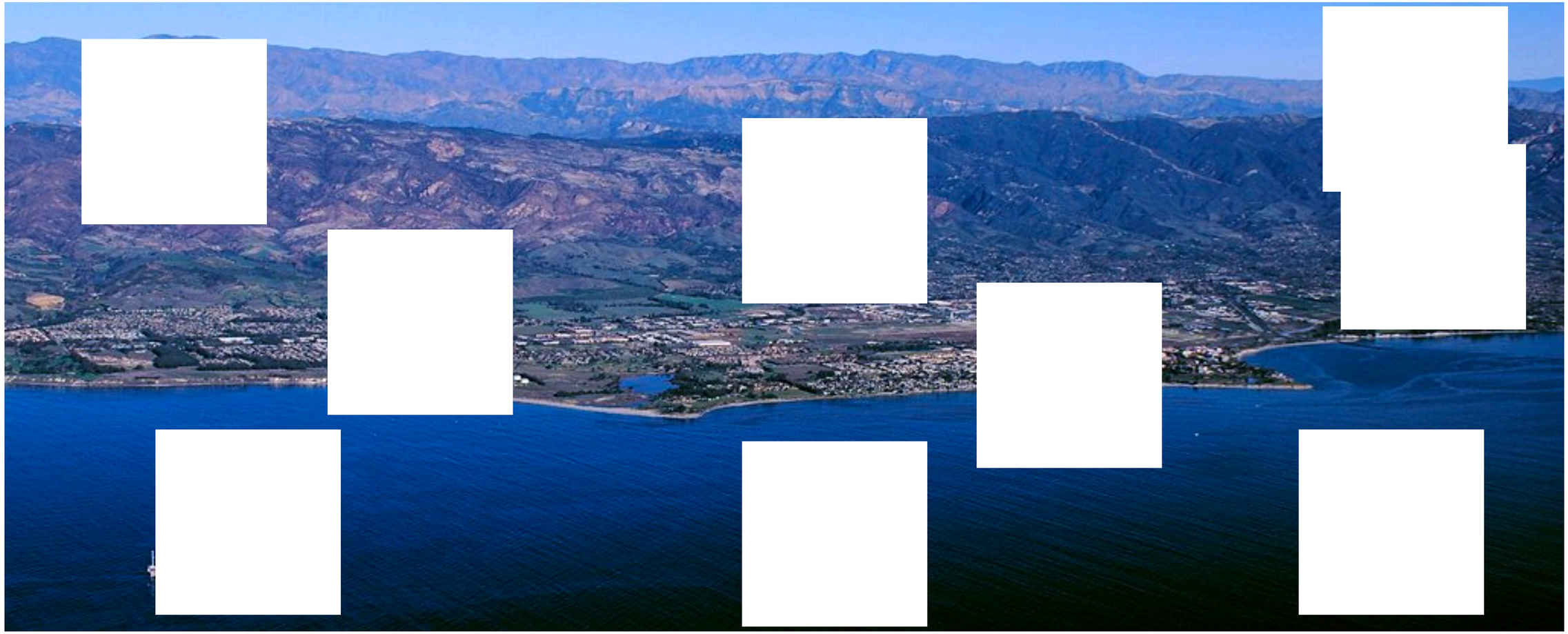
Collusion Resistant Watermarkable PRFs from Standard Assumptions

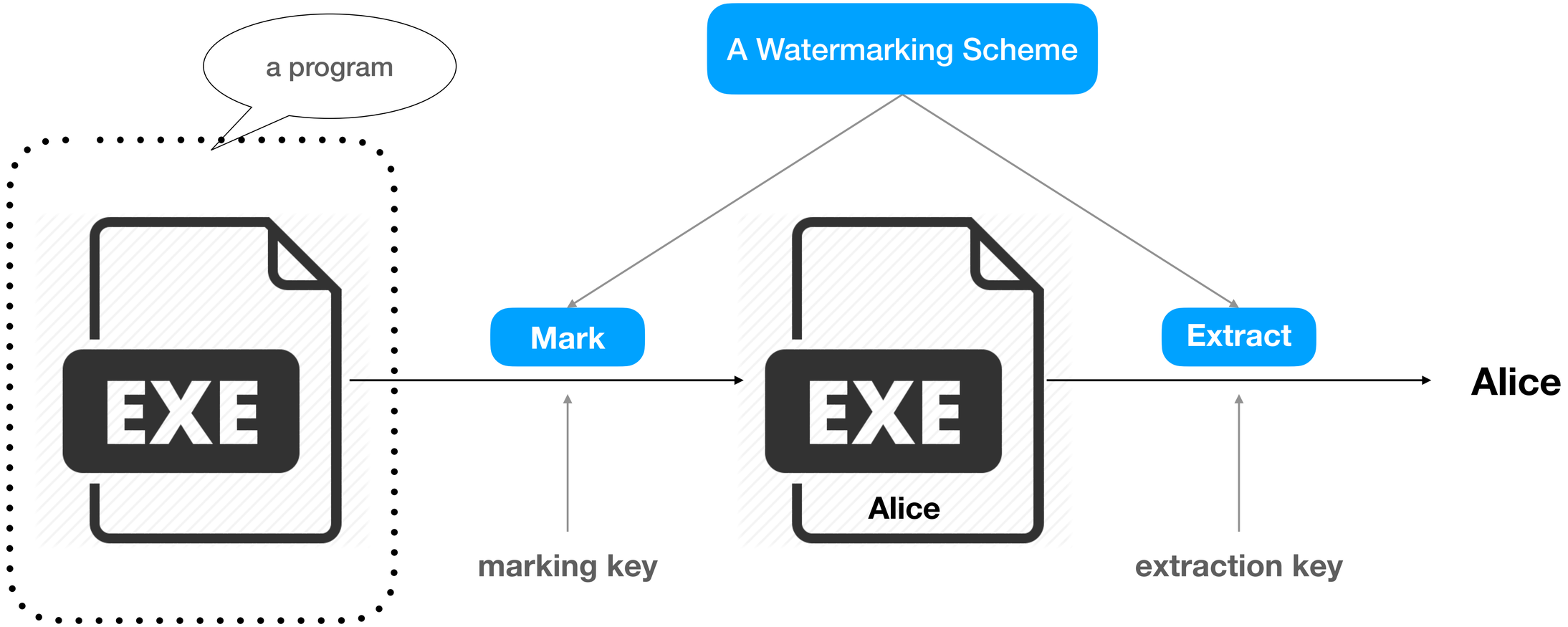
Rupeng Yang, Man Ho Au, Zuoxia Yu, and Qiuliang Xu

