Multi-key and Multi-input Predicate Encryption from Learning with Errors

Danilo Francati
Aarhus University

Daniele Friolo
Sapienza University of Rome

Giulio Malavolta
Max Planck Institute for Security and Privacy

Daniele Venturi
Sapienza University of Rome

https://eprint.iacr.org/2022/806

Overview

Single-input Setting

Predicate Encryption

↑

Functional encryption

Overview

Single-input Setting

Multi-input Setting

Predicate Encryption

Functional encryption

Multi-input Functional Encryption

Overview

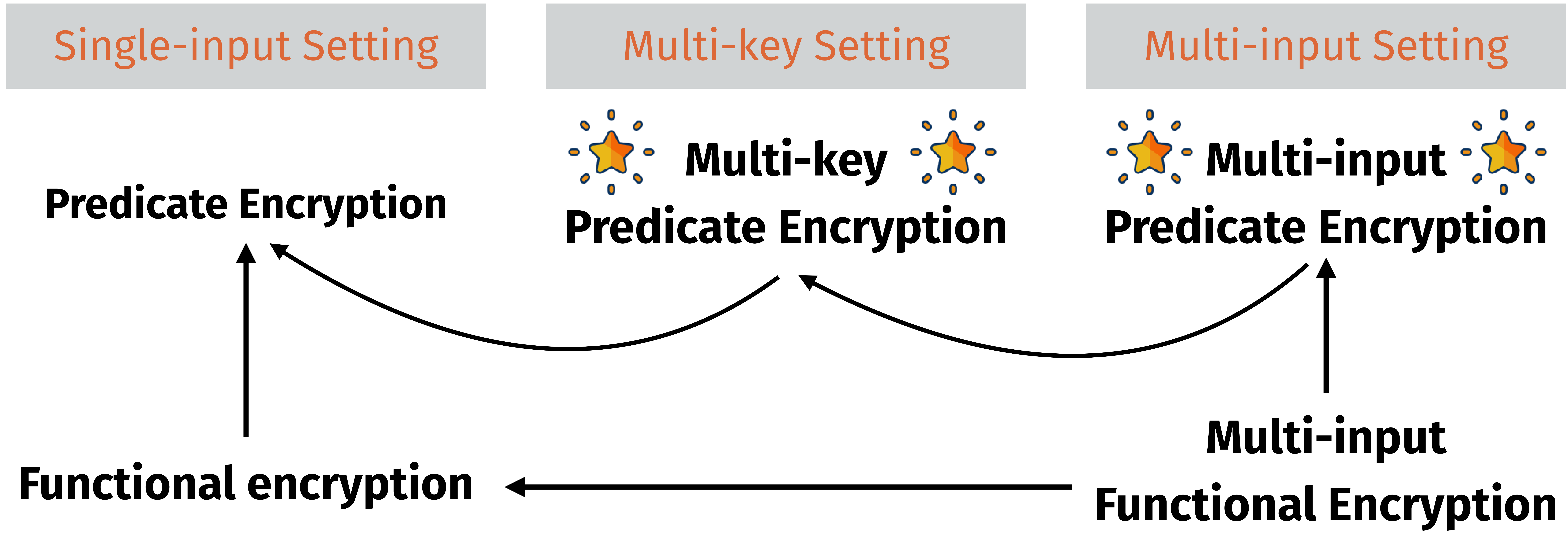
Single-input Setting | **Multi-input Setting**

Predicate Encryption ← **Multi-input Predicate Encryption**

↑ | ↑

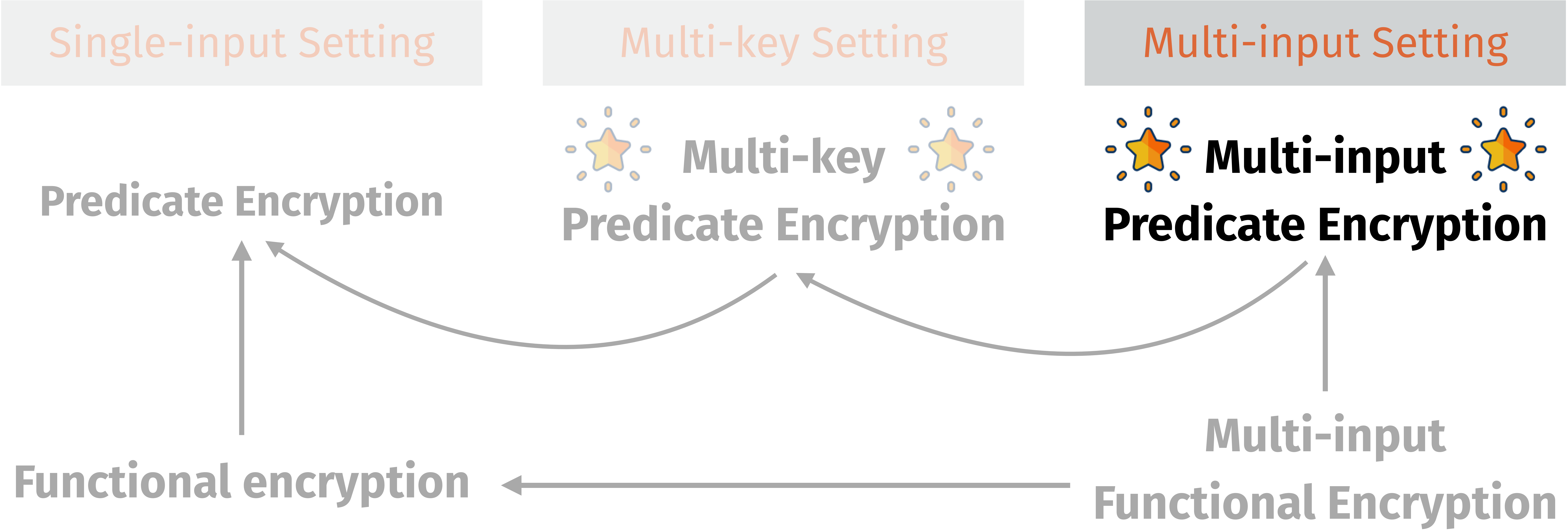
Functional encryption ← **Multi-input Functional Encryption**

