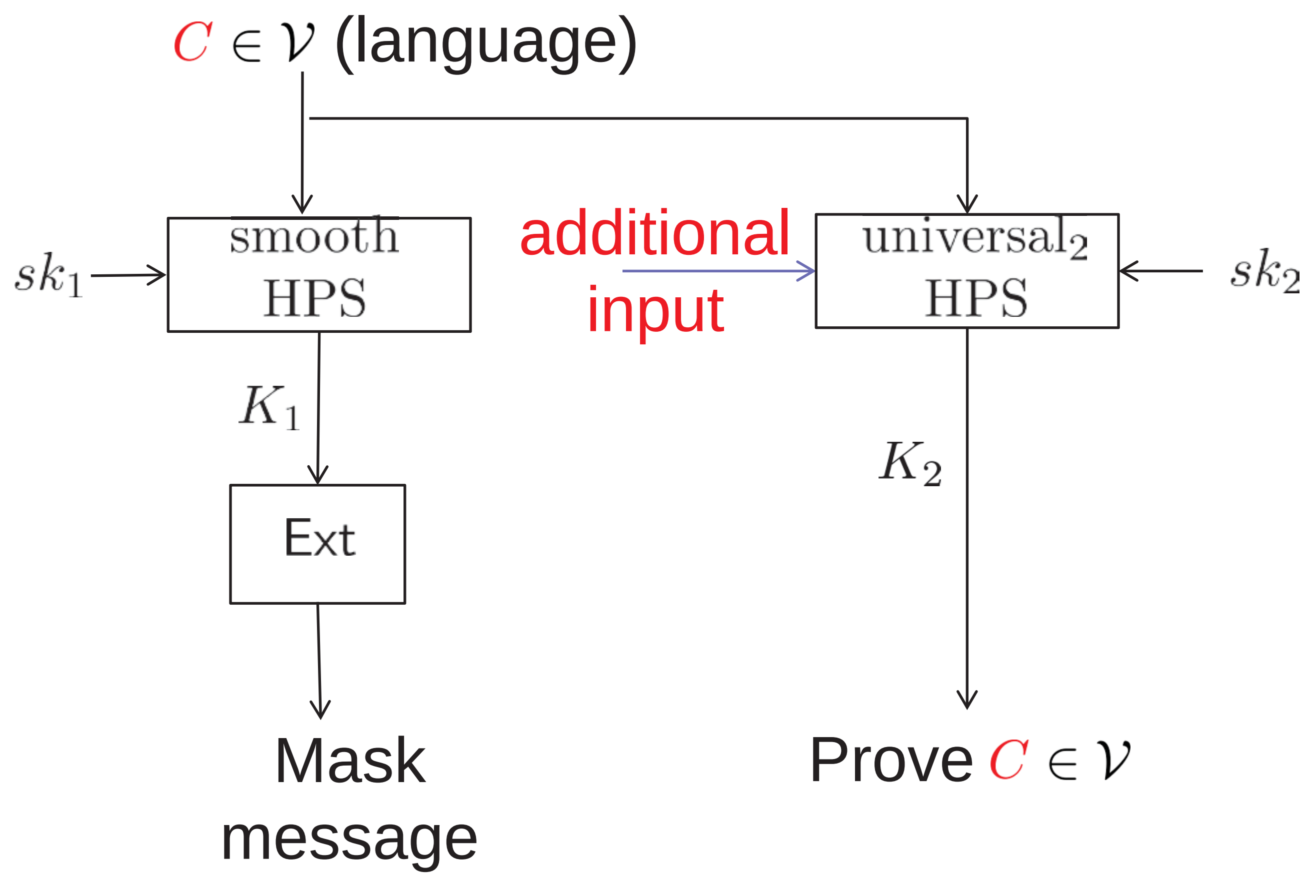
- Subset membership problem: $\mathcal{V} \stackrel{c}{\approx} \mathcal{C} \setminus \mathcal{V}$ (valid/invalid)