Watermarking A Cryptographic Program

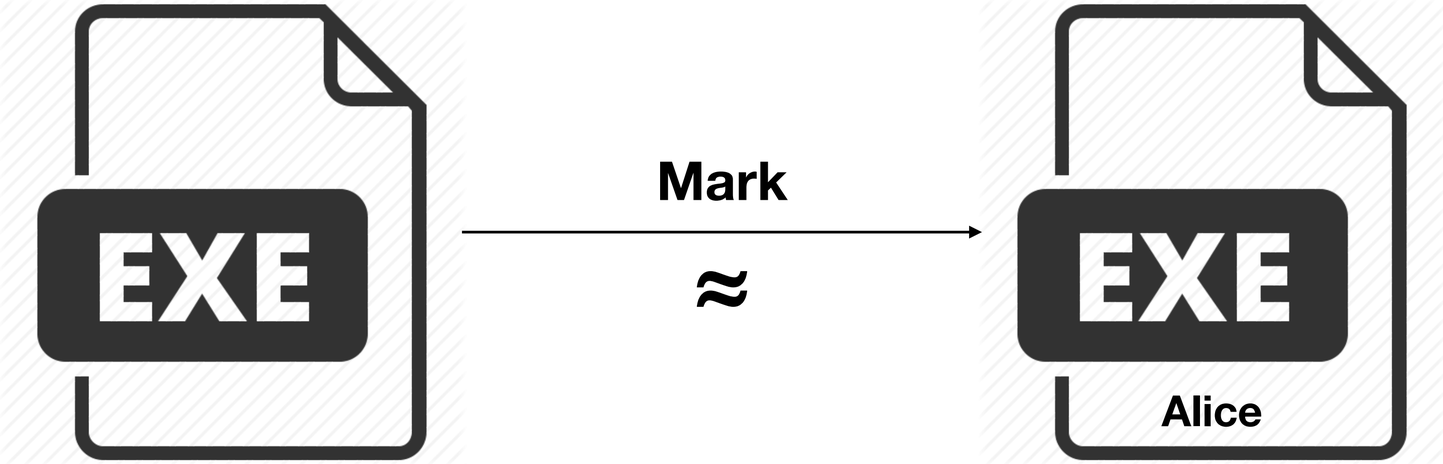
Watermarking A Cryptographic Program



Watermarking A Cryptographic Program

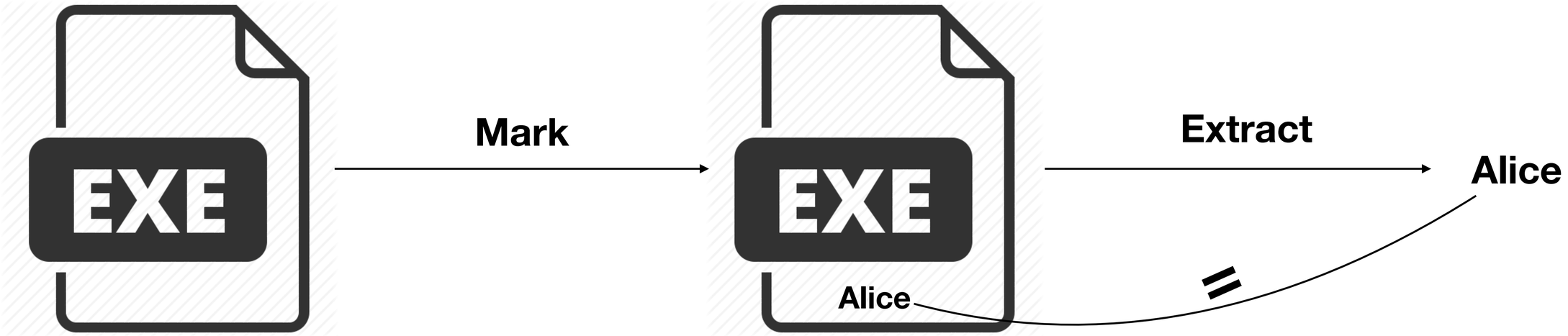


Watermarking A Cryptographic Program



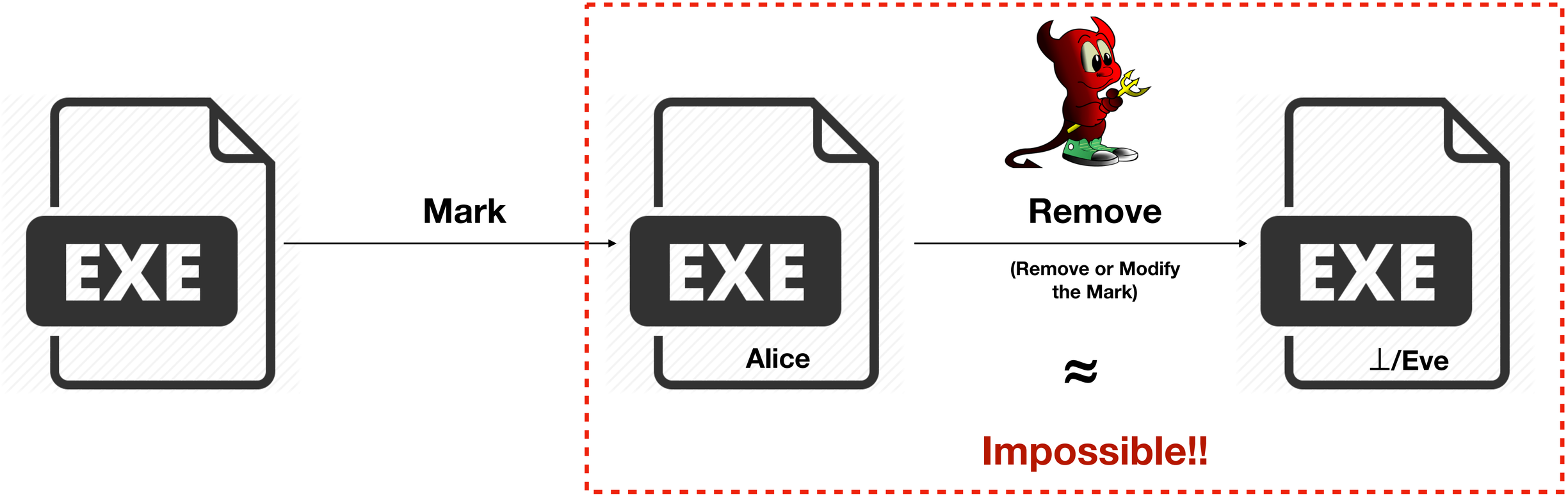
Correctness Requirement: Functionality Preserving

Watermarking A Cryptographic Program



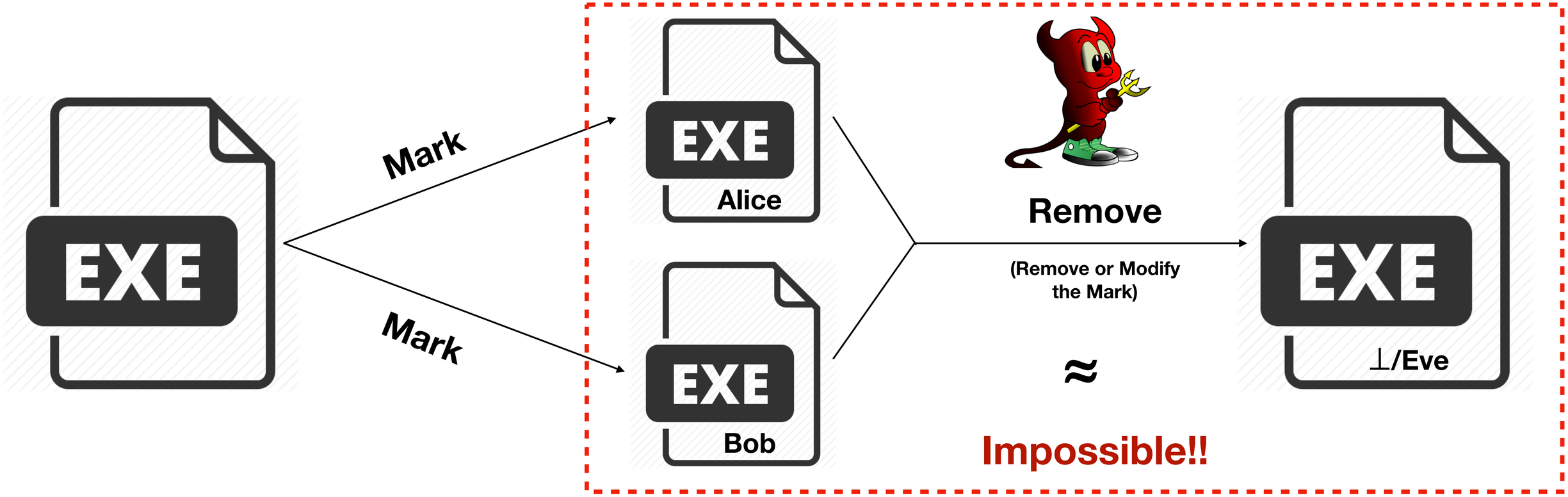
Correctness Requirement: Extraction Correctness

Watermarking A Cryptographic Program



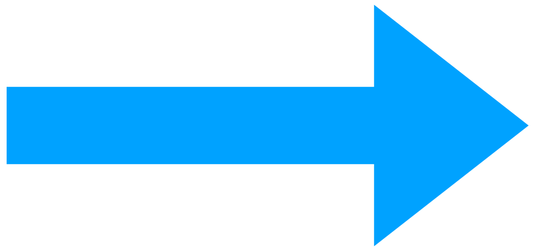
Security Requirement: (Single Challenge) Unremovability

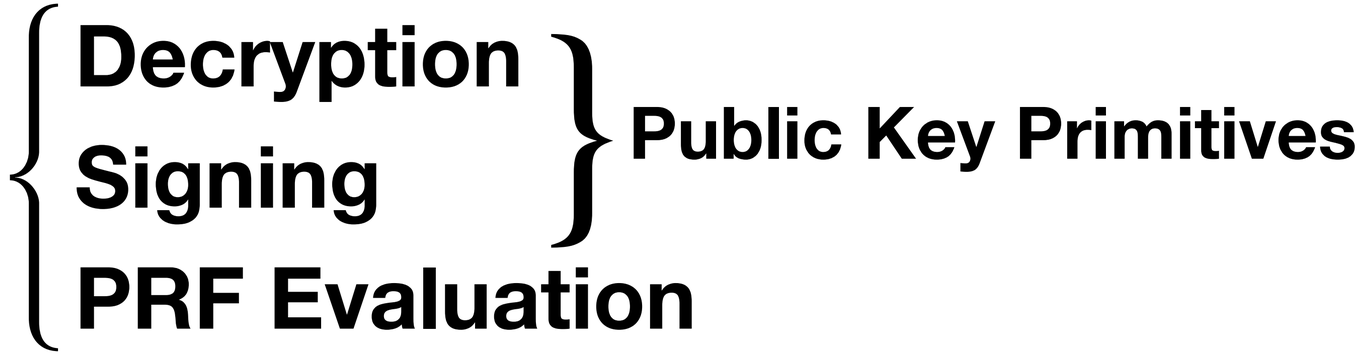
Watermarking A Cryptographic Program



Security Requirement: Collusion Resistant Unremovability

Watermarking A Cryptographic Program





It is *impossible* to watermark a learnable functionality.

Previous Works on Watermarking Cryptographic Programs

- Watermarking for public key primitives
  - Collusion Resistant unremovability from standard assumption ([GKM+ 19])

Previous Works on Watermarking Cryptographic Programs

- Watermarking for public key primitives
  - Collusion Resistant unremovability from standard assumption ([GKM+ 19])
- Watermarking for PRF
  - Single challenge unremovability from standard assumption ([KW 17, QWZ 18, KW 19])
  - Collusion Resistant unremovability from indistinguishability obfuscation ([YAL+ 19])

Previous Works on Watermarking Cryptographic Programs

- Watermarking for public key primitives
  - Collusion Resistant unremovability from standard assumption ([GKM+ 19])
- Watermarking for PRF
  - Single challenge unremovability from standard assumption ([KW 17, QWZ 18, KW 19])
  - Collusion Resistant unremovability from indistinguishability obfuscation ([YAL+ 19])

Can we construct collusion resistant watermarkable PRF from standard assumption?

Warmup: Constrained PRF

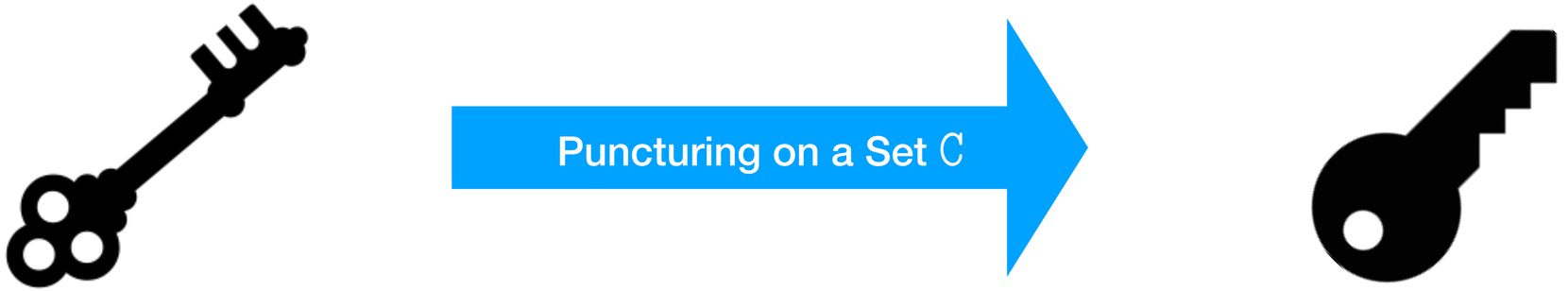
SK

Puncturing on a Set $\mathbb{C}$

$SK_{\mathbb{C}}$

Correctness: if $x \notin \mathbb{C}, \mathsf{F}_{SK}(x) = F_{SK_{\mathbb{C}}}(x)$

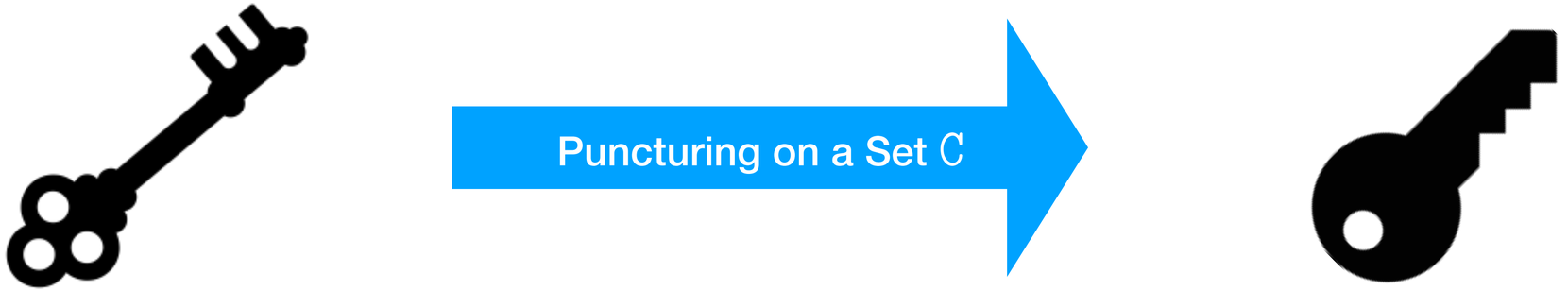
Warmup: Constrained PRF



Correctness: if $x \notin \mathbb{C}, \mathsf{F}_{SK}(x) = F_{SK_{\mathbb{C}}}(x)$