Overview



Single-input Setting
Multi-key Setting
Multi-input Setting
Predicate Encryption
Multi-key
Predicate Encryption
Multi-input
Predicate Encryption
Functional encryption
Multi-input
Functional Encryption

Overview



[AYY22]. Agrawal, Shweta, Anshu Yadav, and Shota Yamada. "Multi-input attribute based encryption and predicate encryption." (CRYPTO 22)

Predicate Encryption (PE)

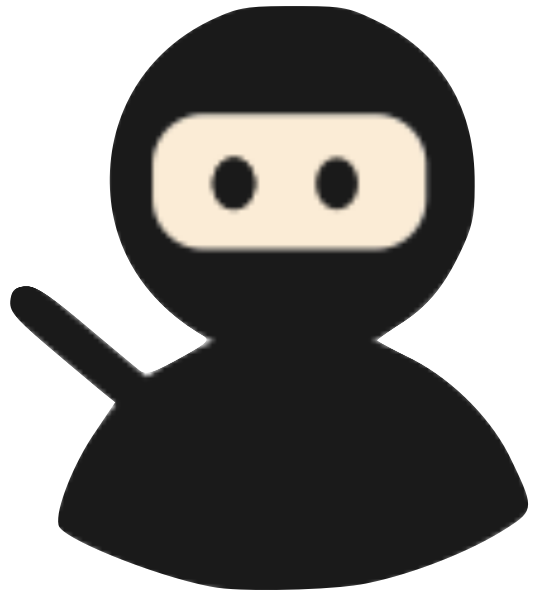
Predicate Encryption (PE)

Authority

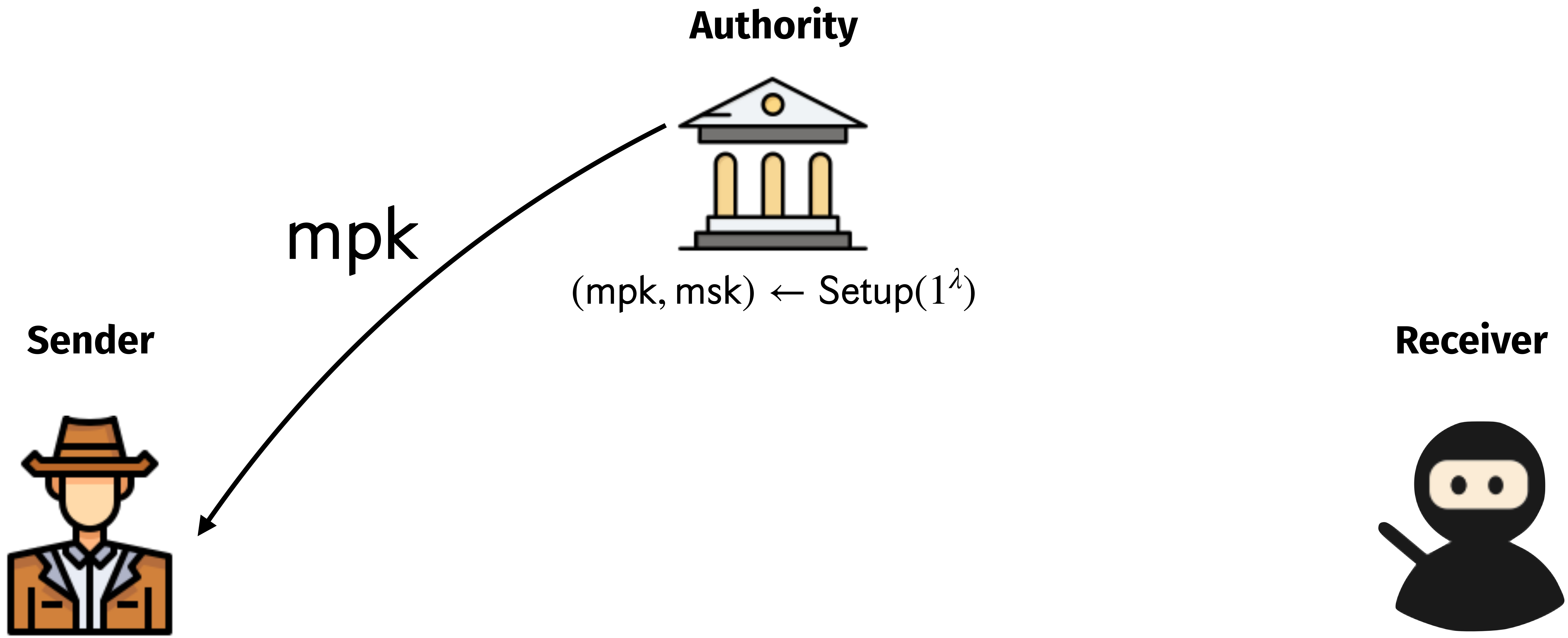
$$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$$