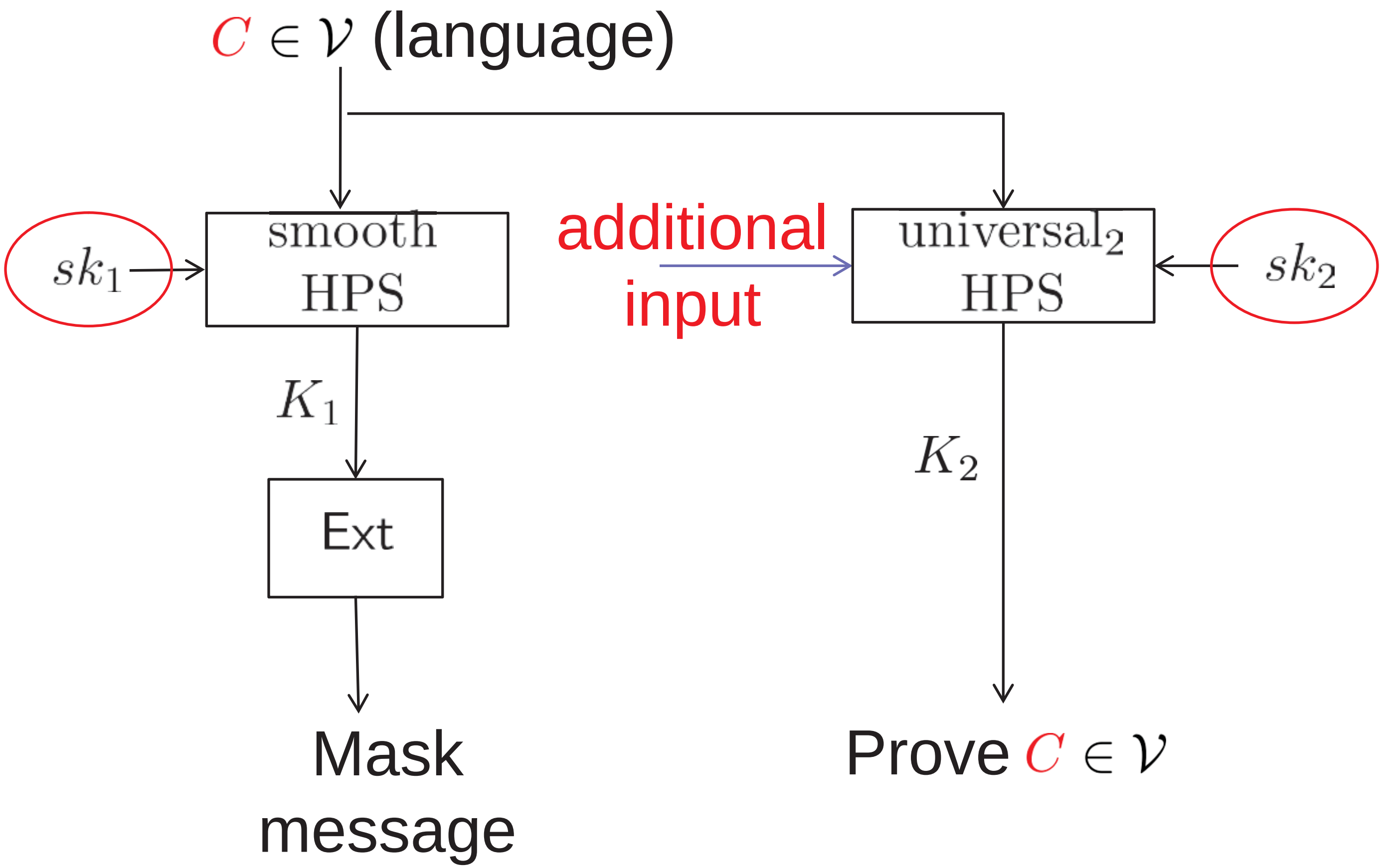


- universal/universal$_2$
- smooth

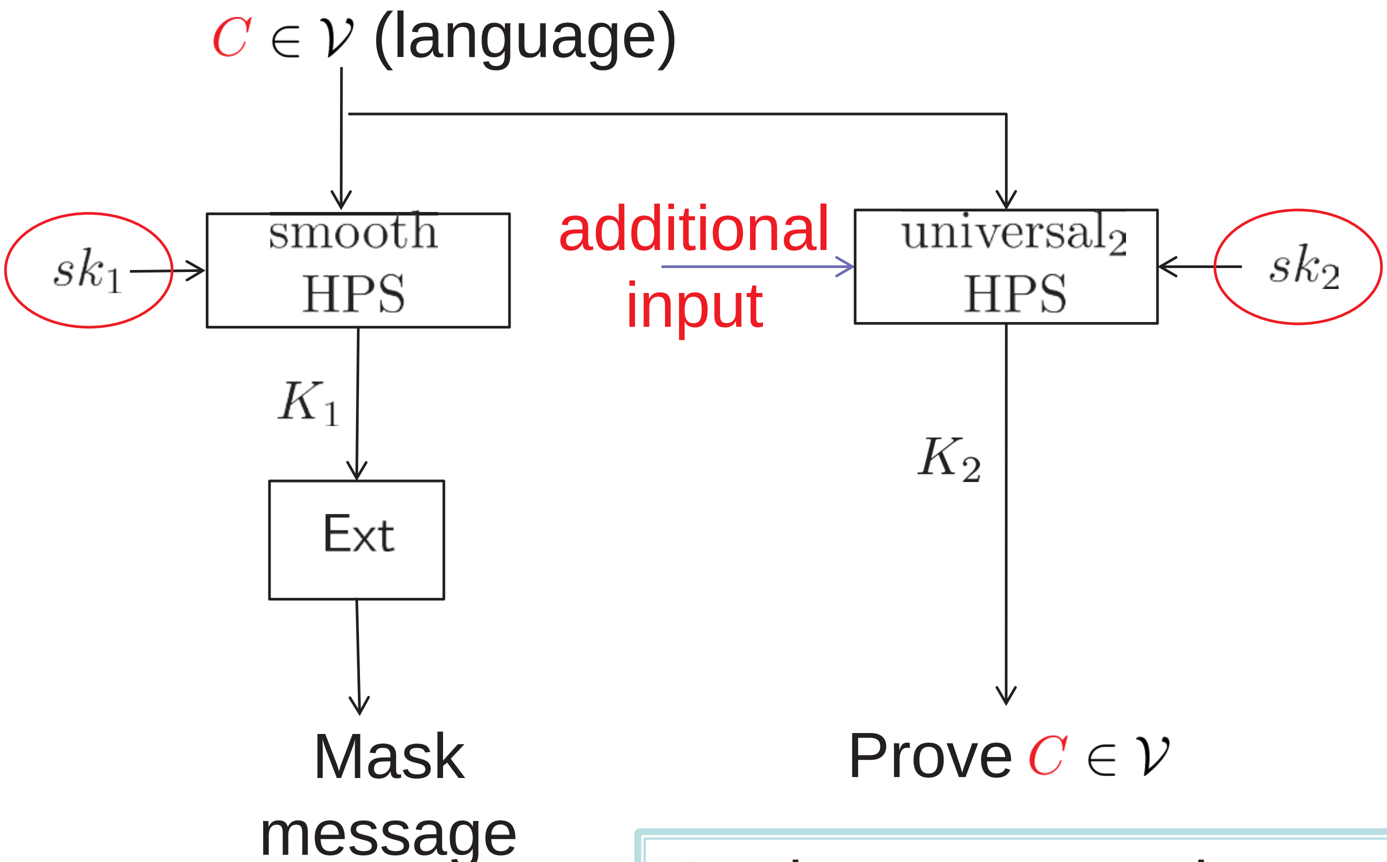
HPS-based Approach



HPS-based Approach

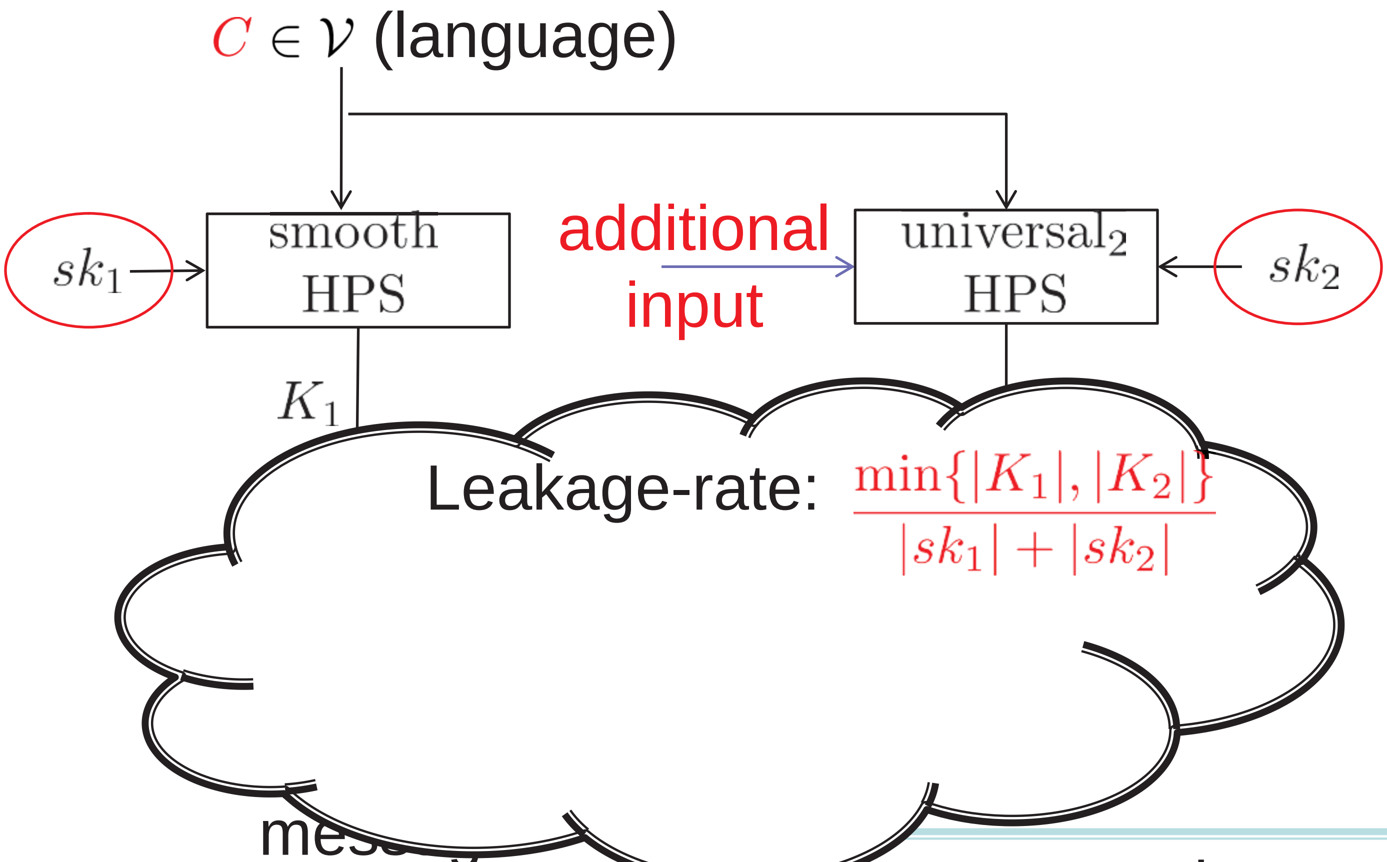


HPS-based Approach



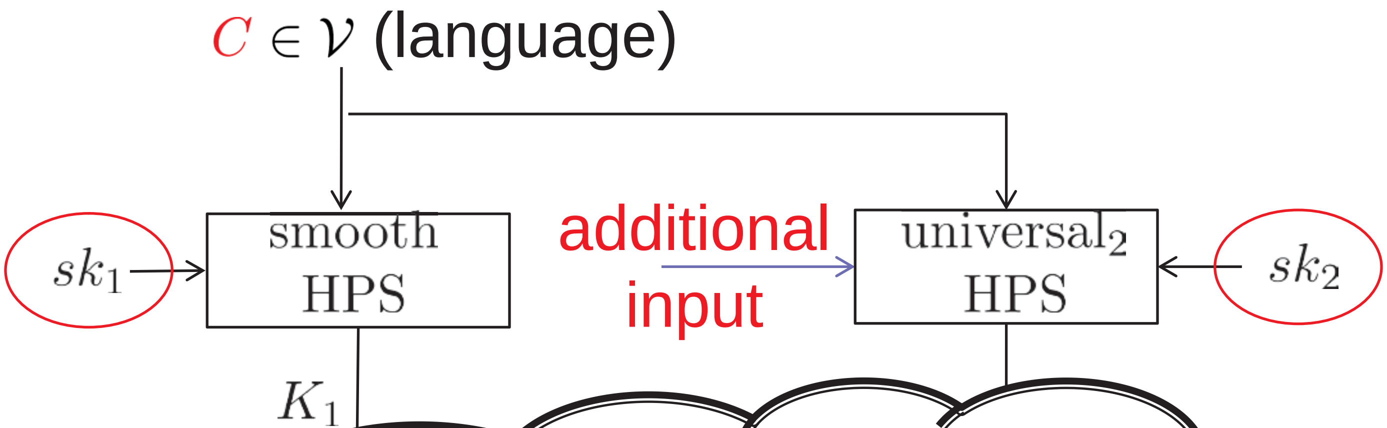
- Leakage amount is at most: $\min\{|sk_1|, |sk_2|\}$
- In fact smaller than $\min\{|K_1|, |K_2|\}$

HPS-based Approach



- Leakage amount is at most: $\min\{|sk_1|, |sk_2|\}$
- In fact smaller than $\min\{|K_1|, |K_2|\}$

HPS-based Approach



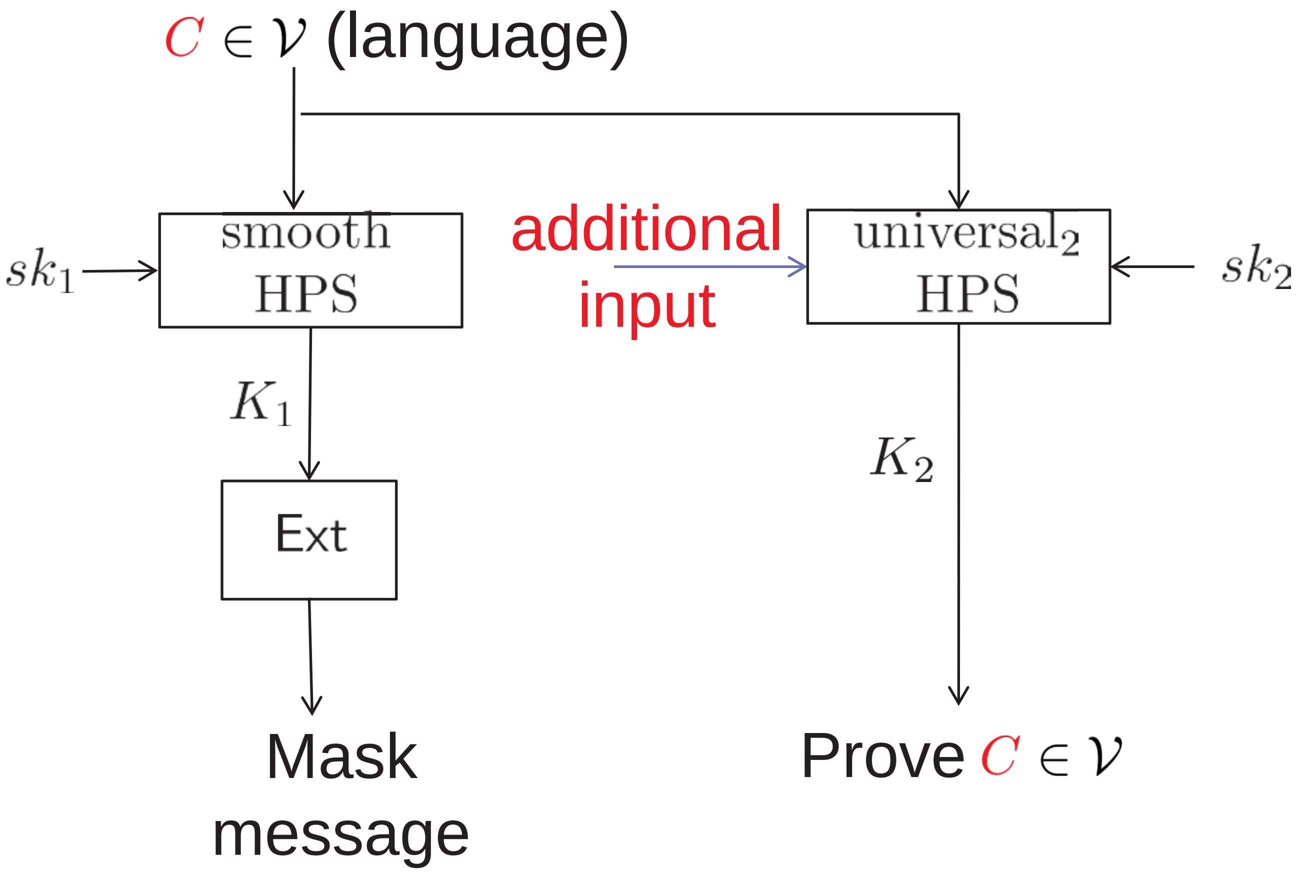
Leakage-rate: $\frac{\min\{|K_1|, |K_2|\}}{|sk_1| + |sk_2|}$