Pseudorandomness: if $x \in \mathbb{C}$, $\mathsf{F}_{SK}(x)$ is hidden given $SK_{\mathbb{C}}$

Warmup: Constrained PRF

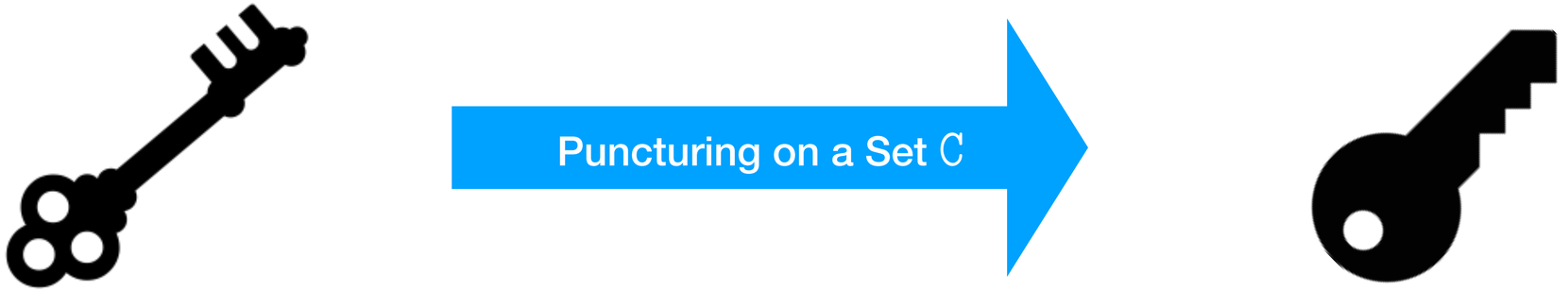


Correctness: if $x \notin \mathsf{C}, \mathsf{F}_{SK}(x) = F_{SK_{\mathsf{C}}}(x)$

Pseudorandomness: if $x \in \mathsf{C}$, $\mathsf{F}_{SK}(x)$ is hidden given SK_{C}

Privacy: C is hidden given SK_{C}

Warmup: Constrained PRF



Correctness: if $x \notin \mathbb{C}, \mathsf{F}_{SK}(x) = F_{SK_{\mathbb{C}}}(x)$

Pseudorandomness: if $x \in \mathbb{C}$, $\mathsf{F}_{SK}(x)$ is hidden given $SK_{\mathbb{C}}$

Privacy: $\mathbb{C}$ is hidden given $SK_{\mathbb{C}}$

Collusion Resistance: Security hold given more than one constrained keys

Warmup: Construction of Watermarkable PRF (Mark Embedding)

PRF Key (SK)

Puncturing on $\{x^*\}$

Watermarked Key ($SK_{\{x^*\}}$)

Mark:

1. **Generate x^***
2. **Puncture SK on $\{x^*\}$**

Extract:

1. **Generate x^***
2. **Test if the circuit is punctured on $\{x^*\}$**

Warmup: Construction of Watermarkable PRF (Mark Embedding)

PRF Key (SK)

Watermarked Key ($SK_{\{x^*\}}$)

Mark:

1. **Generate x^***
2. **Puncture SK on $\{x^*\}$**

Extract:

1. **Generate x^***
2. **Test if the circuit is punctured on $\{x^*\}$**

Correctness: Programmability/Extractability/… of Constrained PRF

Warmup: Construction of Watermarkable PRF (Mark Embedding)

PRF Key (SK)

Puncturing on $\{x^*\}$

Watermarked Key ($SK_{\{x^*\}}$)

Mark:

1. **Generate x^***
2. **Puncture SK on $\{x^*\}$**

Extract:

1. **Generate x^***
2. **Test if the circuit is punctured on $\{x^*\}$**

Correctness: Programmability/Extractability/… of Constrained PRF

Security: Pseudorandomness and/or Privacy of Constrained PRF

Warmup: Construction of Watermarkable PRF (Message Embedding)

Puncturing on $\mathbb{C}_m = \{x^*_{1,m_1}, x^*_{2,m_2}, \ldots, x^*_{n,m_n}\}$

PRF Key (SK)

$m = (m_1, m_2, \ldots m_n) \in \{0,1\}^n$

Watermarked Key ($SK_{\mathbb{C}_m}$)

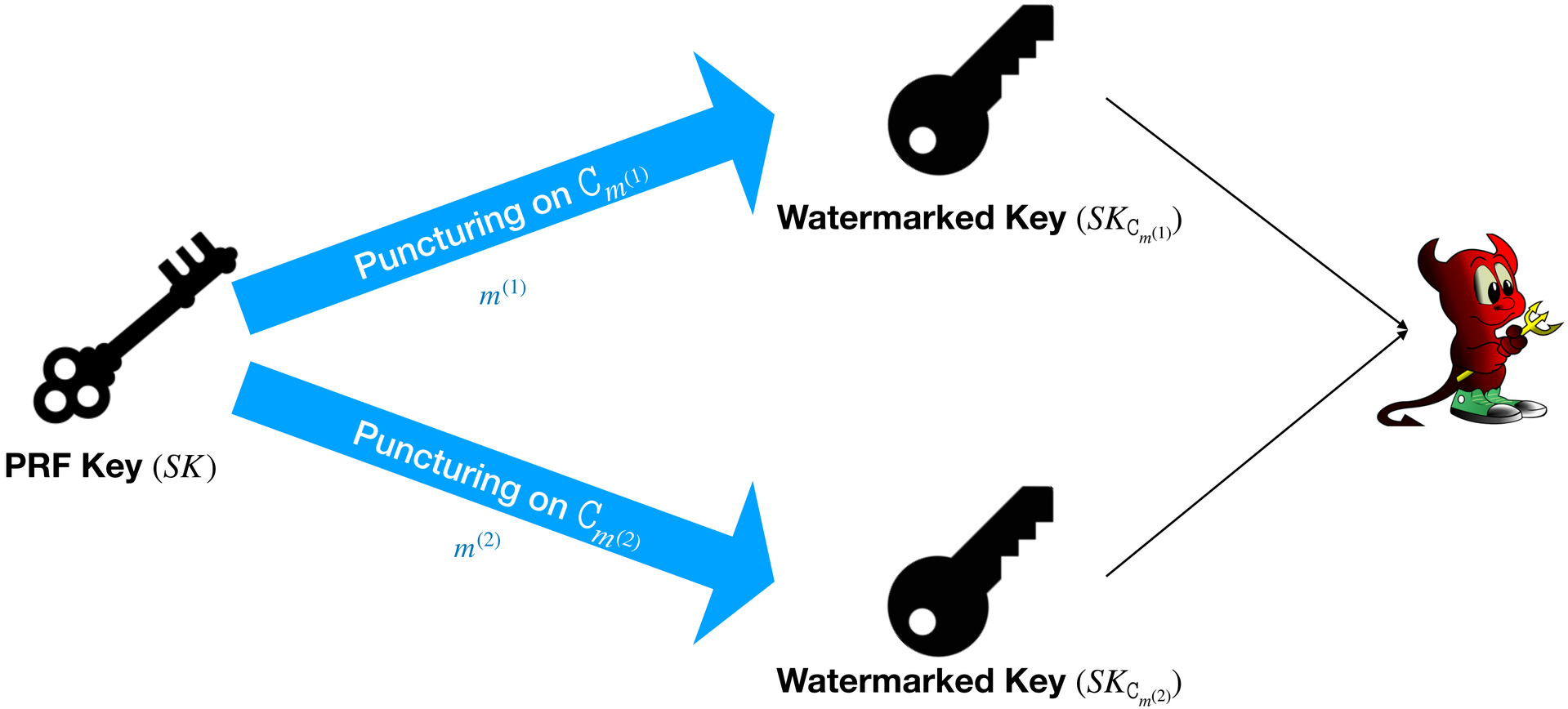
Mark:

1. **Generate** $\begin{bmatrix} x^*_{1,0}, x^*_{2,0}, \ldots x^*_{n,0} \\ x^*_{1,1}, x^*_{2,1}, \ldots x^*_{n,1} \end{bmatrix}$
2. **Puncture** SK **on** $\{x^*_{1,m_1}, x^*_{2,m_2}, \ldots, x^*_{n,m_n}\}$

Extract:

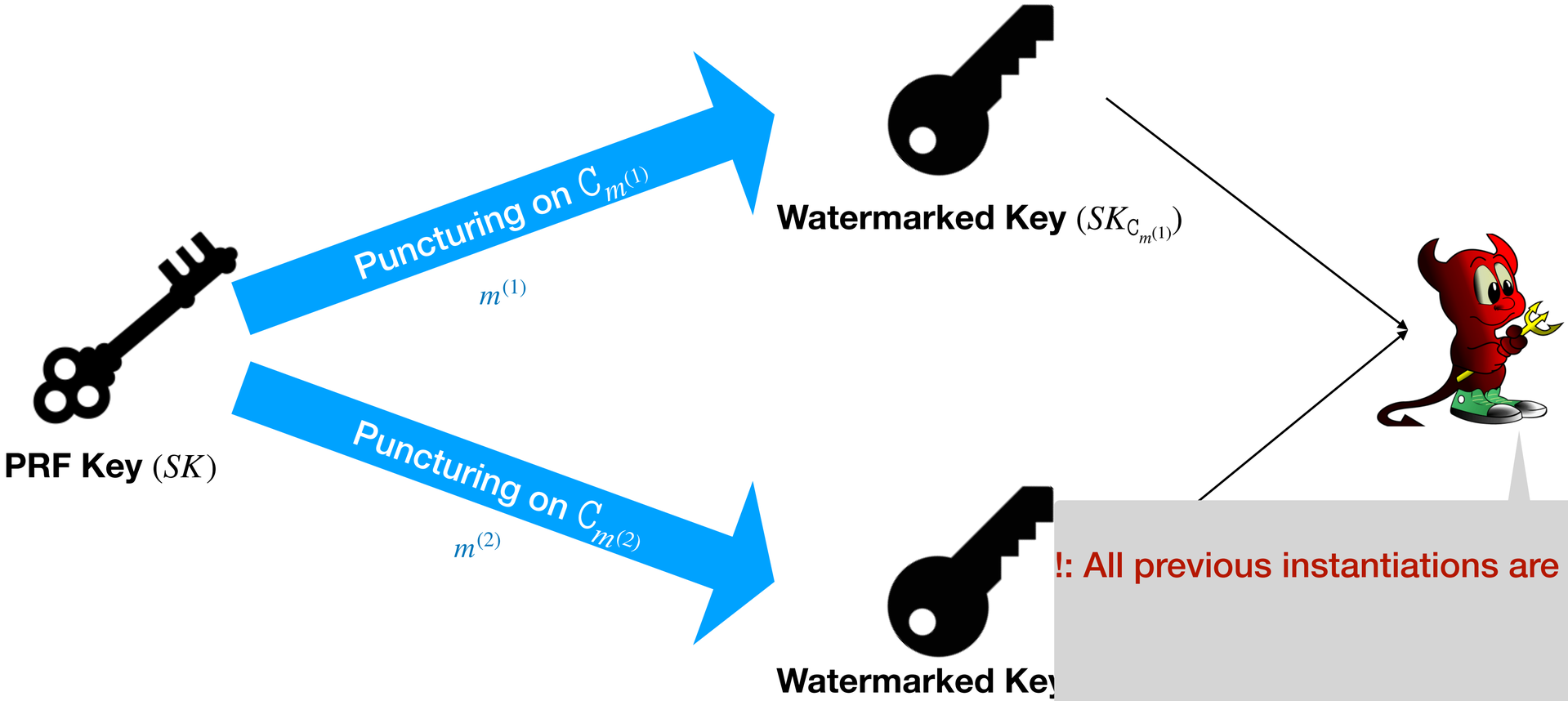
1. **Generate** $\begin{bmatrix} x^*_{1,0}, x^*_{2,0}, \ldots x^*_{n,0} \\ x^*_{1,1}, x^*_{2,1}, \ldots x^*_{n,1} \end{bmatrix}$
2. **For** $i \in [n]$, **recovering** m_i **by testing if the circuit is punctured on** $x^*_{i,0}$ **or** $x^*_{i,1}$

The Problem



$$m^{(1)} \neq m^{(2)} \implies C_{m^{(1)}} \neq C_{m^{(2)}} \implies SK_{C_{m^{(1)}}} \neq SK_{C_{m^{(2)}}}$$

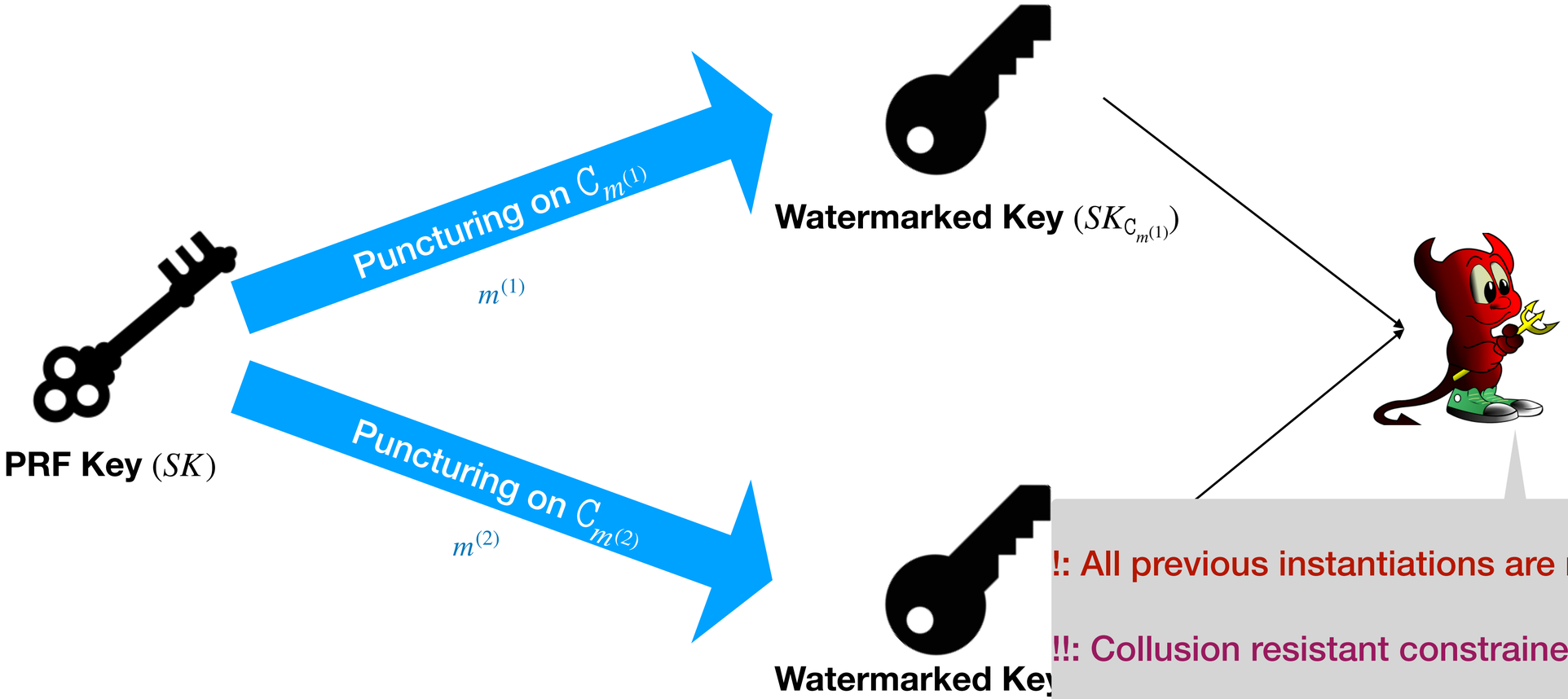
The Problem



!: All previous instantiations are not collusion resistant

$$m^{(1)} \neq m^{(2)} \implies C_{m^{(1)}} \neq C_{m^{(2)}} \implies SK_{C_{m^{(1)}}} \neq SK_{C_{m^{(2)}}}$$

The Problem

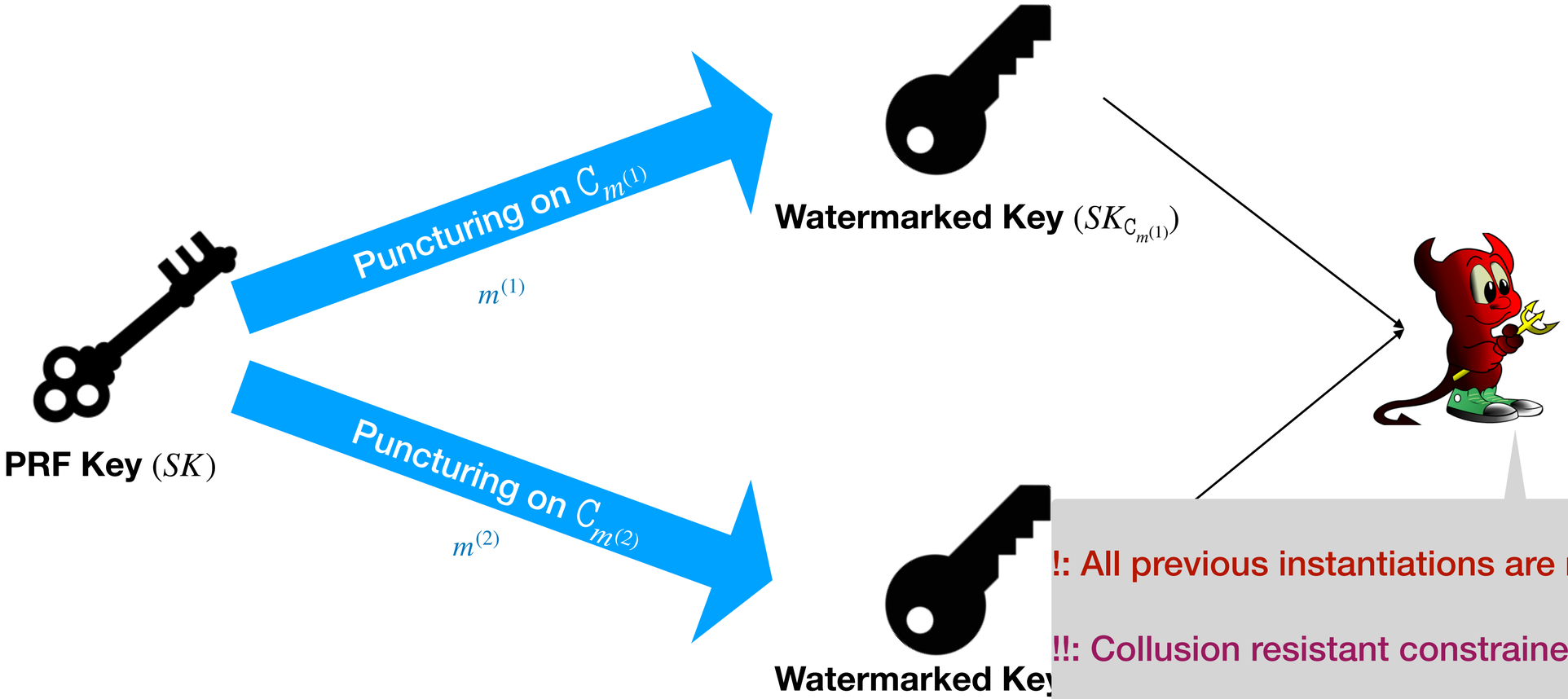


!: All previous instantiations are not collusion resistant

!!: Collusion resistant constrained PRF with privacy implies iO

$$m^{(1)} \neq m^{(2)} \implies C_{m^{(1)}} \neq C_{m^{(2)}} \implies SK_{C_{m^{(1)}}} \neq SK_{C_{m^{(2)}}}$$