Sender

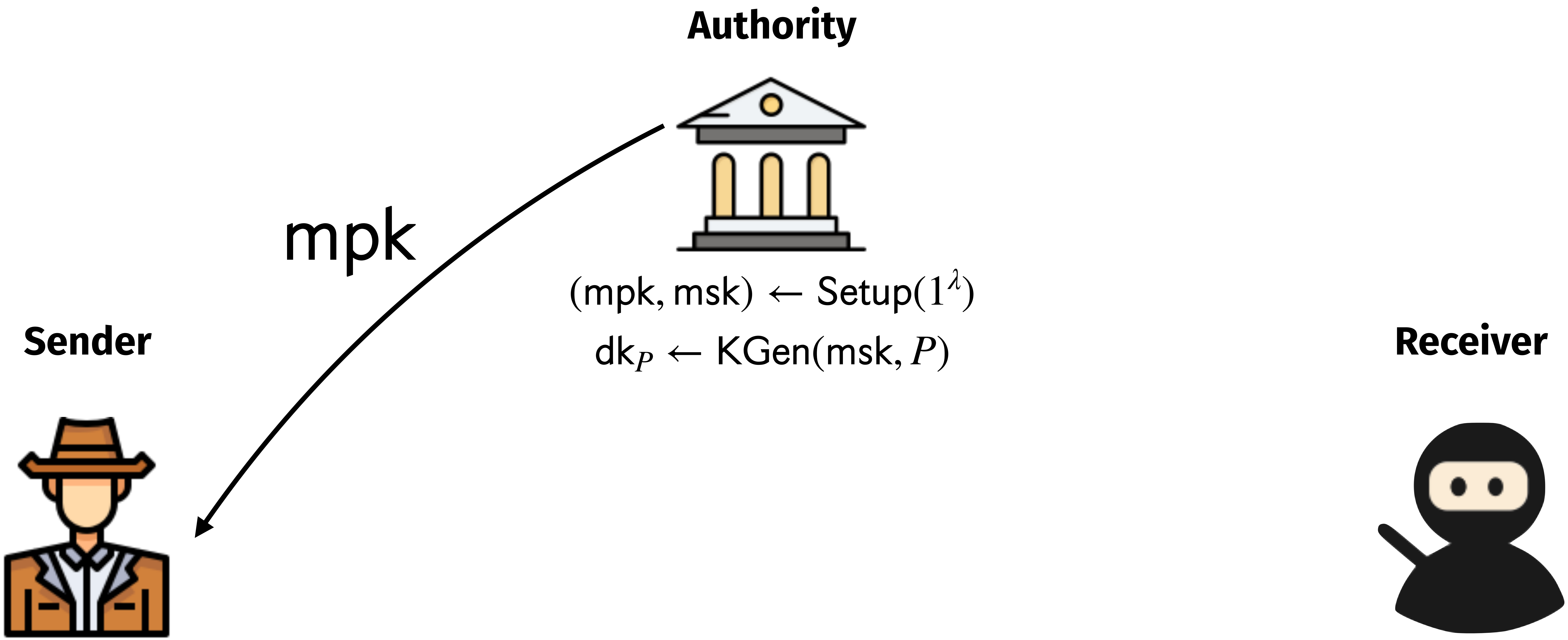
Receiver



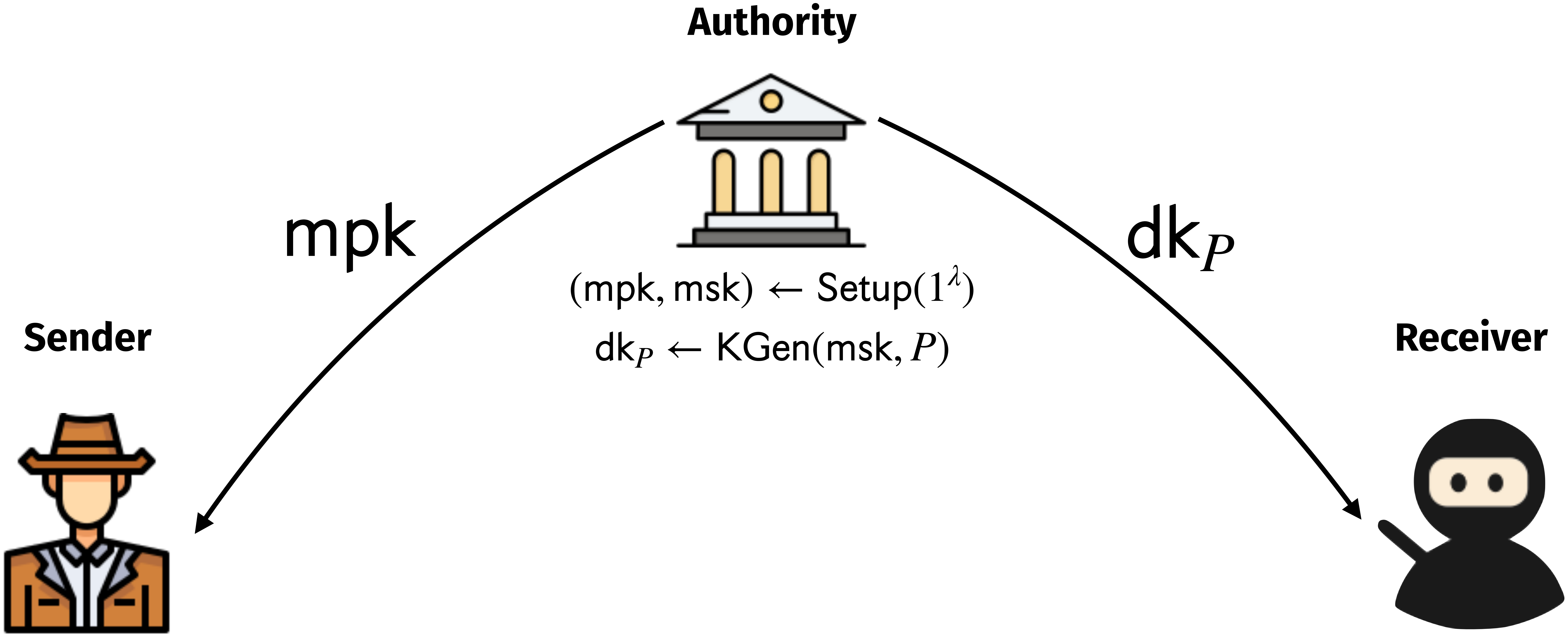
Predicate Encryption (PE)