Best result: 1/4 –o(1)
under DDH assumption

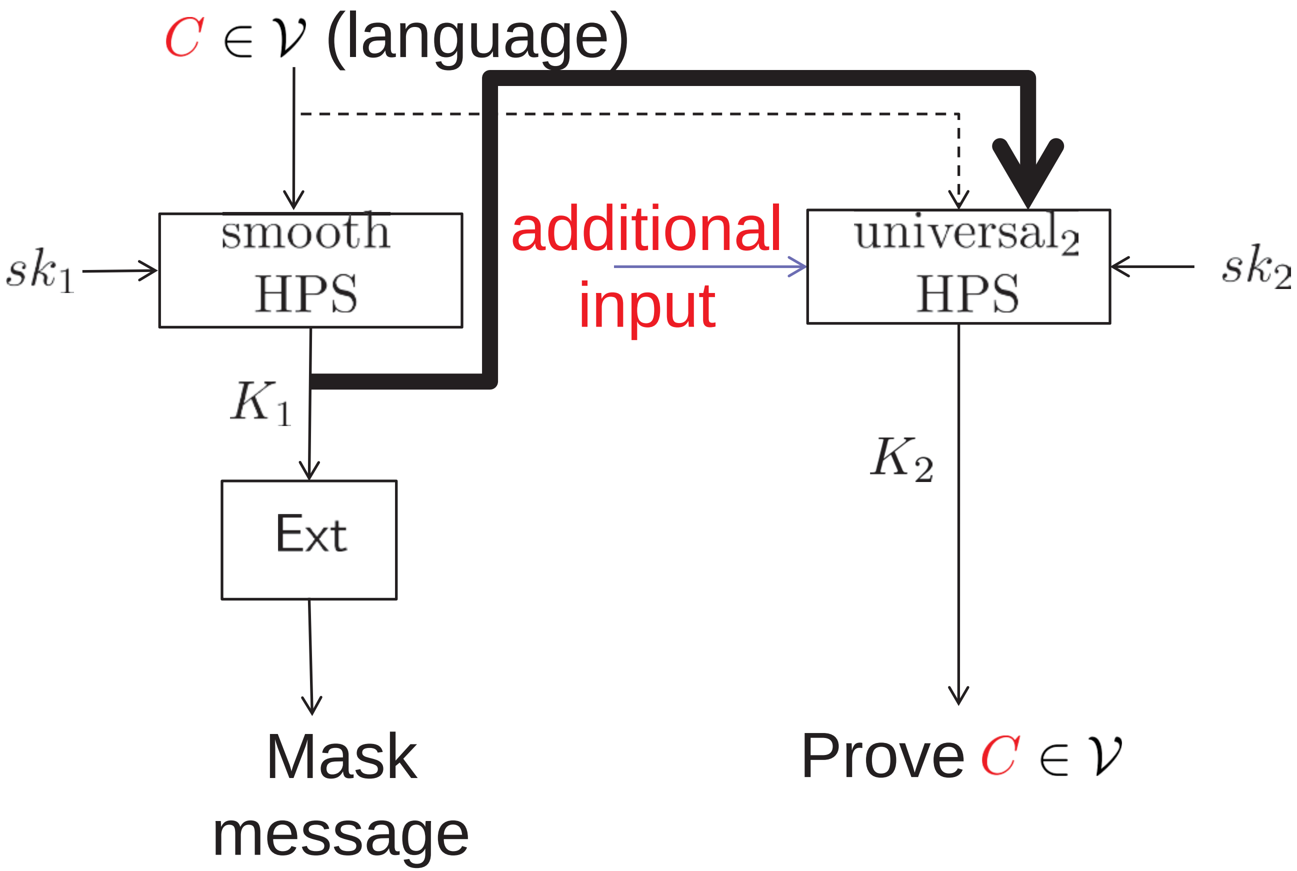
mes

- Leakage amount is at most: $\min\{|sk_1|, |sk_2|\}$
- In fact smaller than $\min\{|K_1|, |K_2|\}$