The Problem



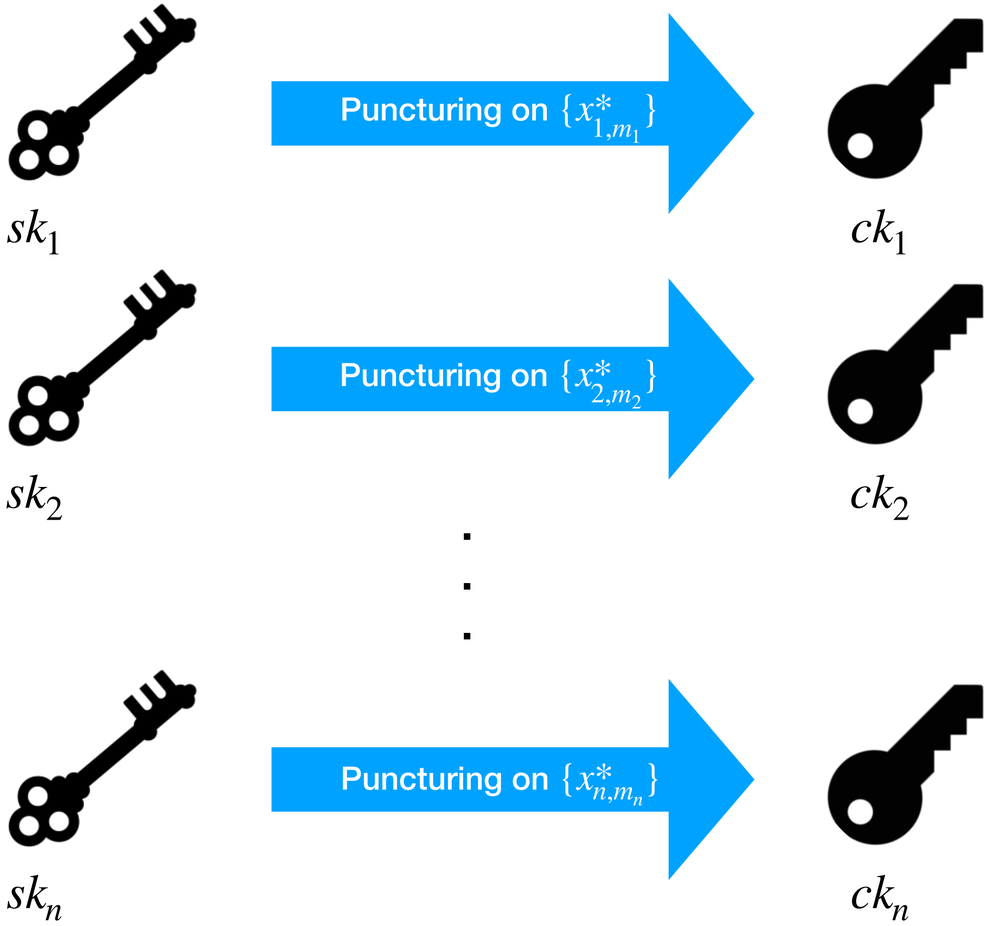
!: All previous instantiations are not collusion resistant

!!: Collusion resistant constrained PRF with privacy implies iO

We construct collusion resistant watermarkable PRF from single key secure (privately) constrained PRF

$$m^{(1)} \neq m^{(2)} \implies C_{m^{(1)}} \neq C_{m^{(2)}} \implies SK_{C_{m^{(1)}}} \neq SK_{C_{m^{(2)}}}$$

First Attempt



$m = (m_1, m_2, \ldots m_n) \in \{0,1\}^n$

Mark:

1. **Generate** $\begin{bmatrix} x^*_{1,0}, x^*_{2,0}, \ldots x^*_{n,0} \\ x^*_{1,1}, x^*_{2,1}, \ldots x^*_{n,1} \end{bmatrix}$
2. **Puncture** sk_i **on** x^*_{i,m_i}

Extract:

1. **Generate** $\begin{bmatrix} x^*_{1,0}, x^*_{2,0}, \ldots x^*_{n,0} \\ x^*_{1,1}, x^*_{2,1}, \ldots x^*_{n,1} \end{bmatrix}$
2. **For** $i \in [n]$, **recovering** m_i **by testing if** ck_i **is punctured on** $x^*_{i,0}$ **or** $x^*_{i,1}$

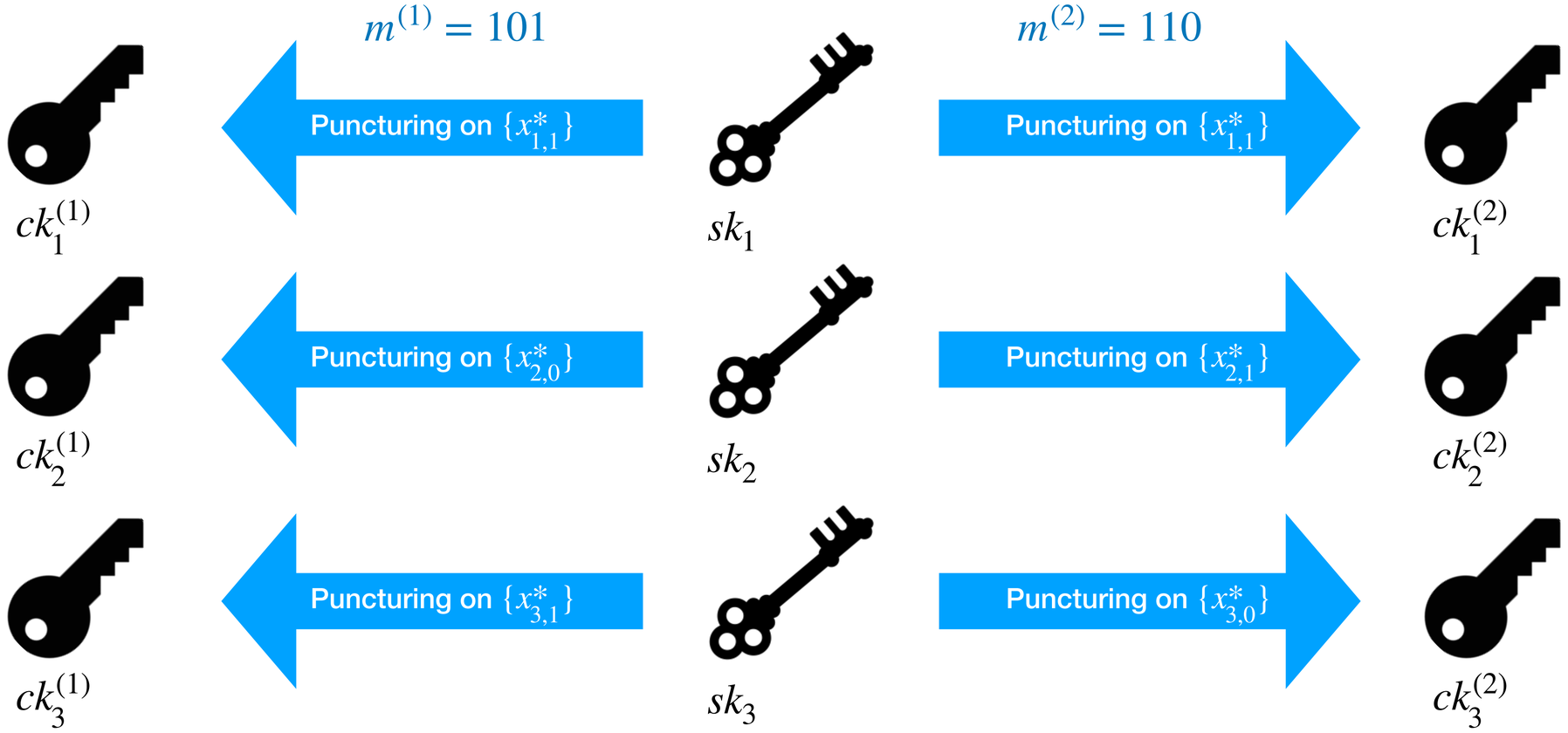
PRF Key $SK = (sk_1, sk_2, \ldots, sk_n)$