Predicate Encryption (PE)



Predicate Encryption (PE)



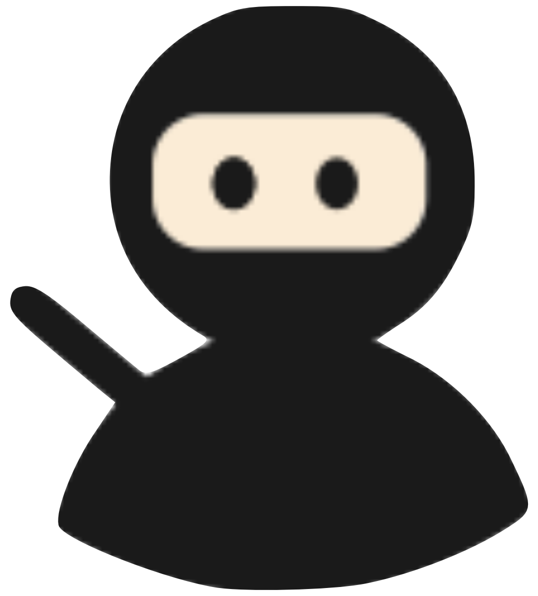
Predicate Encryption (PE)

Authority

Sender

mpk

Receiver



dk_P

Predicate Encryption (PE)

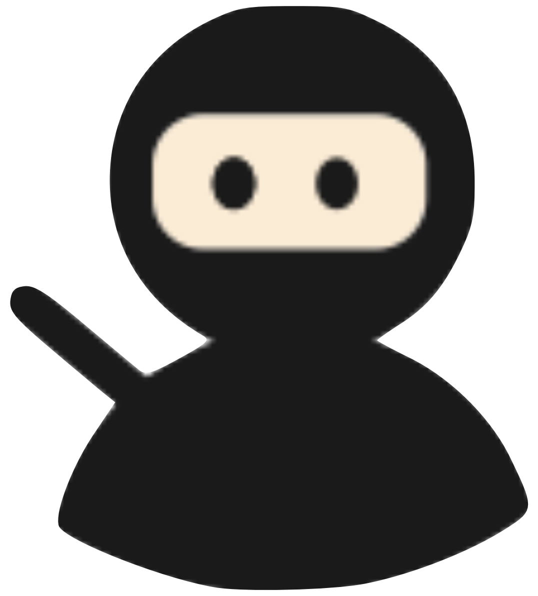
Authority

Sender

Receiver

c

mpk



dk_P

$c \leftarrow \mathsf{Enc}(\mathsf{mpk}, x, m)$

Predicate Encryption (PE)

Predicate Encryption (PE)

CPA-1-sided security of PE

$(m^0, m^1, x^0, x^1 \alpha) \leftarrow \mathsf{A_1}^{\mathsf{KGen}}(1^\lambda, \mathsf{mpk})$

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

$c \leftarrow \mathsf{Enc}(\mathsf{mpk}, x^b, m^b)$

$b' \leftarrow \mathsf{A_2}^{\mathsf{KGen}}(c, \alpha)$

WIN if $b = b'$

CPA-1-sided security of PE

$(m^0, m^1, x^0, x^1 \alpha) \leftarrow \mathsf{A}_1{}^{\mathsf{KGen}}(1^\lambda, \mathsf{mpk})$

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

$c \leftarrow \mathsf{Enc}(\mathsf{mpk}, x^b, m^b)$

$b' \leftarrow \mathsf{A}_2{}^{\mathsf{KGen}}(c, \alpha)$

WIN if $b = b'$

CPA-1-sided security of PE

$\Pr[\textbf{WIN}] \leq \mathsf{negl}(\lambda)$, and $\forall \mathsf{dk}_P$ we have

$$P(x^0) = P(x^1) = 0$$

Multi-input PE (2-input case)

Multi-input PE (2-input case)

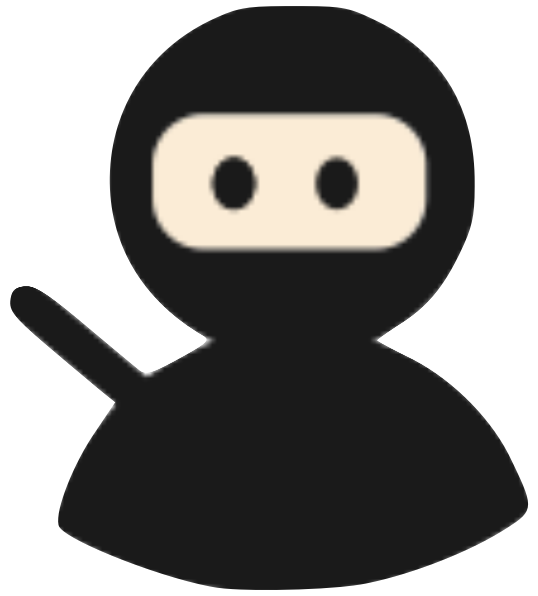
Authority

$(\mathsf{ek}_1, \mathsf{ek}_2, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$