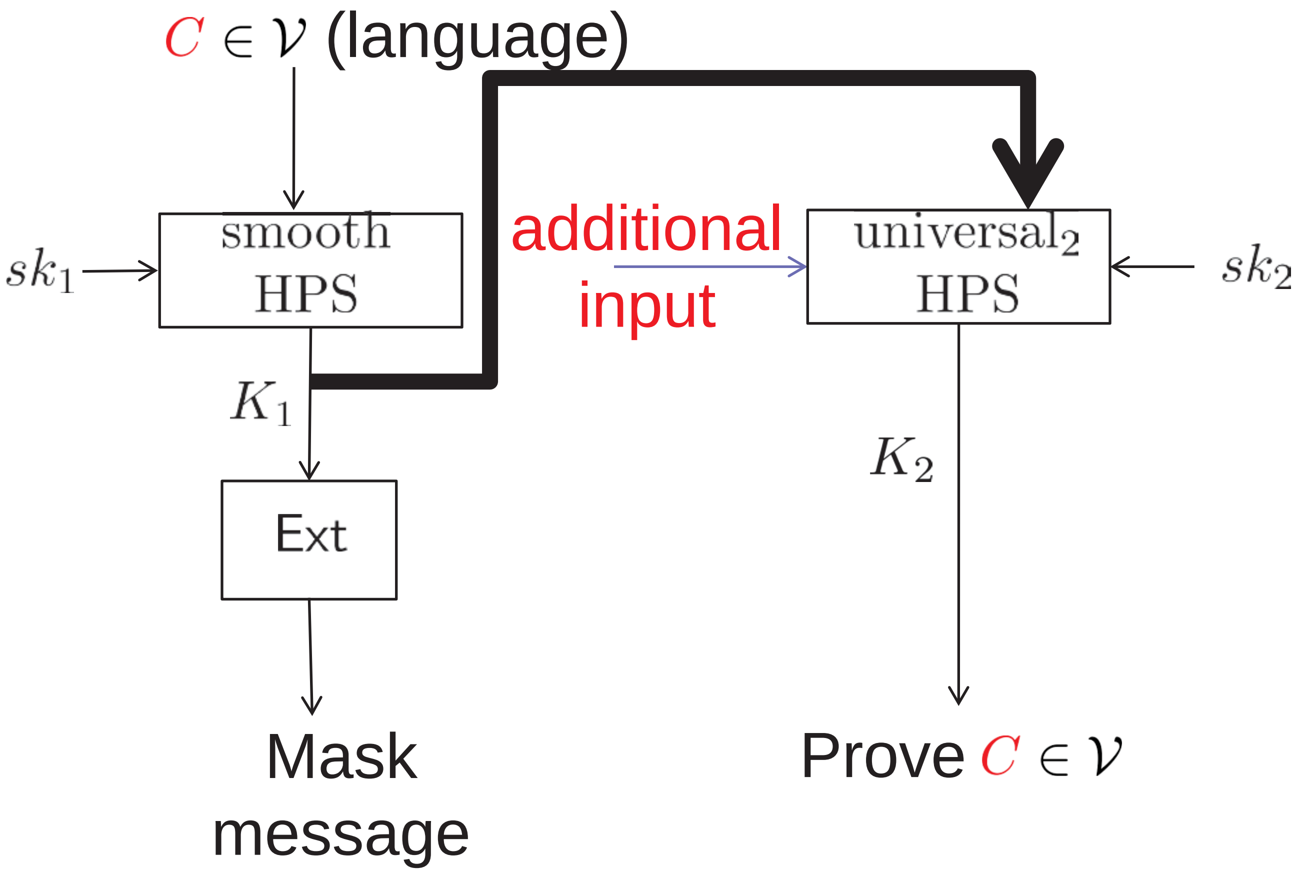
Our Approach



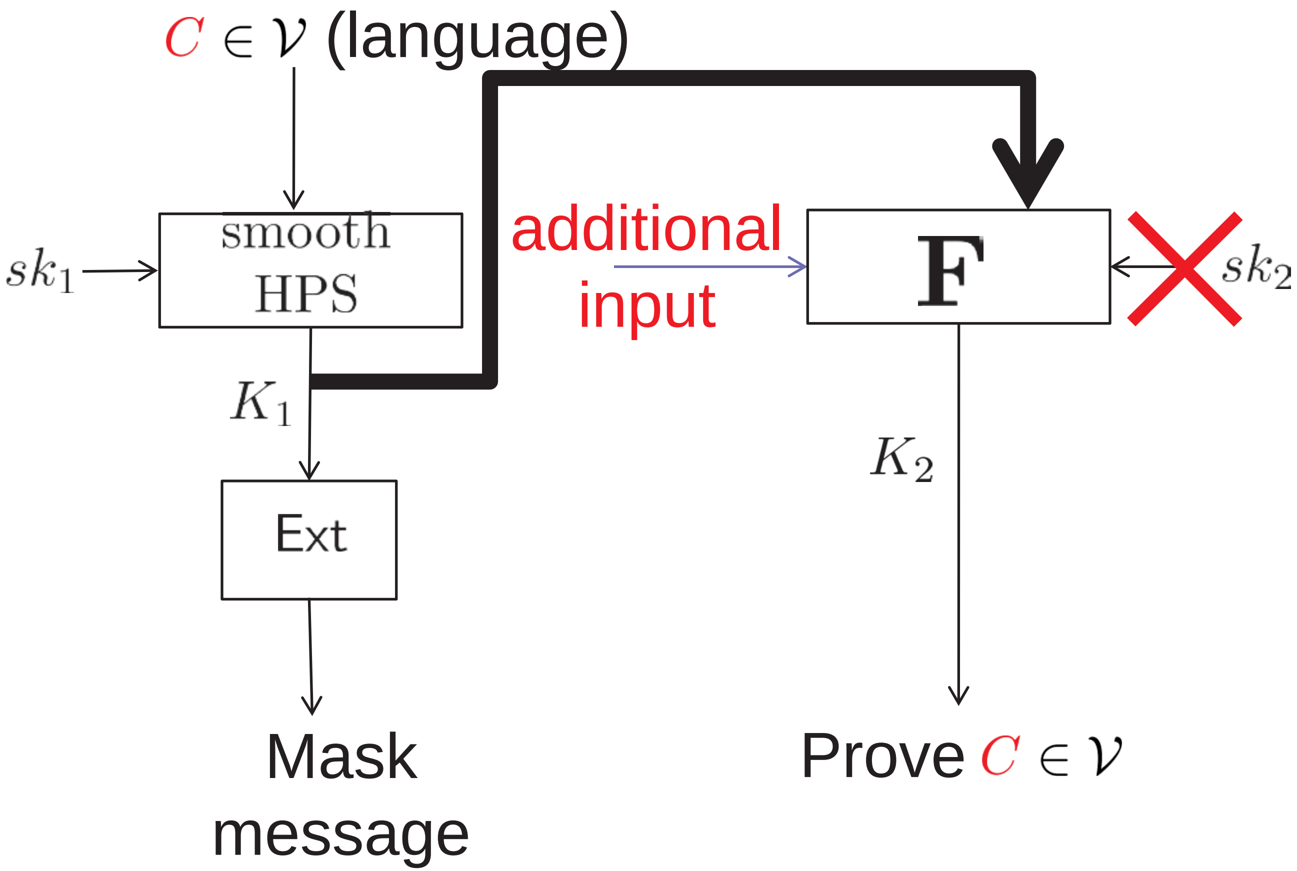
Our Approach



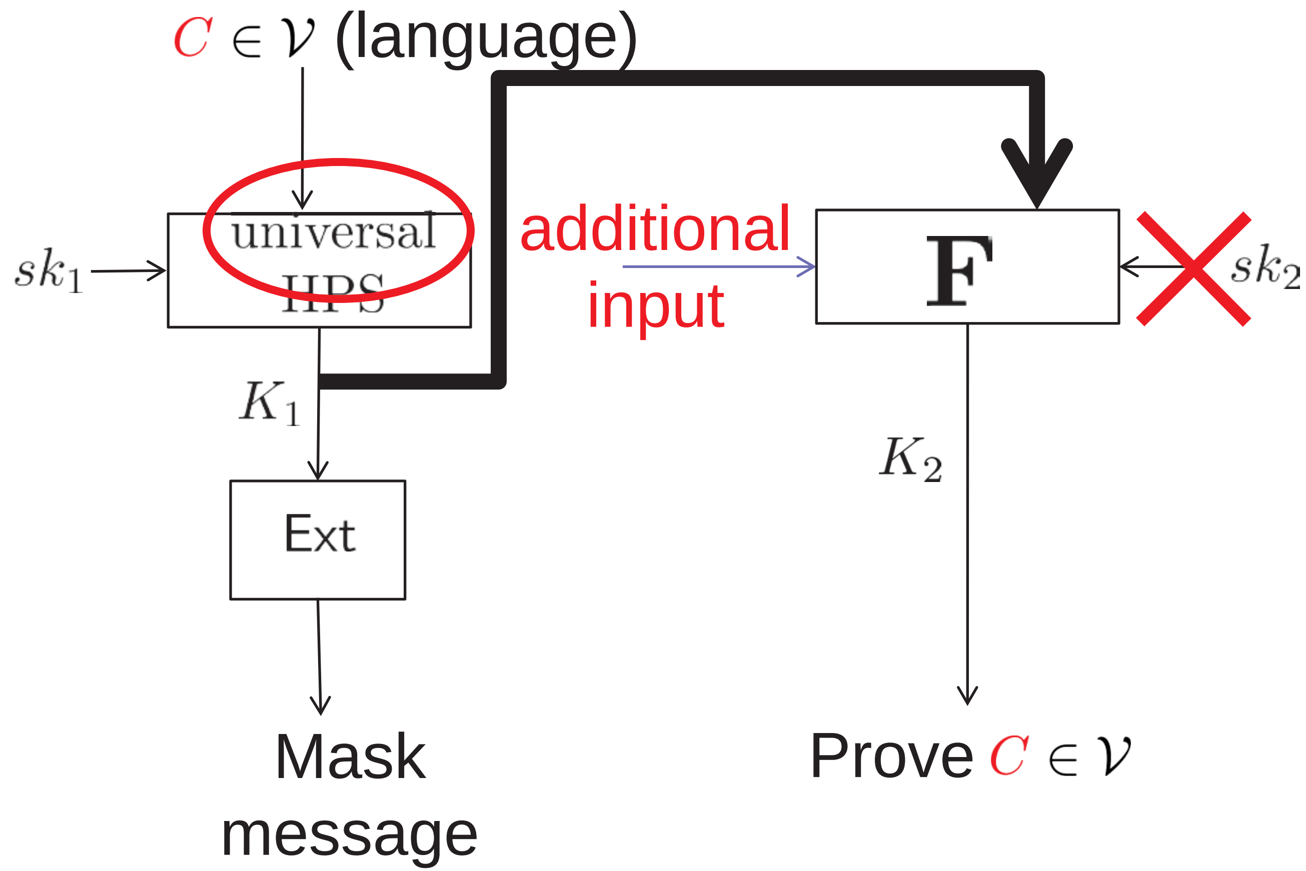
Our Approach



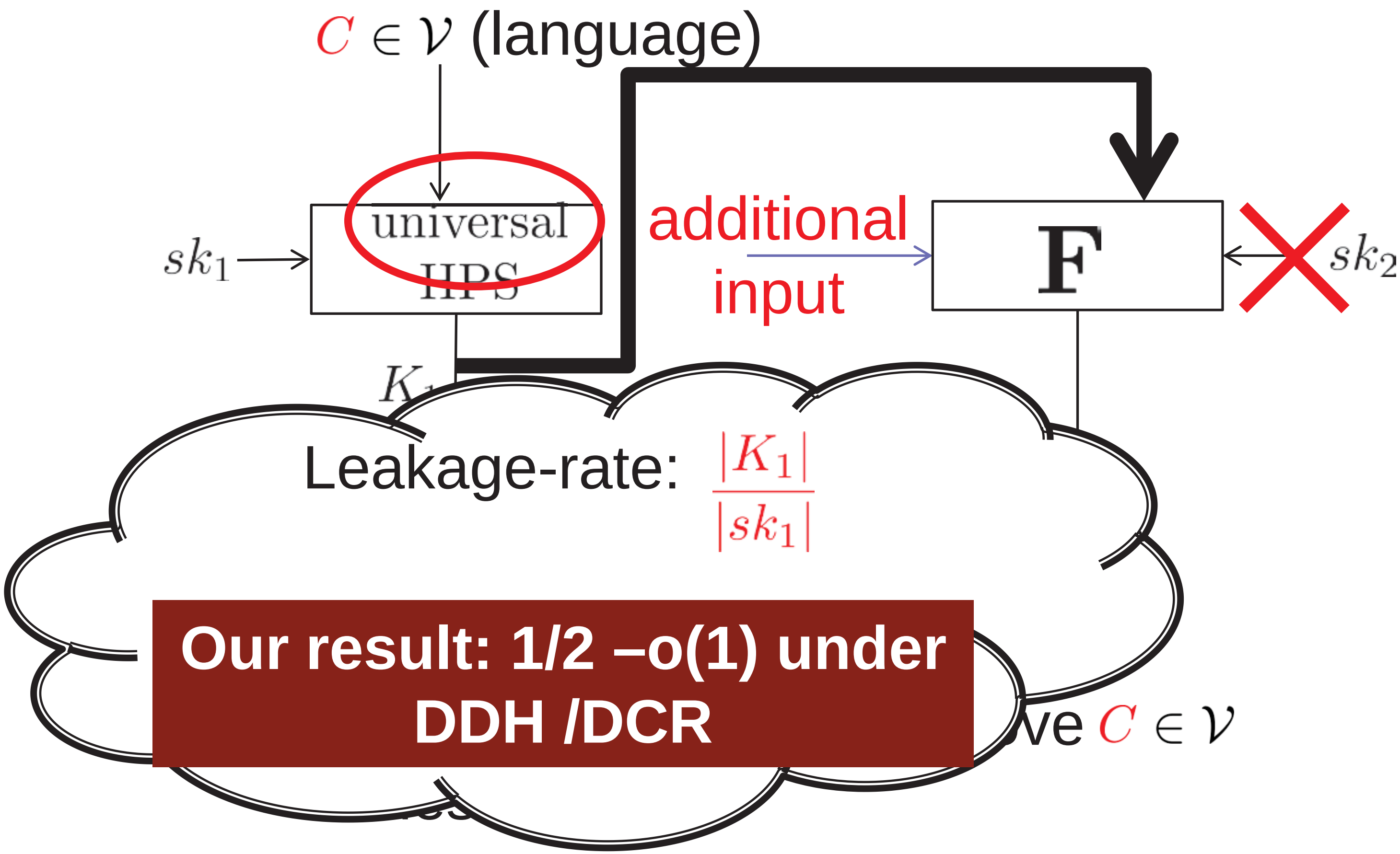
Our Approach



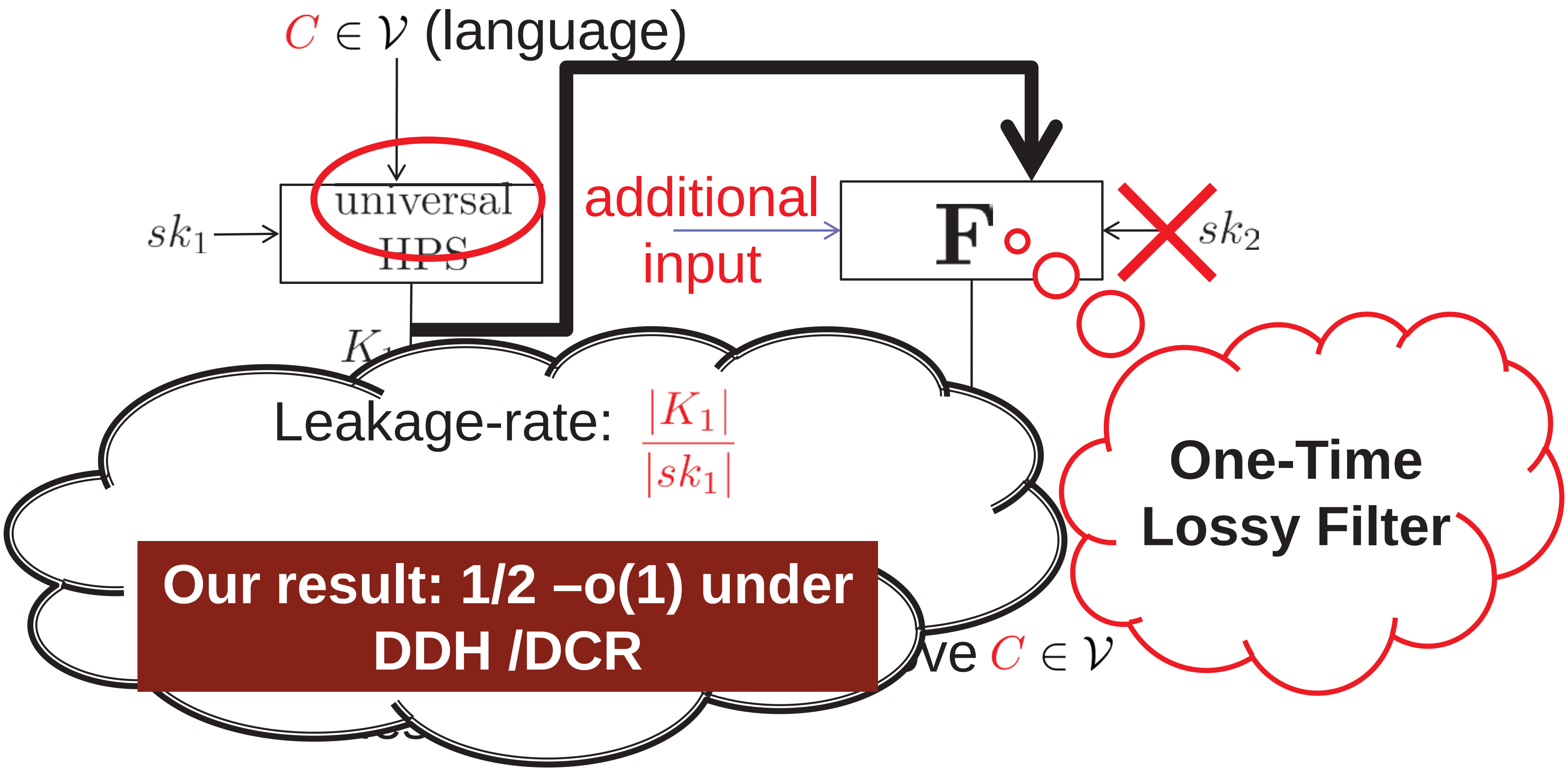
Our Approach



Our Approach



Our Approach



Part I:
One-Time Lossy Filter

Part II:
The Construction and Security Proof

Part III:
Instantiation and Comparison

Part I:
One-Time Lossy Filter

Part II:
The Construction and Security Proof

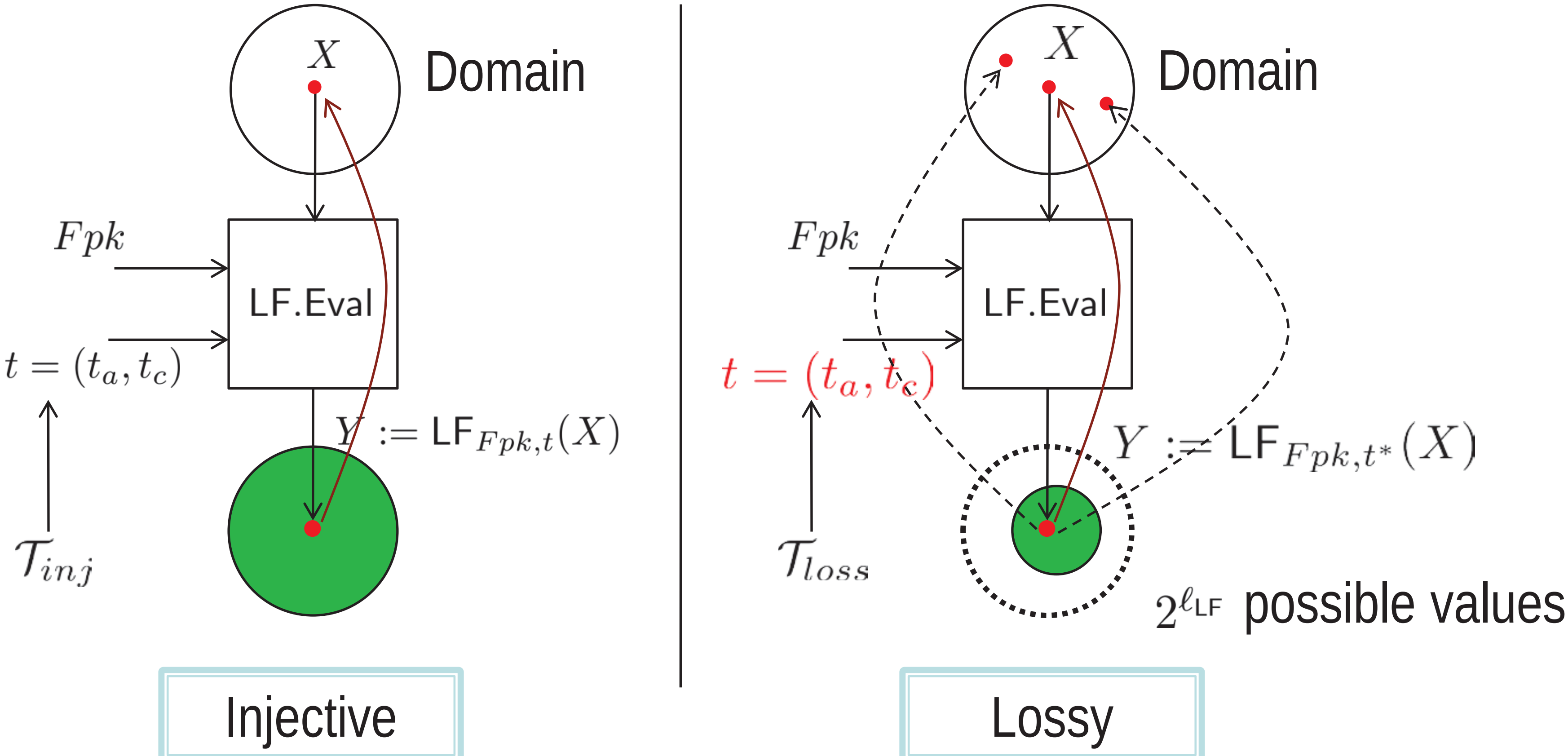
Part III:
Instantiation and Comparison

One-Time Lossy Filter

- Similar to (chameleon) all-but-one lossy trapdoor functions [PW08,LDL11]
  - not require efficient inversion.
- Simplified version of lossy algebraic filter (for CIRC-CCA security) [Hof13]
  - not require any algebraic property,
  - but require that lossy function reveals constant information of its input even for larger domain (by adapting some public parameters).