$F_{SK}(x) = F_{sk_1}(x) \| F_{sk_2}(x) \| \ldots F_{sk_n}(x)$

Watermarked Key $CK = (ck_1, ck_2, \ldots, ck_n)$

$F_{CK}(x) = F_{ck_1}(x) \| F_{ck_2}(x) \| \ldots F_{ck_n}(x)$

First Attempt

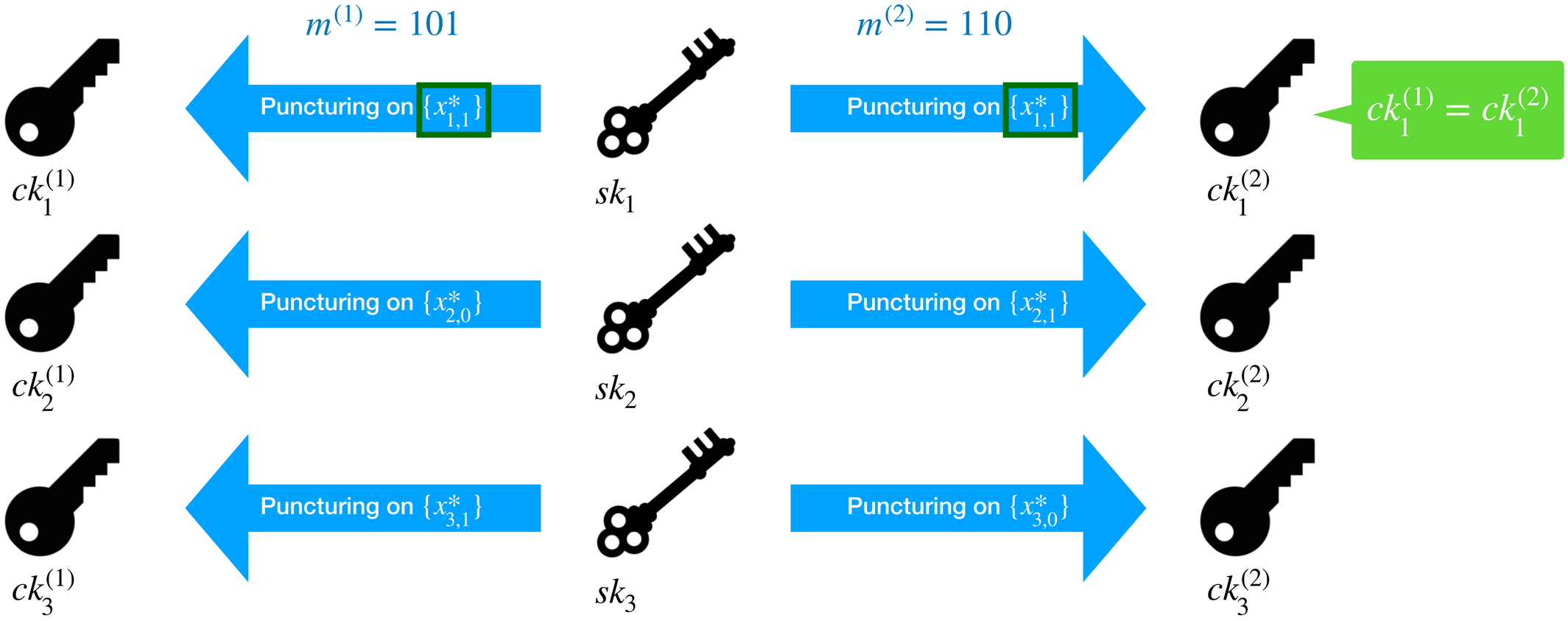


Watermarked Key $CK^{(1)} = (ck_1^{(1)}, ck_2^{(1)}, ck_3^{(1)})$

PRF Key $SK = (sk_1, sk_2, sk_3)$

Watermarked Key $CK^{(2)} = (ck_1^{(2)}, ck_2^{(2)}, ck_3^{(2)})$

First Attempt

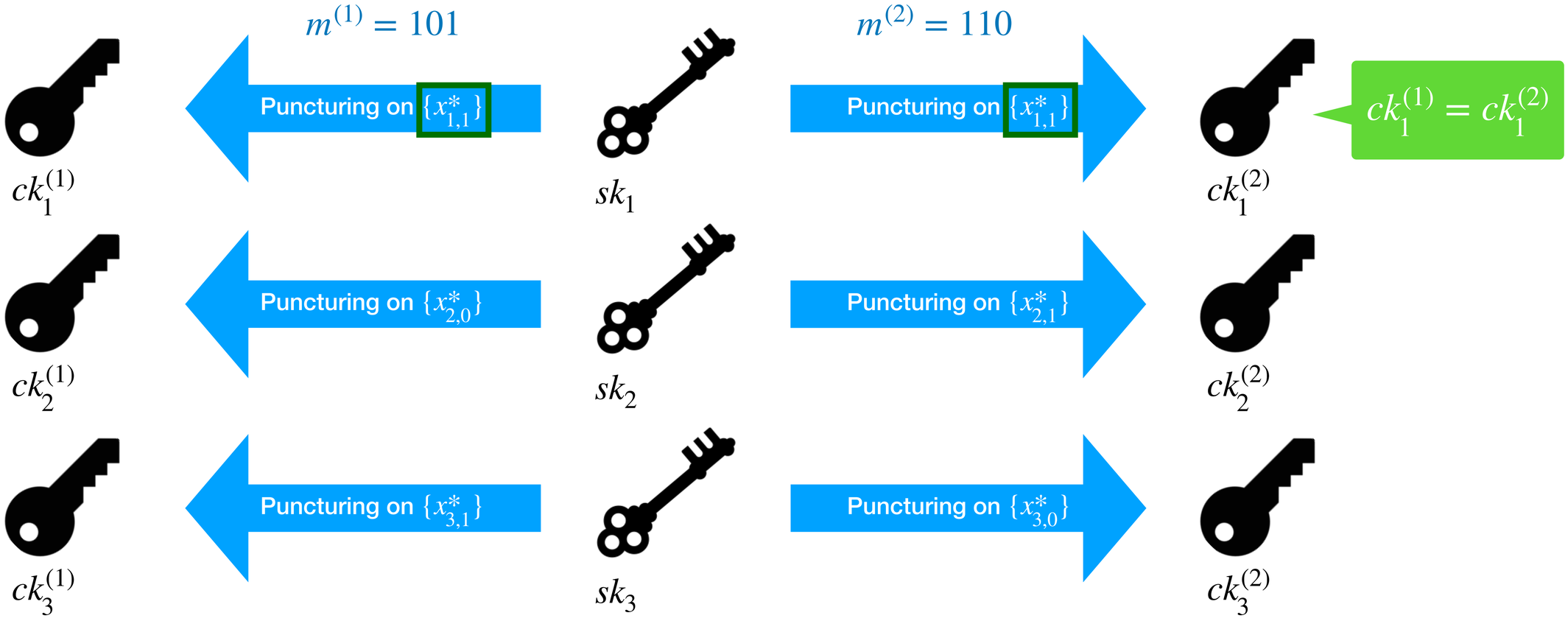


Watermarked Key $CK^{(1)} = (ck_1^{(1)}, ck_2^{(1)}, ck_3^{(1)})$

PRF Key $SK = (sk_1, sk_2, sk_3)$

Watermarked Key $CK^{(2)} = (ck_1^{(2)}, ck_2^{(2)}, ck_3^{(2)})$

First Attempt



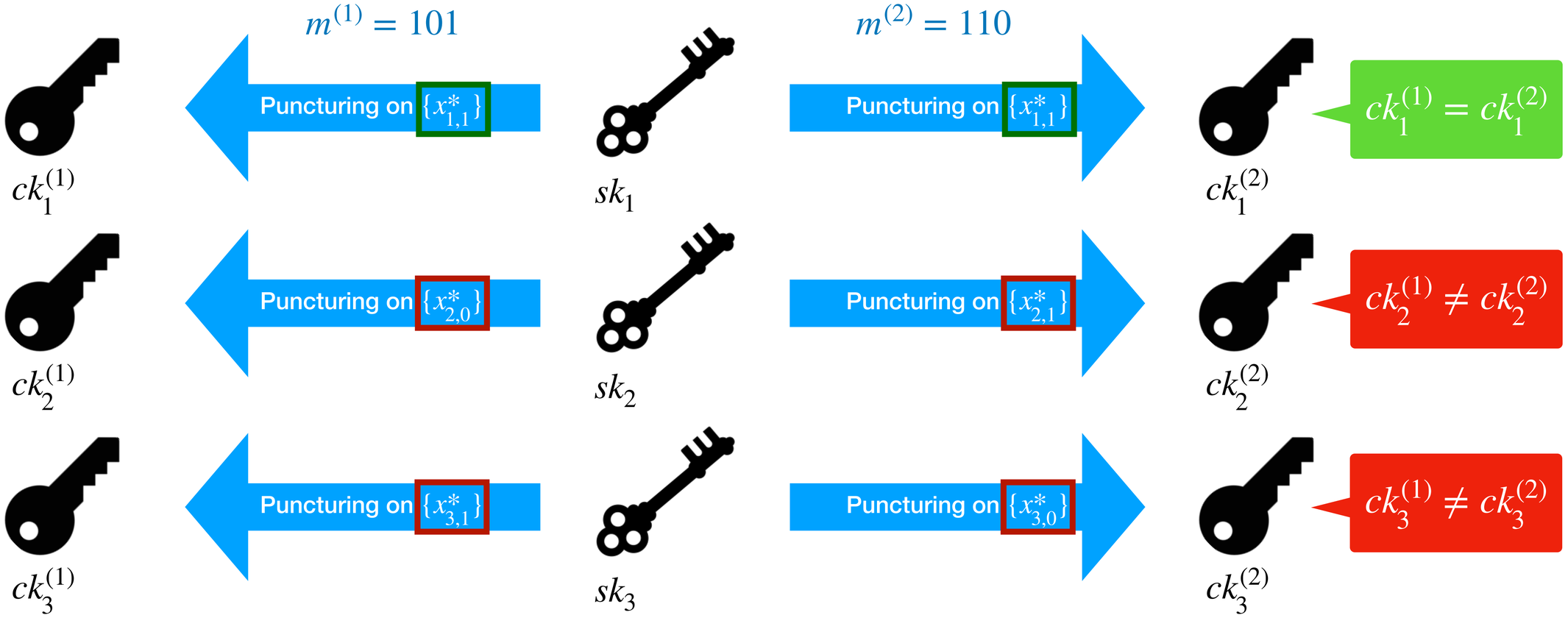
Watermarked Key $CK^{(1)} = (ck_1^{(1)}, ck_2^{(1)}, ck_3^{(1)})$

PRF Key $SK = (sk_1, sk_2, sk_3)$

Watermarked Key $CK^{(2)} = (ck_1^{(2)}, ck_2^{(2)}, ck_3^{(2)})$

✔: The bit embedded in sk_1 is secure

First Attempt



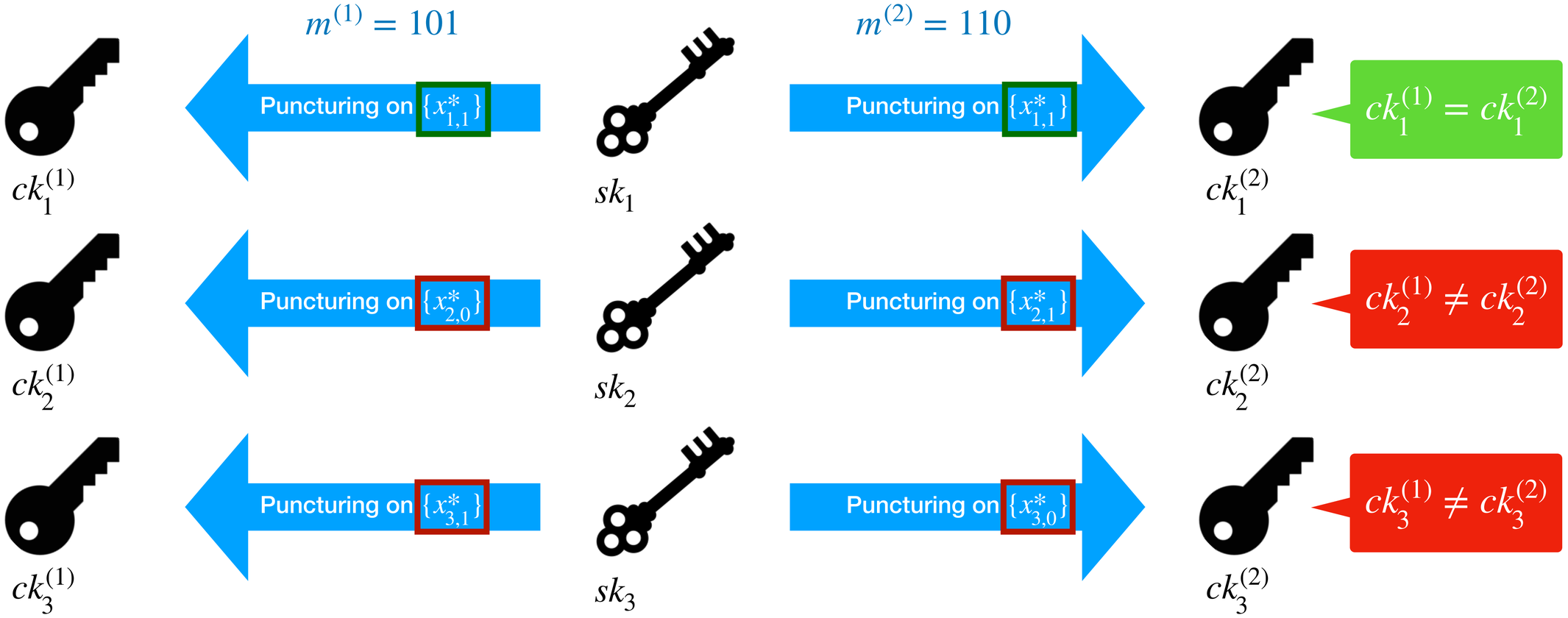
Watermarked Key $CK^{(1)} = (ck_1^{(1)}, ck_2^{(1)}, ck_3^{(1)})$