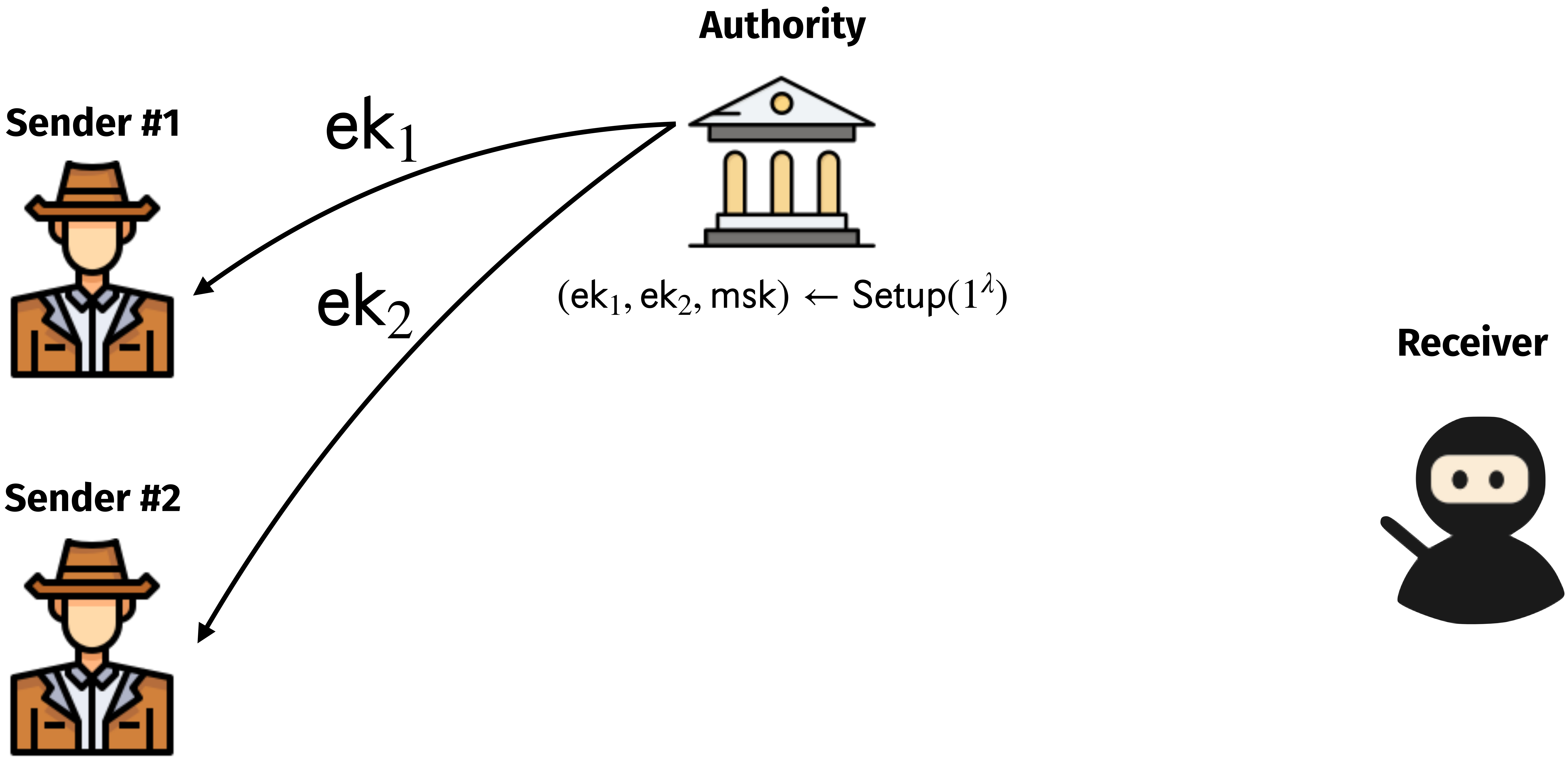
Sender #1

Sender #2

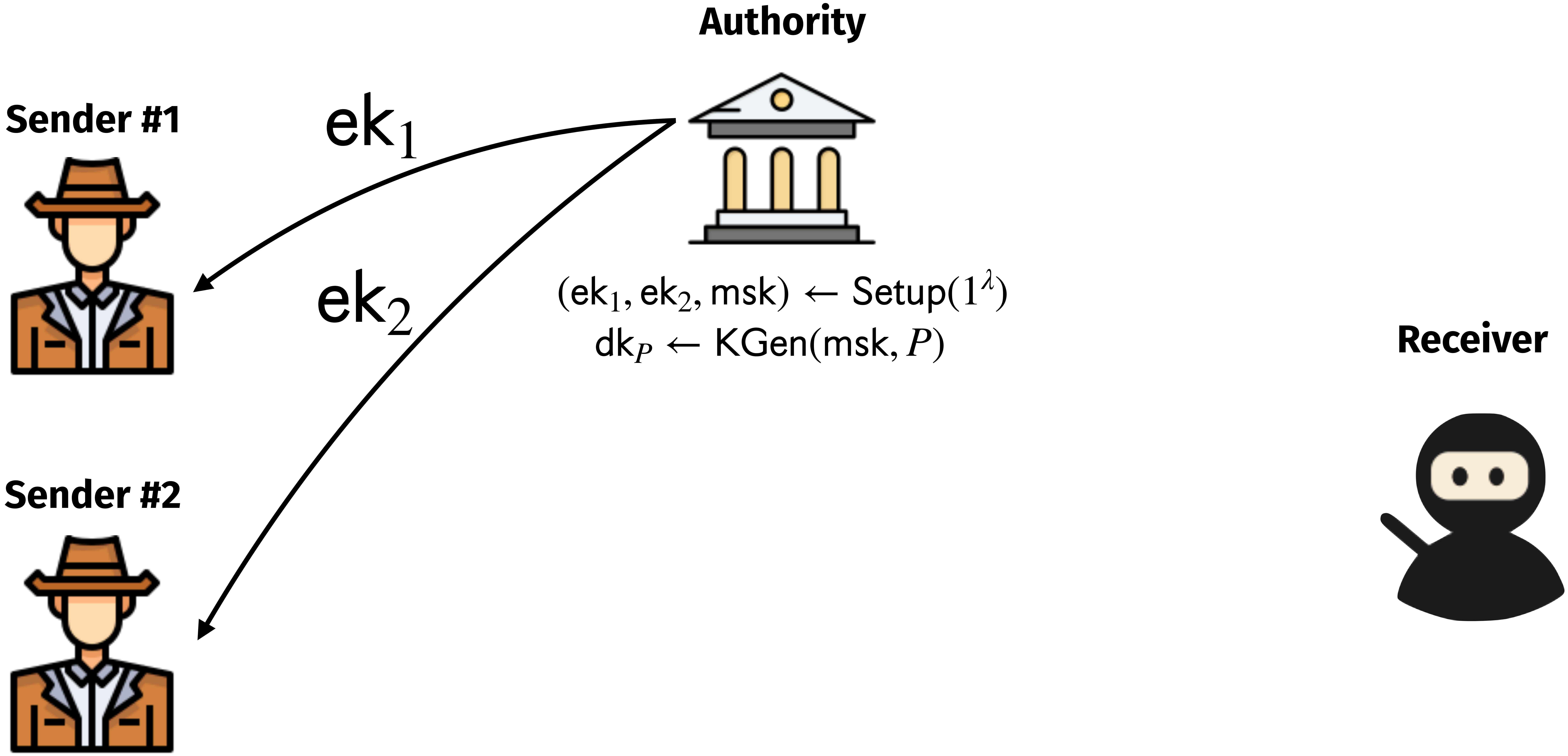