- Tag space: $\mathcal{T} = \underbrace{\boxed{\{0,1\}^*}}_{\text{auxiliary input part}} \times \underbrace{\boxed{\mathcal{T}_c}}_{\text{core tag part}} = \underbrace{\boxed{\mathcal{T}_{loss}}}_{\text{lossy tags}} \cup \underbrace{\boxed{\mathcal{T}_{inj}}}_{\text{injective tags}}$

Properties

lossiness/ indistinguishability/evasiveness



$$(t_a, t_c) \overset{c}{=} (t_a, t_c)$$

Properties

- Lossy tag is generated via a trapdoor Ftd.
  - For any auxiliary input t_a, it is easy to compute a core tag t_c, such that (t_a, t_c) is a lossy tag via the trapdoor.
  - Without the trapdoor, it is hard to generate a new non-injective tag even seen one lossy tag.

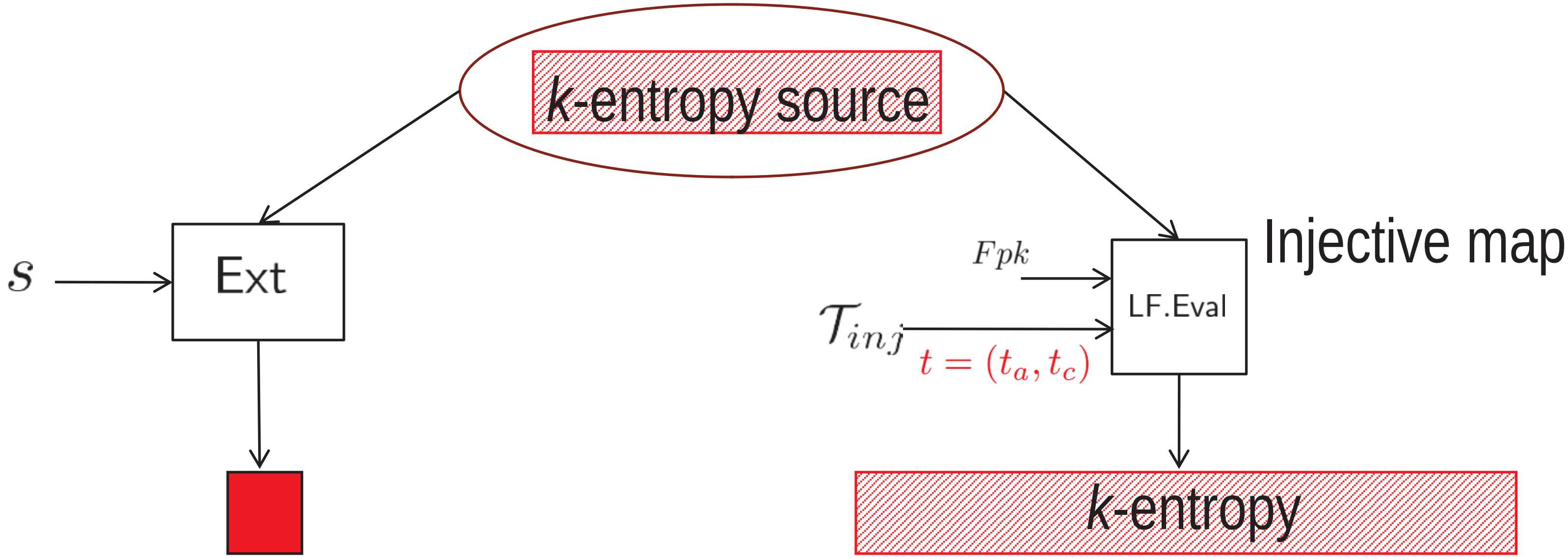
Part I:
One-Time Lossy Filter

Part II:
The Construction and Security Proof

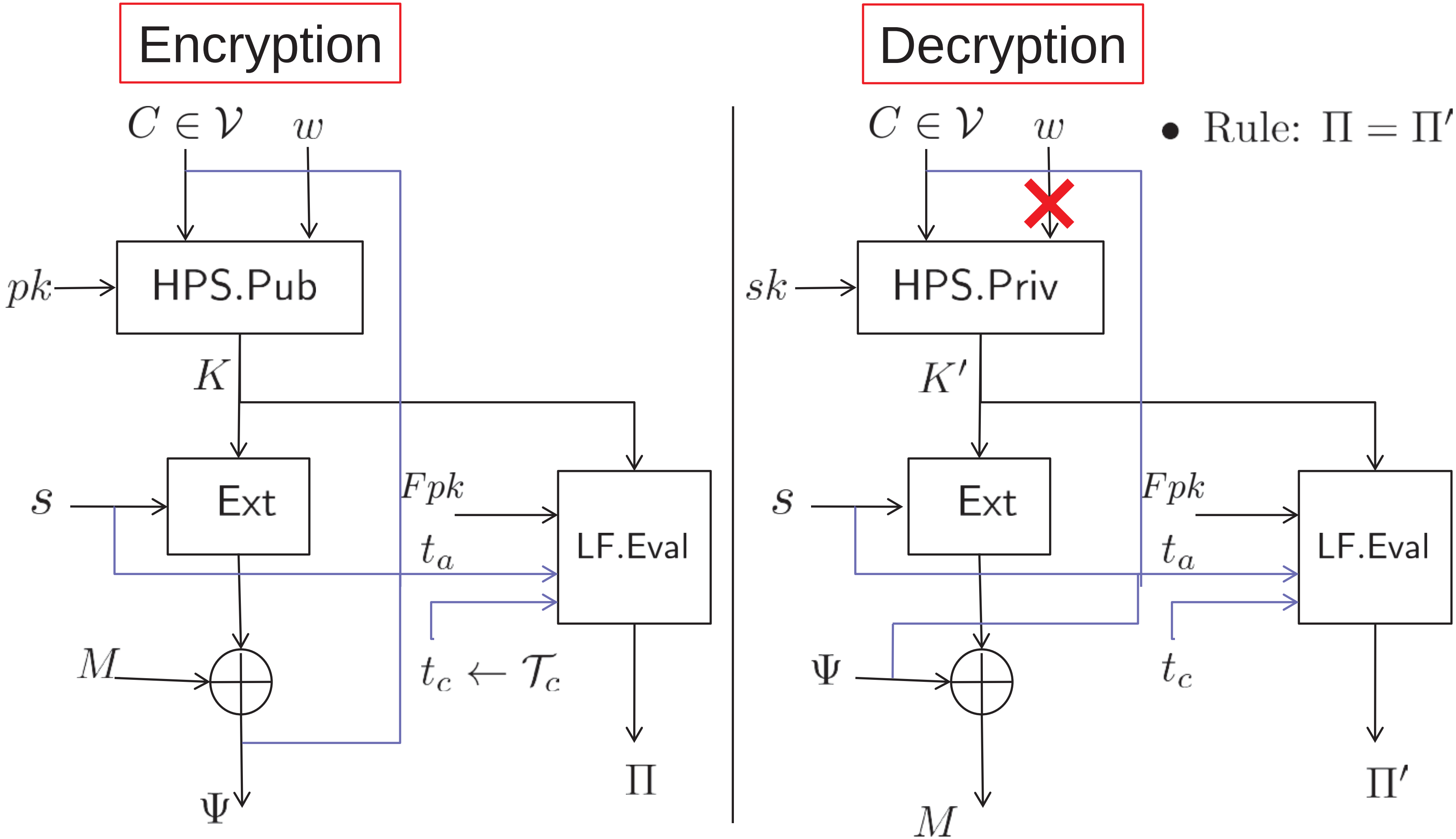
Part III:
Instantiation and Comparison

Construction Idea

- One entropy source used in two purposes.
  - Mask the plaintext (applying an extractor)
  - Verify the well-formedness of the ciphertext (applying a special injective function: one-time lossy filter)