PRF Key $SK = (sk_1, sk_2, sk_3)$

Watermarked Key $CK^{(2)} = (ck_1^{(2)}, ck_2^{(2)}, ck_3^{(2)})$

✔: **The bit embedded in sk_1 is secure**

First Attempt



Watermarked Key $CK^{(1)} = (ck_1^{(1)}, ck_2^{(1)}, ck_3^{(1)})$

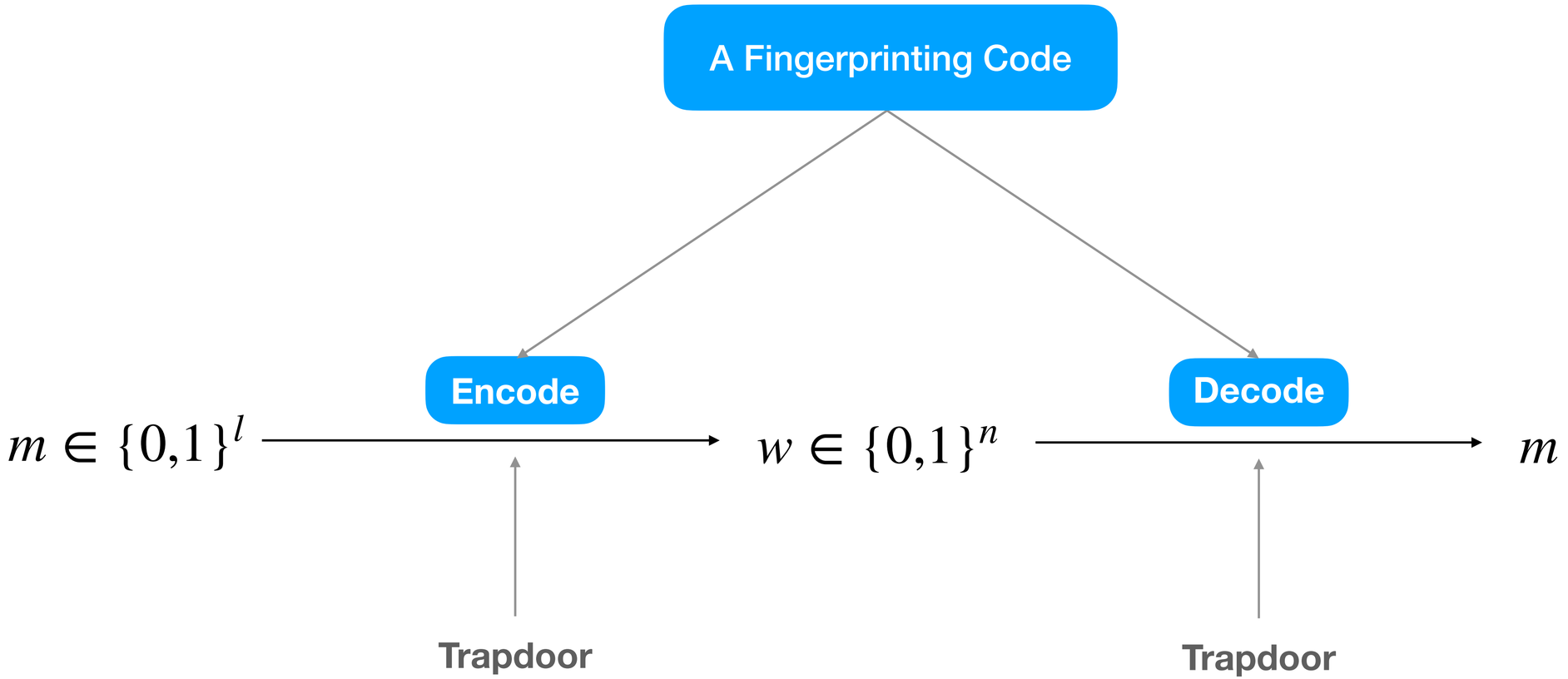
PRF Key $SK = (sk_1, sk_2, sk_3)$

Watermarked Key $CK^{(2)} = (ck_1^{(2)}, ck_2^{(2)}, ck_3^{(2)})$

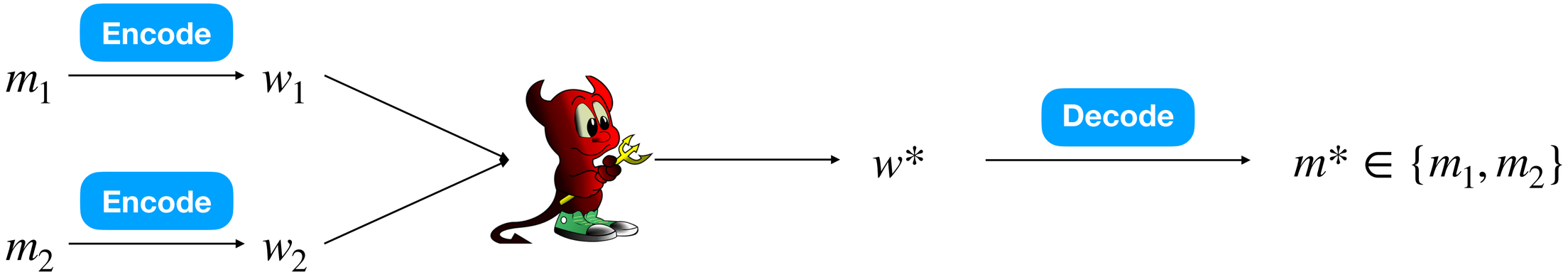
✔: The bit embedded in sk_1 is secure

✘: The adversary can still remove/modify bits embedded in sk_2 and sk_3

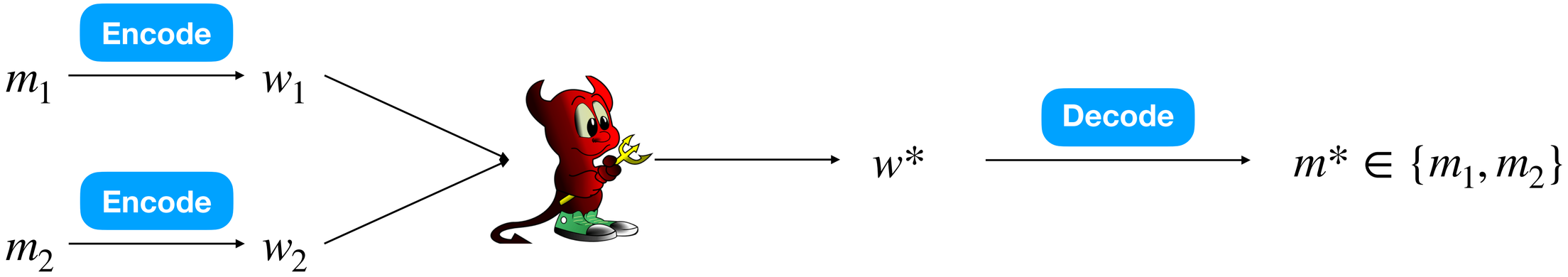
Fingerprinting Code



Fingerprinting Code



Fingerprinting Code

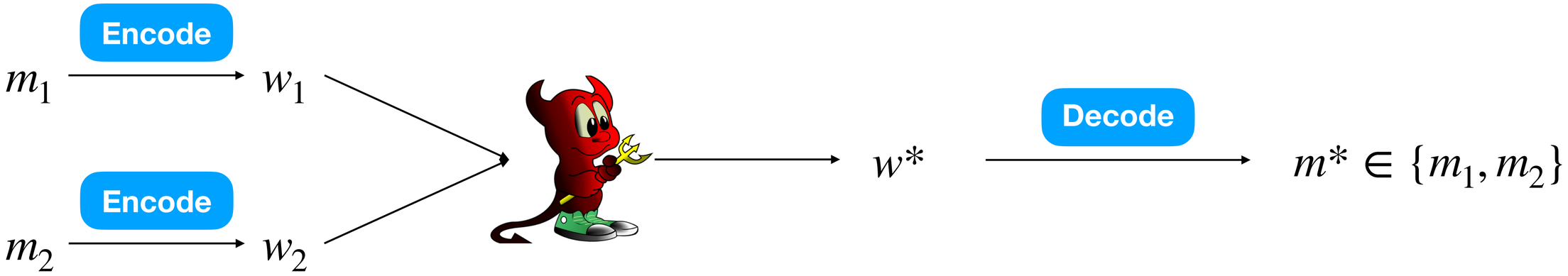


Marking Assumption:

For $i \in [n]$, if $w_1[i] = w_2[i]$:

$$w^*[i] = w_1[i] = w_2[i]$$

Fingerprinting Code



Marking Assumption:

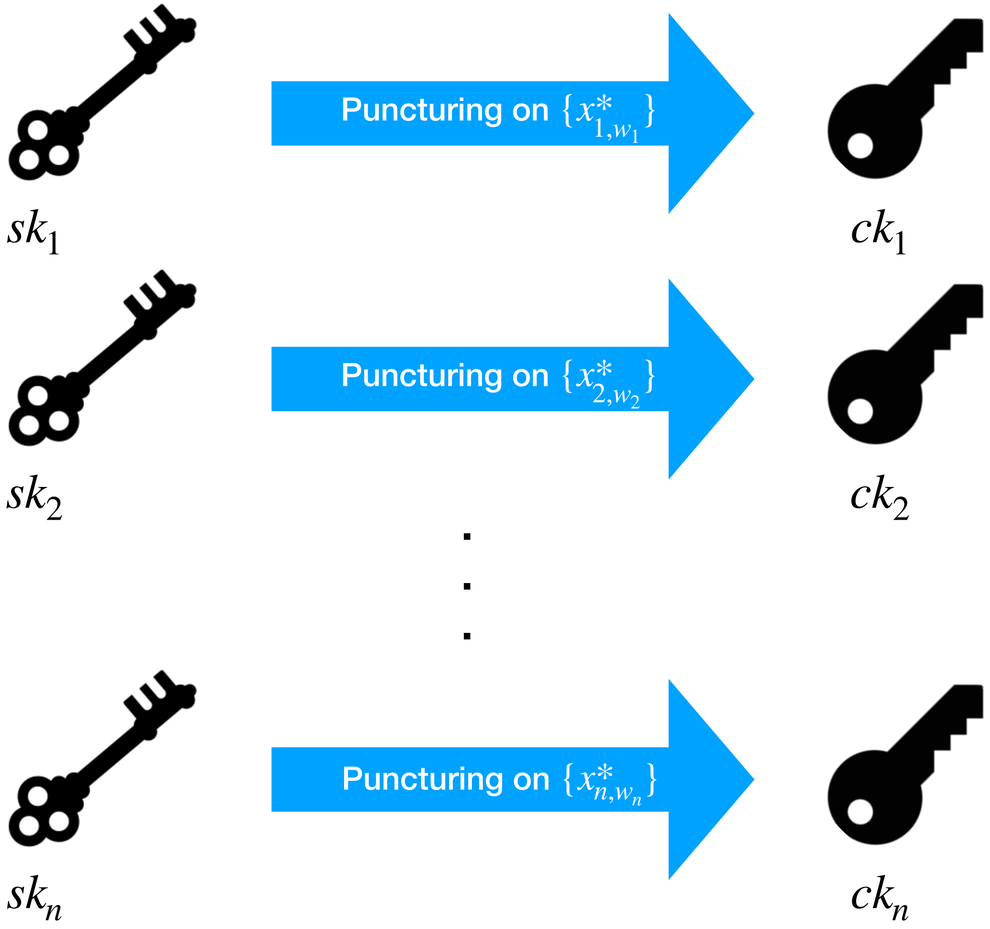
For $i \in [n]$, if $w_1[i] = w_2[i]$:

$$w^*[i] = w_1[i] = w_2[i]$$

E.G. If $w_1 = 101, w_2 = 110$,

$$w^*[1] = 1$$

Second Attempt



PRF Key $SK = (sk_1, sk_2, \ldots, sk_n)$

$F_{SK}(x) = F_{sk_1}(x) \| F_{sk_2}(x) \| \ldots F_{sk_n}(x)$

Watermarked Key $CK = (ck_1, ck_2, \ldots, ck_n)$

$F_{CK}(x) = F_{ck_1}(x) \| F_{ck_2}(x) \| \ldots F_{ck_n}(x)$

$m = (m_1, m_2, \ldots m_l) \in \{0,1\}^l$

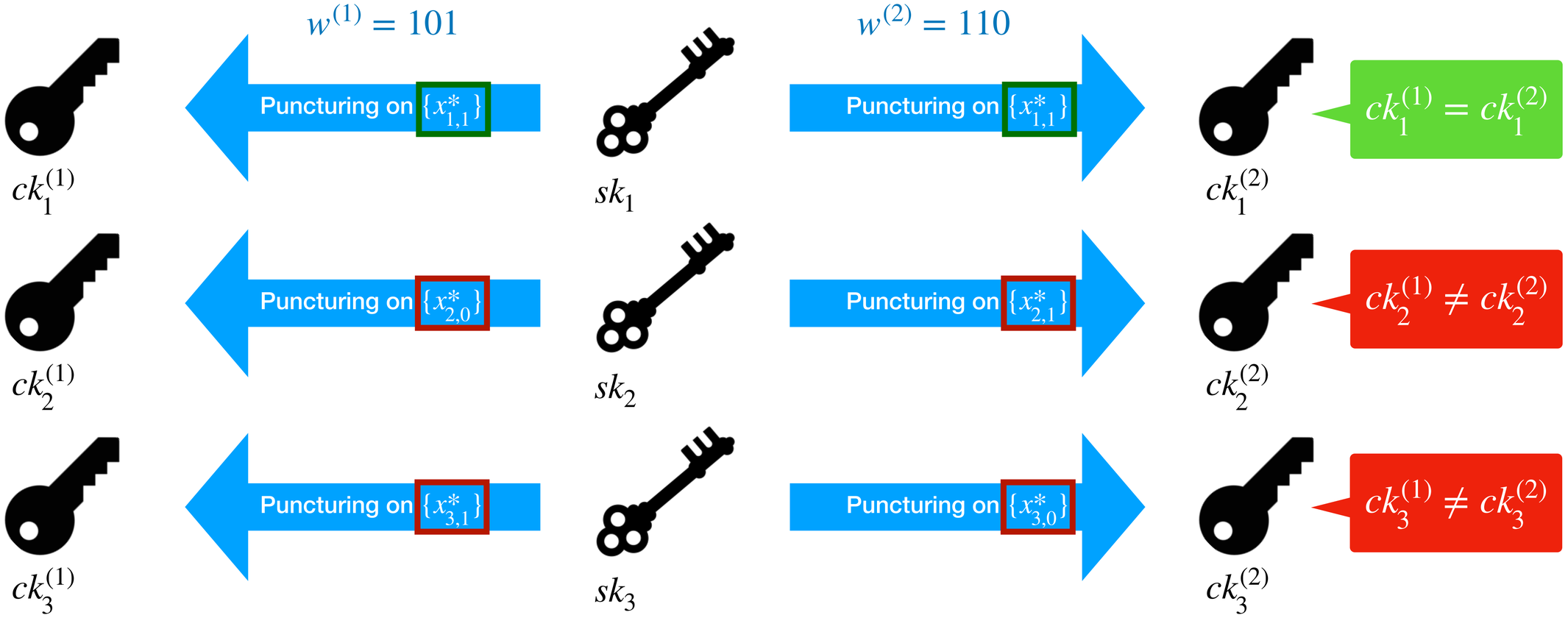
Mark:

1. $w = (w_1, w_2, \ldots, w_n) = \mathtt{Encode}(m)$
2. **Generate** $\begin{bmatrix} x^*_{1,0}, x^*_{2,0}, \ldots x^*_{n,0} \\ x^*_{1,1}, x^*_{2,1}, \ldots x^*_{n,1} \end{bmatrix}$
3. **Puncture** sk_i **on** x^*_{i,w_i}

Extract:

1. **Generate** $\begin{bmatrix} x^*_{1,0}, x^*_{2,0}, \ldots x^*_{n,0} \\ x^*_{1,1}, x^*_{2,1}, \ldots x^*_{n,1} \end{bmatrix}$
2. **For** $i \in [n]$, **recovering** w_i **by testing if** ck_i **is punctured on** $x^*_{i,0}$ **or** $x^*_{i,1}$
3. $m = \mathtt{Decode}(w)$

Second Attempt



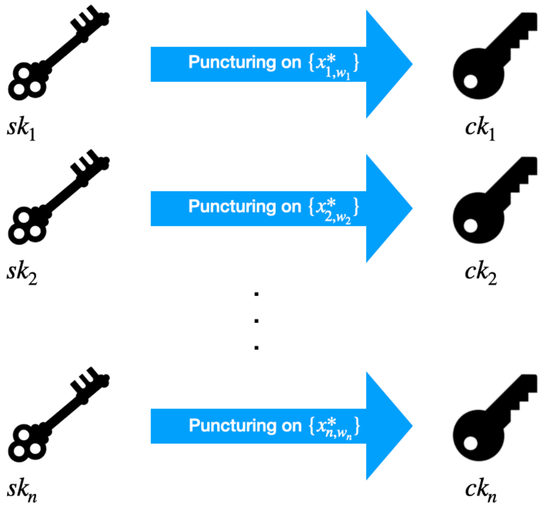
Watermarked Key $CK^{(1)} = (ck_1^{(1)}, ck_2^{(1)}, ck_3^{(1)})$

PRF Key $SK = (sk_1, sk_2, sk_3)$

Watermarked Key $CK^{(2)} = (ck_1^{(2)}, ck_2^{(2)}, ck_3^{(2)})$

The bit embedded in sk_1 is secure $\Longrightarrow$ Marking Assumption $\Longrightarrow$ Collusion Resistant Unremovability

Second Attempt

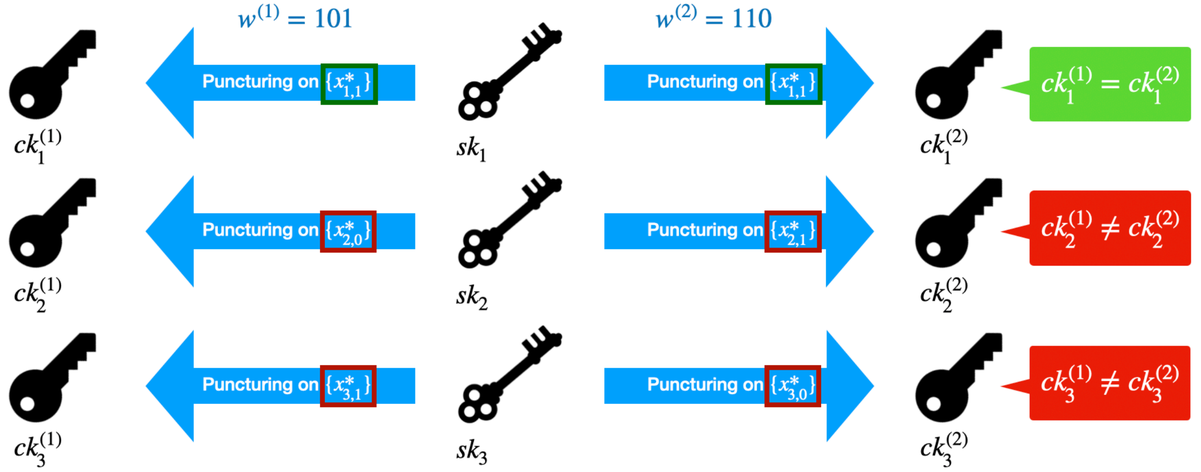


PRF Key $SK = (sk_1, sk_2, \ldots, sk_n)$

$F_{SK}(x) = F_{sk_1}(x) \| F_{sk_2}(x) \| \ldots F_{sk_n}(x)$

Watermarked Key $CK = (ck_1, ck_2, \ldots, ck_n)$

$F_{CK}(x) = F_{ck_1}(x) \| F_{ck_2}(x) \| \ldots F_{ck_n}(x)$



Watermarked Key $CK^{(1)} = (ck_1^{(1)}, ck_2^{(1)}, ck_3^{(1)})$

PRF Key $SK = (sk_1, sk_2, sk_3)$

Watermarked Key $CK^{(2)} = (ck_1^{(2)}, ck_2^{(2)}, ck_3^{(2)})$

The underlying constrained PRF Can be replaced with any single key secure watermarkable PRF

Second Attempt

Single-Key Secure Watermarkable PRF

Collusion Resistant Watermarkable PRF

The underlying constrained PRF Can be replaced with any single key secure watermarkable PRF

Extensions of Basic Solution

- Stronger Unremovability
  - Unremovability with Marking Queries/Public Marking
  - Unremovability with Extraction Queries
- Strong Unforgeability

Please see the paper (eprint.iacr.org/2020/695) for more details

Conclusion

Single-Key Secure Watermarkable PRF

Collusion Resistant Watermarkable PRF

Conclusion

Single-Key Secure Watermarkable PRF

Collusion Resistant Watermarkable PRF

	Unremovability	Unforgeability
[KW 17]	Marking Queries	Marking Queries
[QWZ 18]	Public Marking+ Extraction Queries	✖
[KW 19]	Public Marking+ Extraction Queries	✖

	Unremovability	Unforgeability
Scheme I	Marking Queries	Marking Queries+ Public Extraction
Scheme II	Public Marking+ Bounded Extraction Queries	Marking Queries+ Public Extraction
Scheme III	Public Marking+ Bounded Extraction Queries	Marking Queries+ Public Extraction

Open Problems

- Collusion Resistant Watermarkable PRF from Standard Assumptions with
  - Unremovability against Unbounded Extraction Queries
  - Exponential Message Spaces

Open Problems

- Collusion Resistant Watermarkable PRF from Standard Assumptions with
  - Unremovability against Unbounded Extraction Queries
  - Exponential Message Spaces

Thanks for your Attention!