Receiver



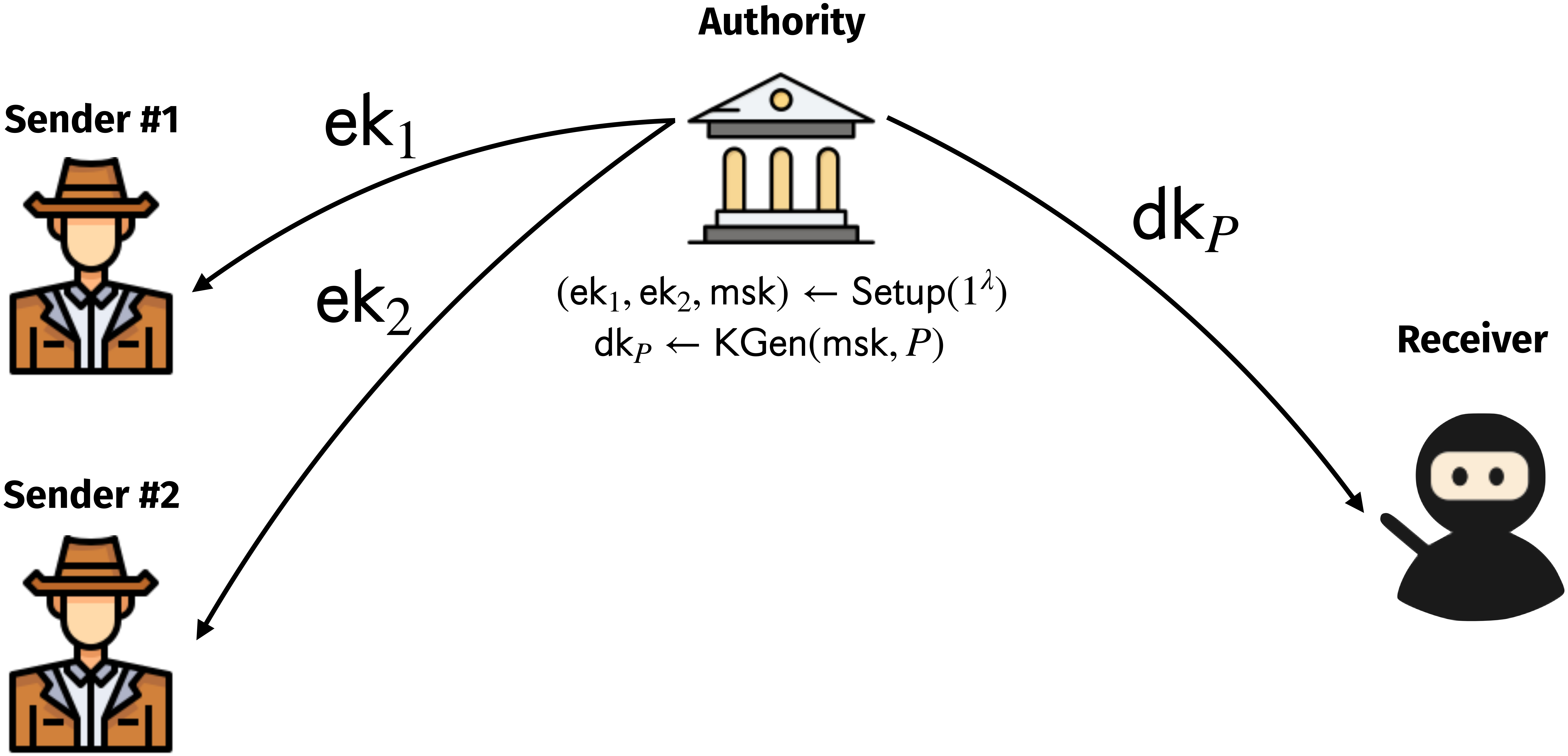
Multi-input PE (2-input case)