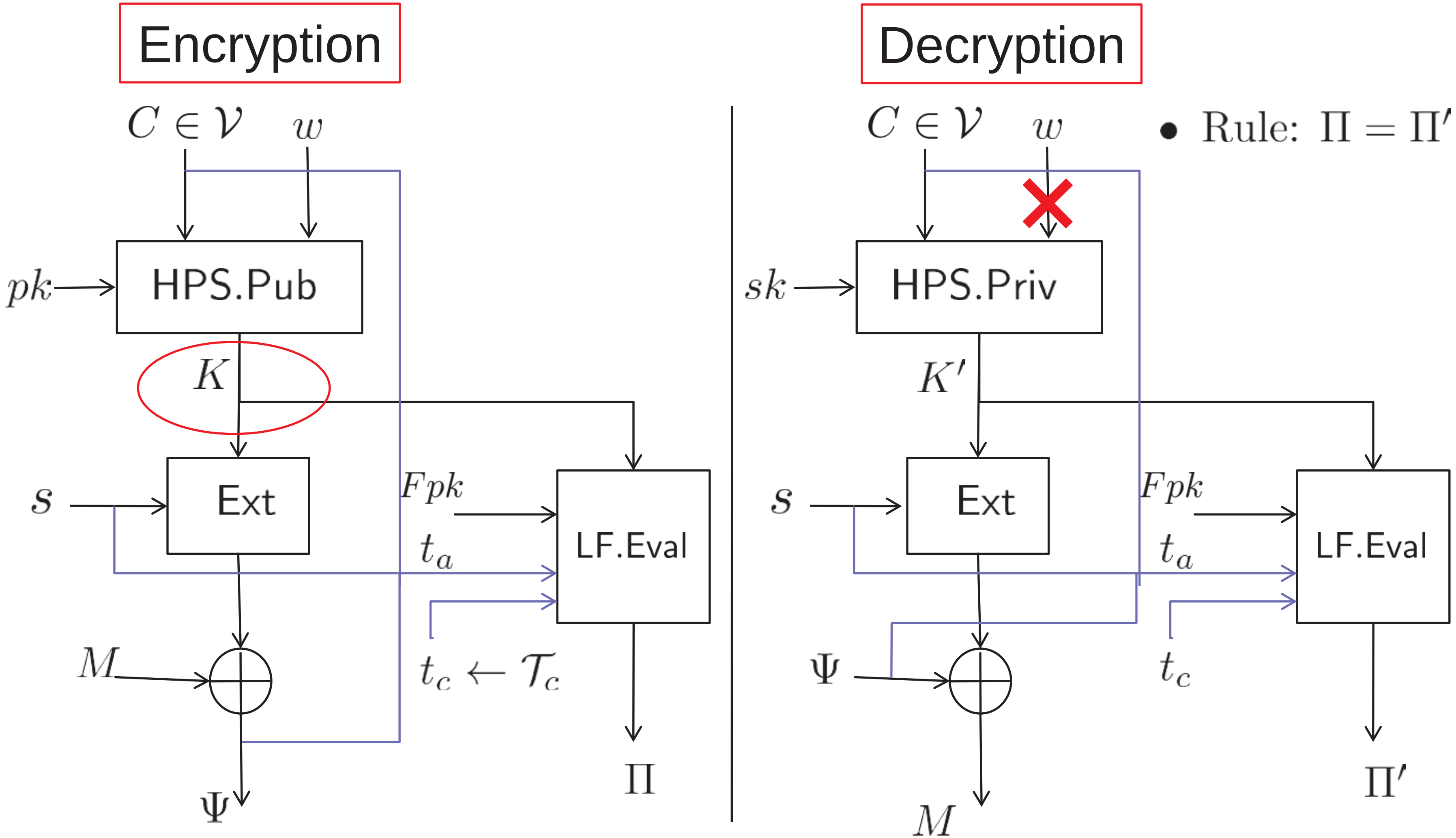


The PKE Scheme



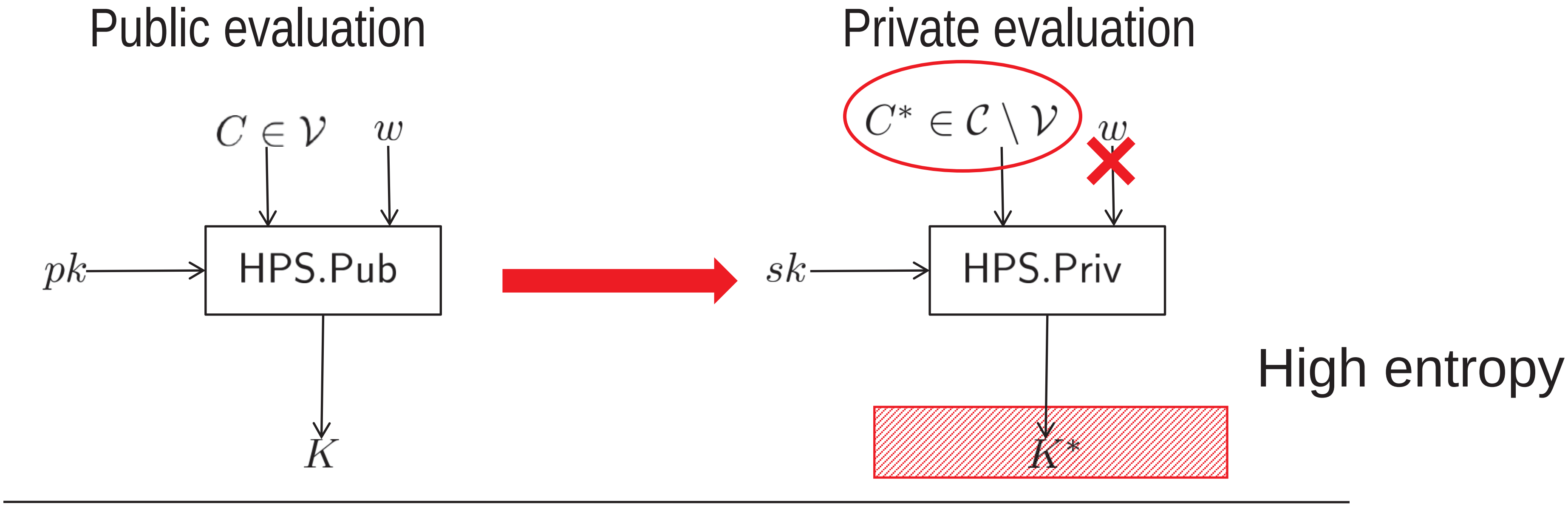
Ciphertext: $CT = (C, s, \Psi, \Pi, t_c)$

The PKE Scheme

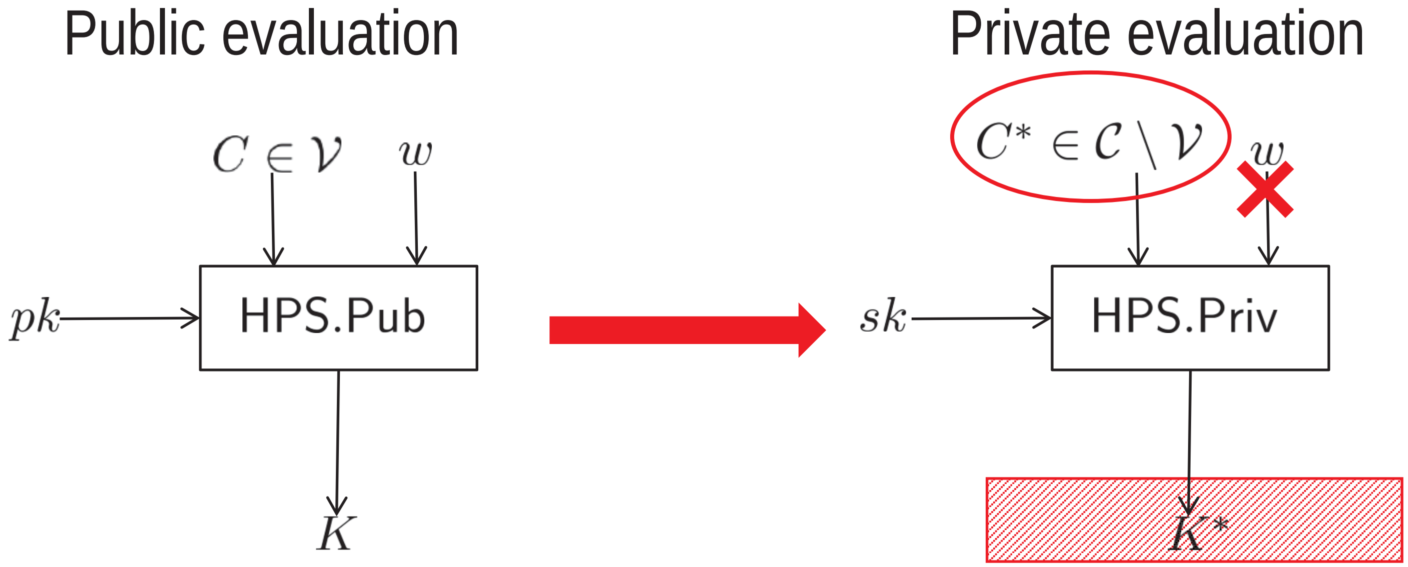


Ciphertext: $CT = (C, s, \Psi, \Pi, t_c)$

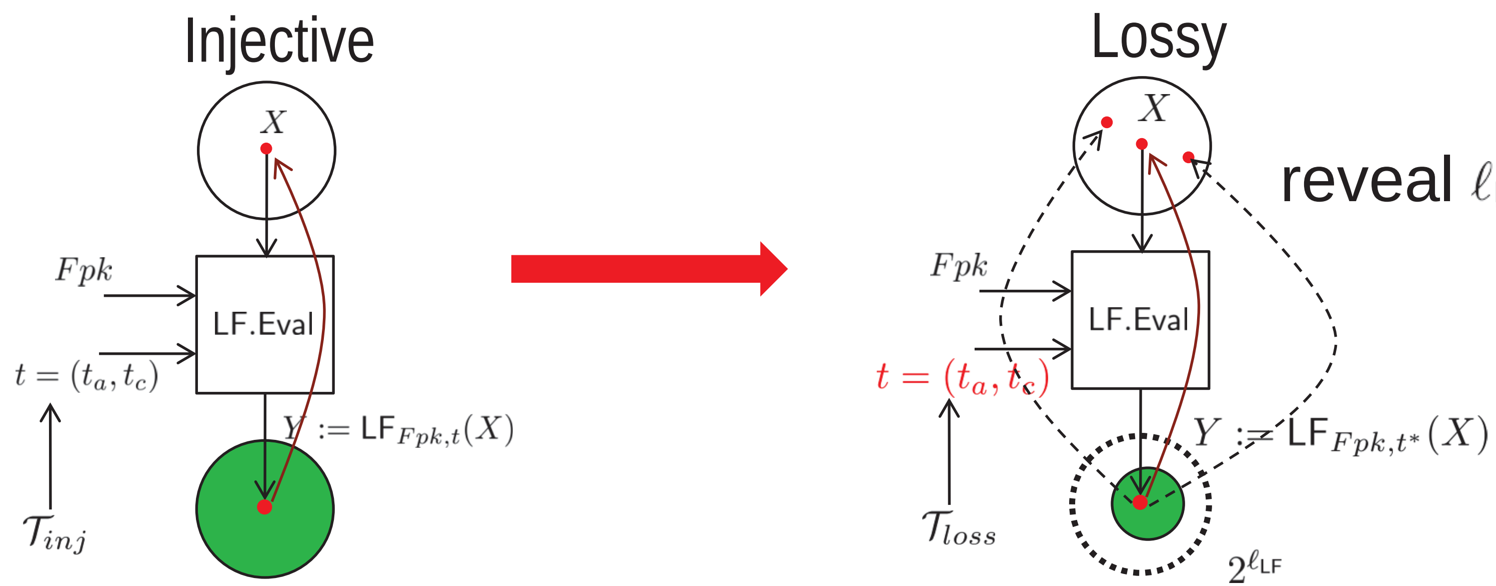
Proof Idea: challenge ciphertext



Proof Idea: challenge ciphertext

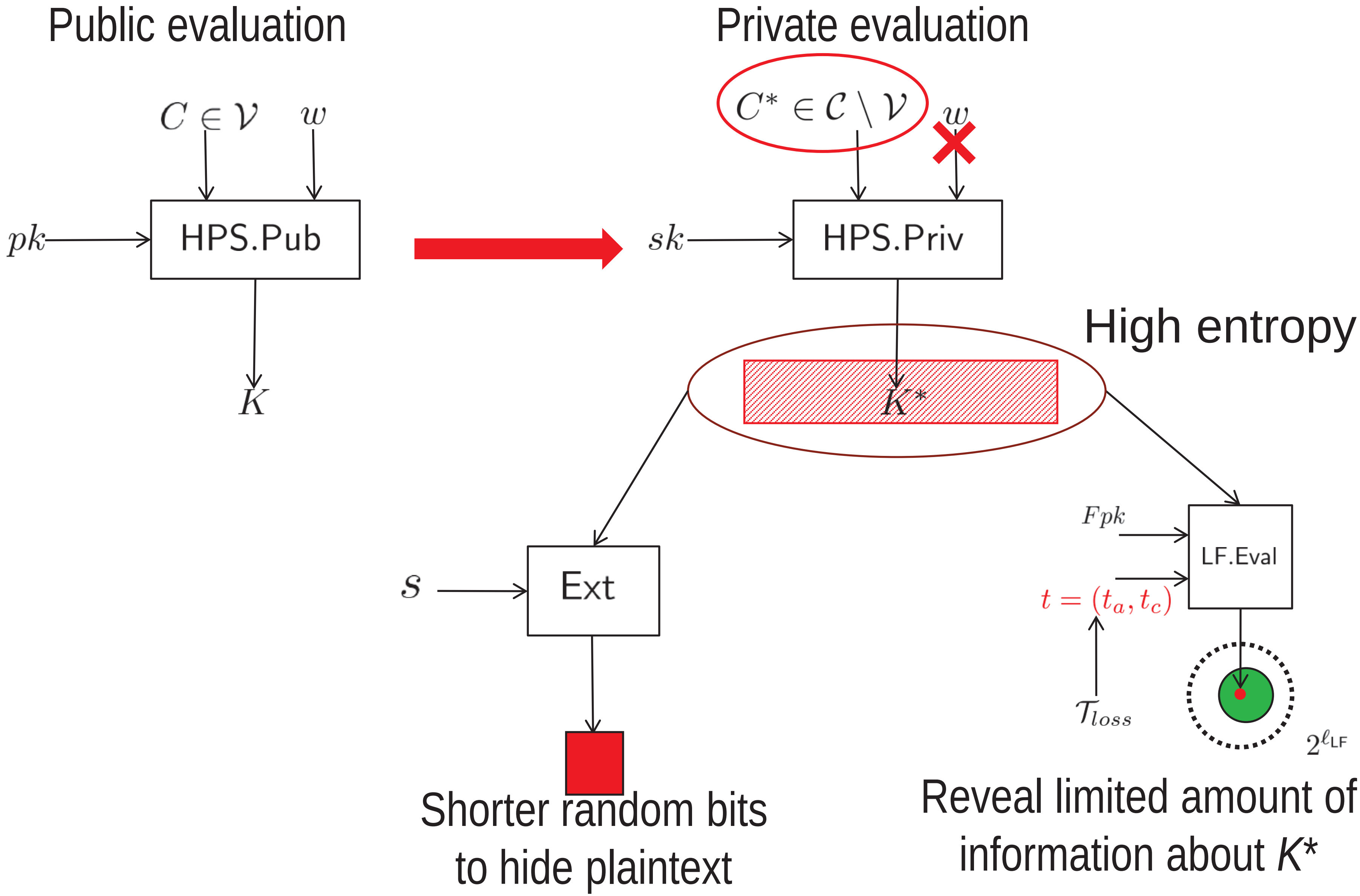


High entropy

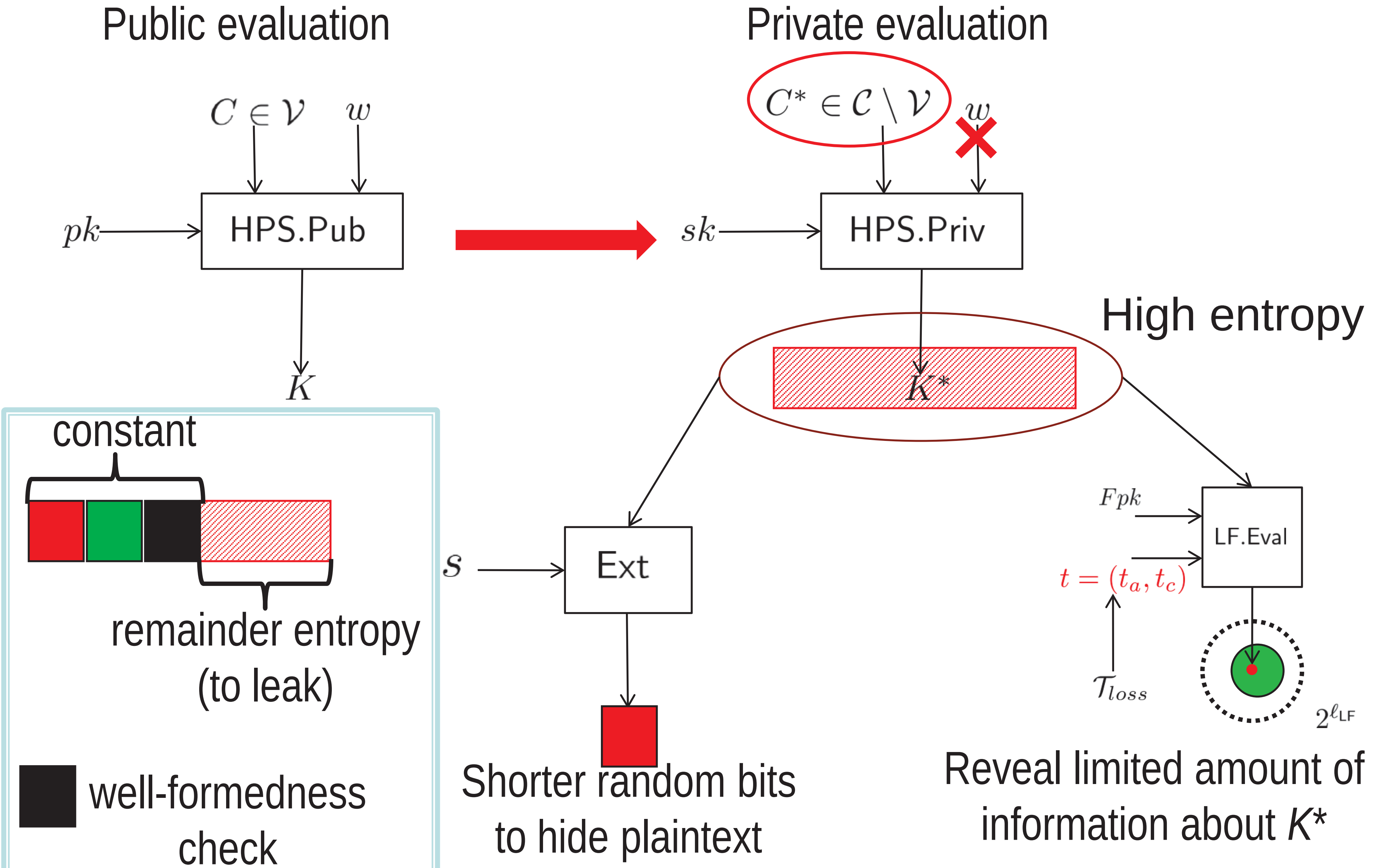


reveal ℓ_{LF} bits info.