Multi-input PE (2-input case)



Multi-input PE (2-input case)



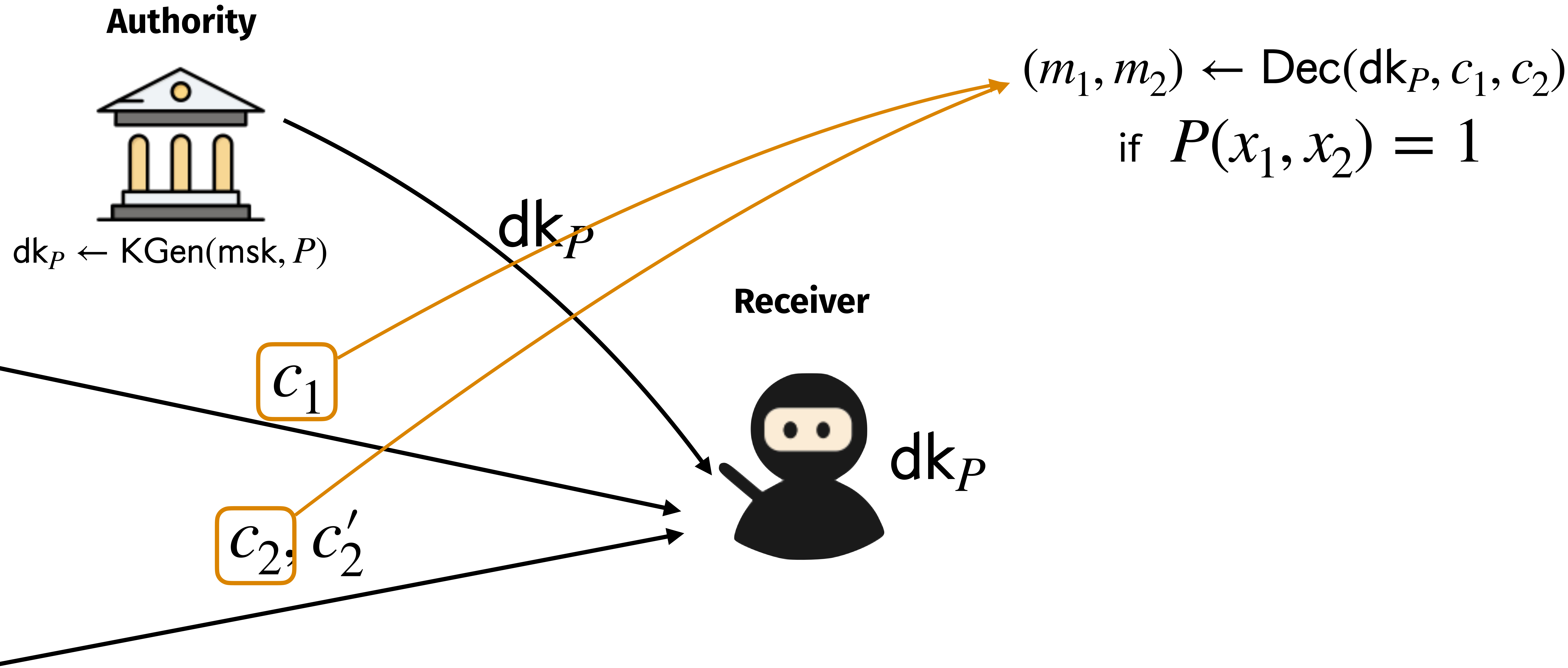
Multi-input PE (2-input case)

Multi-input PE (2-input case)