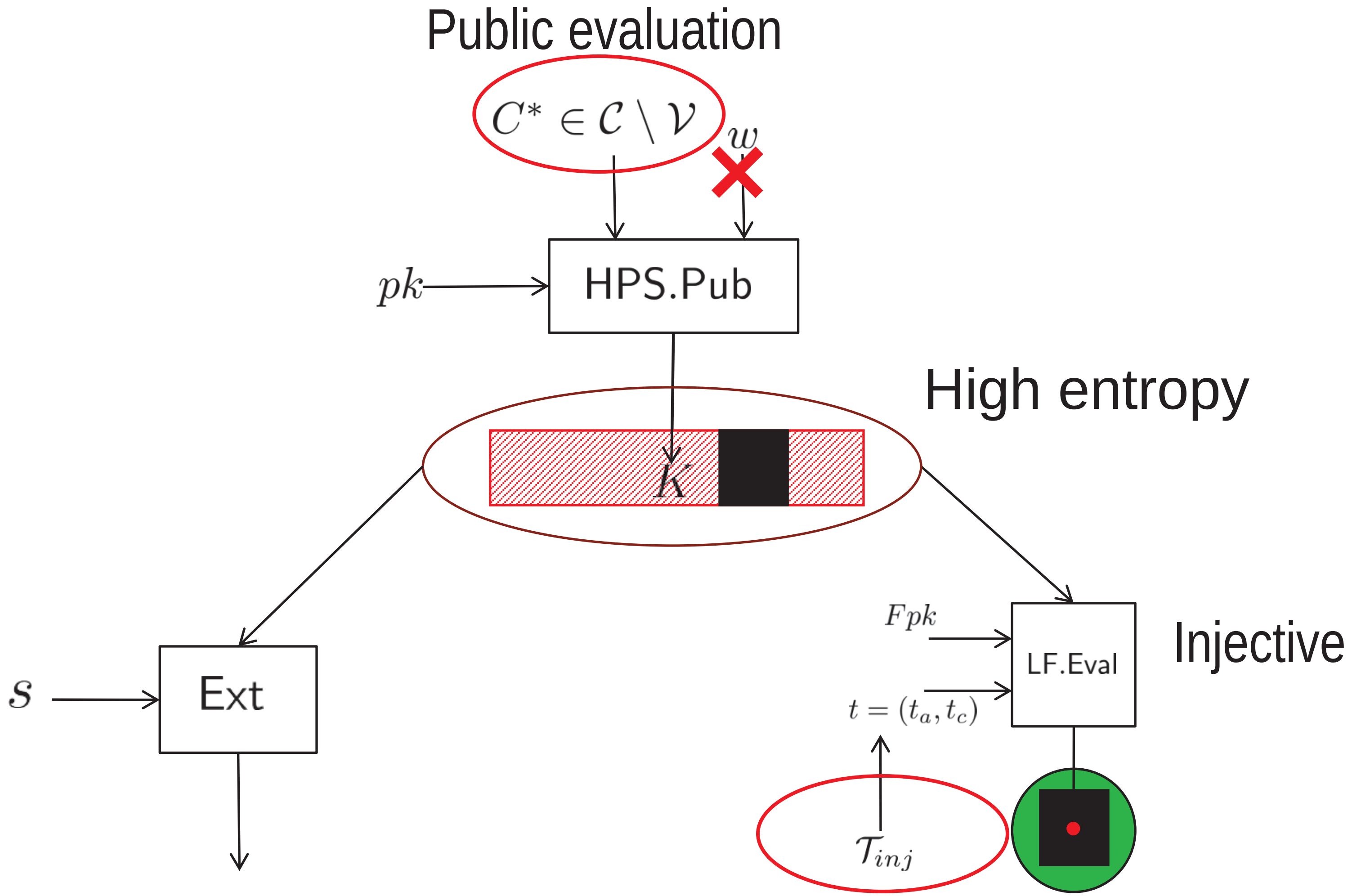
Proof Idea: challenge ciphertext



Proof Idea: challenge ciphertext



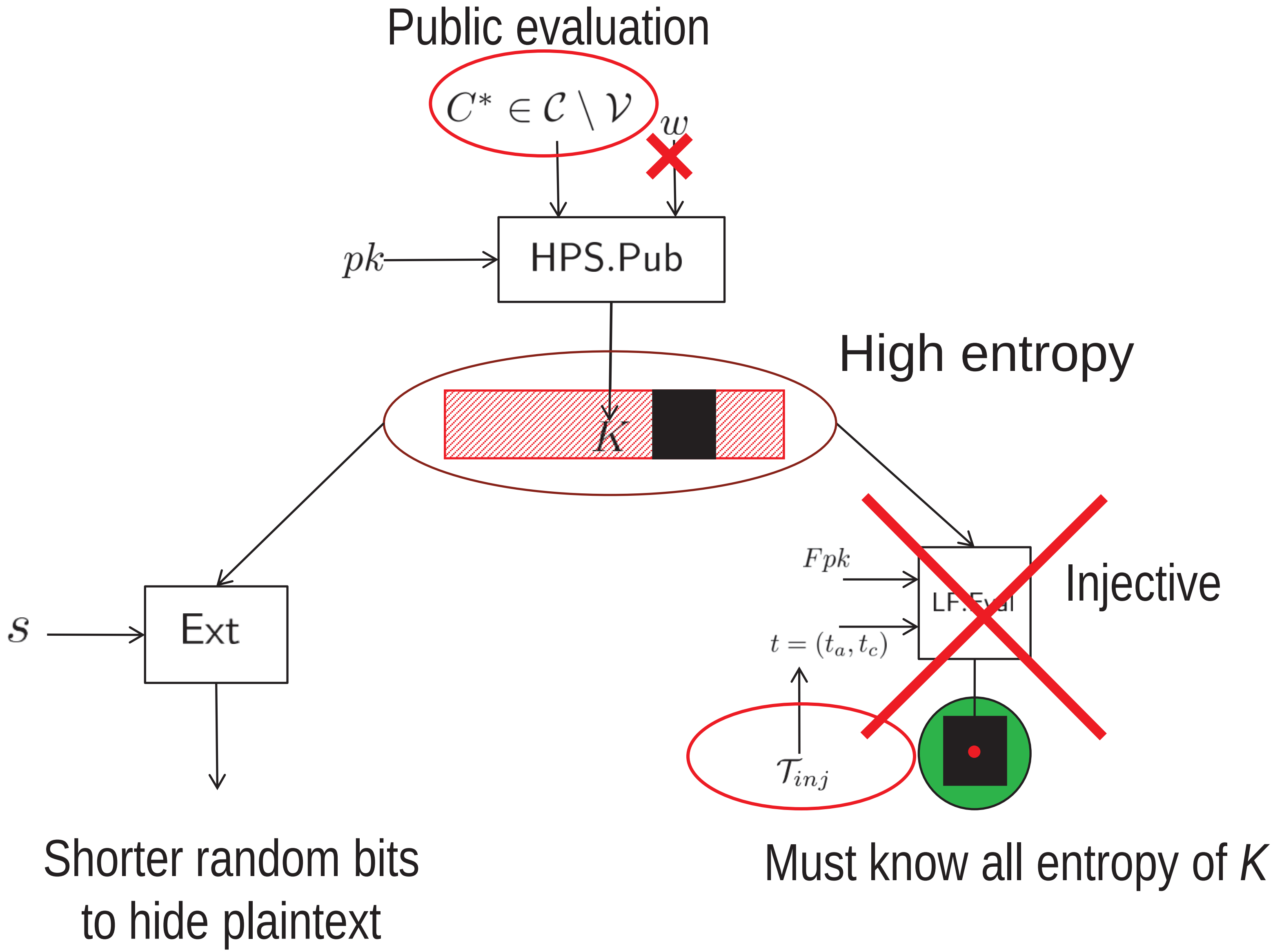
Proof Idea: decryption query



Shorter random bits
to hide plaintext

Must know all entropy of *K*

Proof Idea: decryption query



Proof Summary

Encryption query

HPS / OT-LF	valid	invalid
injective	√	×
lossy	×	√

Decryption queries

HPS / OT-LF	valid	invalid
injective	√	×
lossy	×	×

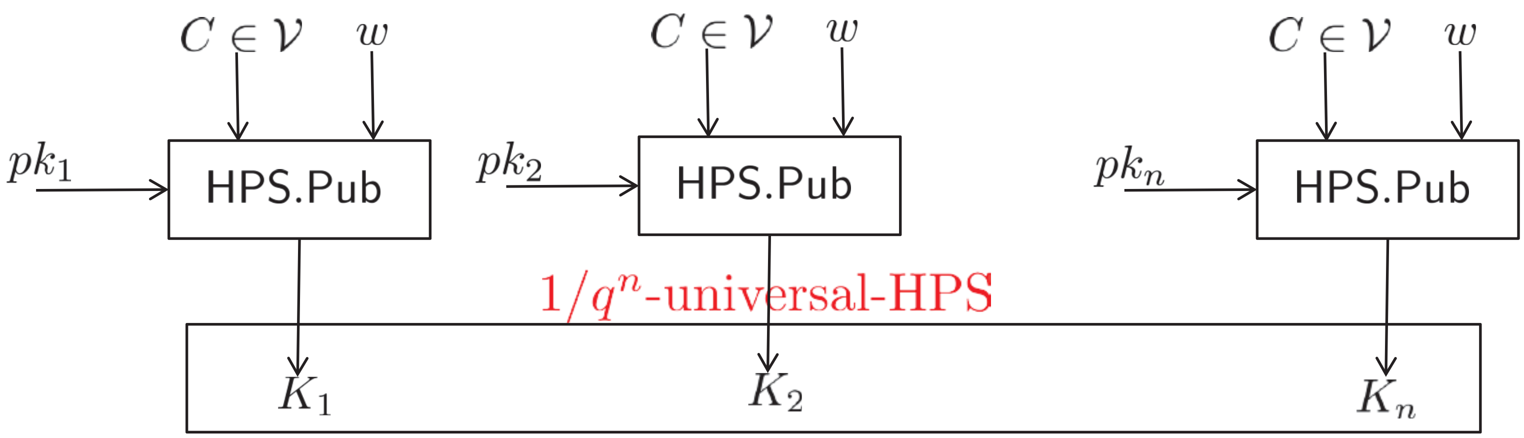
Part I:
One-Time Lossy Filter

Part II:
The Construction and Security Proof

Part III:
Instantiation and Comparison

Instantiation: <q, G, g>

- n-fold parallelization of [CS02] construction.



$$\begin{cases} |\mathcal{K}| & = q^n \\ |\mathcal{SK}| & = q^{2n} \end{cases} \Longrightarrow \text{leakage-rate is } 1/2 - o(1)$$

- OT-LF, similar to DDH-based lossy trapdoor function: Domain: $\mathbb{Z}_q^n$, image values: $|q|$

$$F(x, t^*)$$

Chameleon hash

$$\mathsf{CH}(t_a, t_c)$$

Efficiency Comparison

Table 1: Relations between leakage-rate and ciphertext overhead (# 80-bit)

Schemes \ Leakage-rate	1/8	1/6	1/4	1/3	3/8	2/5	1/2	1
DHLW10 [11]	94	95.2	98	101.5	103.6	105	112	-
GHV12 [16]	32	32	36	36	40	40	44	-
NS12 [28]	36	-	-	-	-	-	-	-
LZSS12 [25]	18	27	-	-	-	-	-	-
Ours	**12**	**12**	**14**	**20**	**24**	**30**	-	-

- Advantages:
  - Achieve 1/2-o(1) under DDH/DCR
  - shorter ciphertext overhead (when leakage rate ≤2/5)
  - better than HPS-based construction [28,25]
- Disadvantages: below 1/2.

Conclusion and Further Work

- A new primitive: one-time lossy filter
- A generic construction of LR-CCA-secure PKE
- Efficient instantiations under DDH and DCR assumptions (with better leakage-rate 1/2-o(1))

Further work:

- Improve the leakage-rate to [1/2, 1) without loss the practicality.
- Leakage-flexible CCA-secure PKE without pairing.

上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

Thank You!