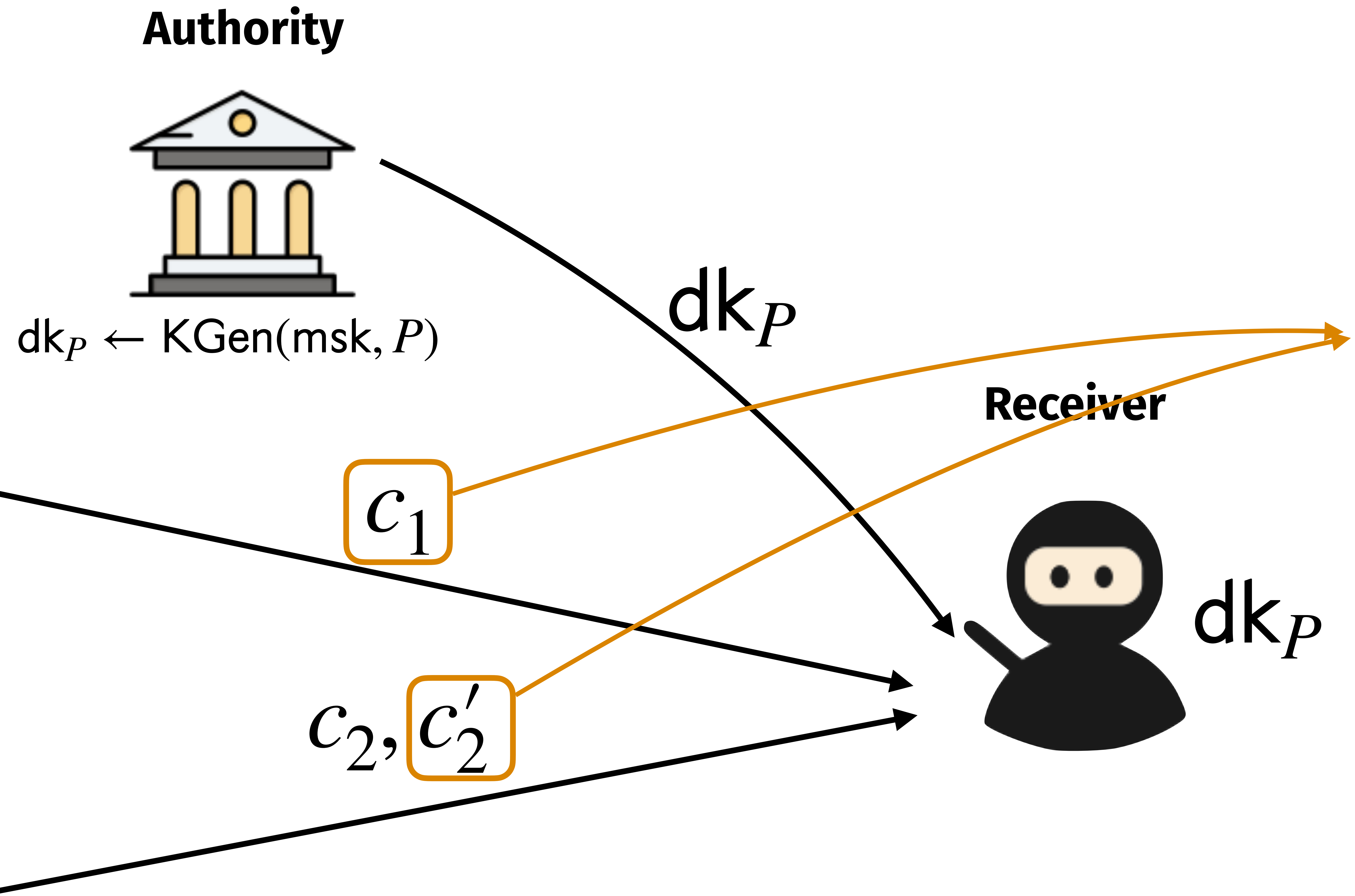
Correctness

$$(m_1, m_2) = \mathsf{Dec}(\mathsf{dk}_P, c_1, c_2) \text{ if } P(x_1, x_2) = 1$$

Multi-input PE (2-input case)



Multi-input PE (2-input case)



$(m_1, m_2) \leftarrow \mathsf{Dec}(\mathsf{dk}_P, c_1, c_2)$

if $P(x_1, x_2) = 1$

$(m_1, m_2') \leftarrow \mathsf{Dec}(\mathsf{dk}_P, c_1, c_2')$

if $P(x_1, x_2') = 1$

CPA-1-sided security of secret-key multi-input PE

$$((m_i^0)_{i\in[n]}, (m_i^1)_{i\in[n]}, (x_i^0)_{i\in[n]}, (x_i^1)_{i\in[n]}\alpha) \leftarrow \mathsf{A}_1^{\mathsf{Enc}(\mathsf{ek}_1,\cdot),\ldots,\mathsf{Enc}(\mathsf{ek}_n,\cdot),\mathsf{KGen}(\mathsf{msk},\cdot)}(1^\lambda, \mathsf{mpk})$$

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

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1^b, m_1^b), \ldots, c_n \leftarrow \mathsf{Enc}(\mathsf{ek}_n, x_n^b, m_n^b)$

$$b' \leftarrow \mathsf{A}_2^{\mathsf{Enc}(\mathsf{ek}_1,\cdot),\ldots,\mathsf{Enc}(\mathsf{ek}_n,\cdot),\mathsf{KGen}(\mathsf{msk},\cdot)}(c_1, \ldots, c_n, \alpha)$$

WIN if $b = b'$

CPA-1-sided security of secret-key multi-input PE

$$((m_i^0)_{i\in[n]}, (m_i^1)_{i\in[n]}, (x_i^0)_{i\in[n]}, (x_i^1)_{i\in[n]}\alpha) \leftarrow \mathsf{A}_1^{\mathsf{Enc}(\mathsf{ek}_1,\cdot),\ldots,\mathsf{Enc}(\mathsf{ek}_n,\cdot),\mathsf{KGen}(\mathsf{msk},\cdot)}(1^\lambda, \mathsf{mpk})$$

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

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1^b, m_1^b), \ldots, c_n \leftarrow \mathsf{Enc}(\mathsf{ek}_n, x_n^b, m_n^b)$

$$b' \leftarrow \mathsf{A}_2^{\mathsf{Enc}(\mathsf{ek}_1,\cdot),\ldots,\mathsf{Enc}(\mathsf{ek}_n,\cdot),\mathsf{KGen}(\mathsf{msk},\cdot)}(c_1, \ldots, c_n, \alpha)$$

WIN if $b = b'$

CPA-1-sided security of Multi-input PE

$\Pr[$**WIN**$] \leq \mathsf{negl}(\lambda)$, and $\forall \mathsf{dk}_P$ we have

$$\forall i \in [n]\ \ P(\ldots, x_i^0, \ldots) = P(\ldots, x_i^1, \ldots) = 0$$

CPA-1-sided security of <u>secret-key</u> multi-input PE

$$((m_i^0)_{i\in[n]}, (m_i^1)_{i\in[n]}, (x_i^0)_{i\in[n]}, (x_i^1)_{i\in[n]}\alpha) \leftarrow \mathsf{A}_1^{\mathsf{Enc}(\mathsf{ek}_1,\cdot),\ldots,\mathsf{Enc}(\mathsf{ek}_n,\cdot),\mathsf{KGen}(\mathsf{msk},\cdot)}(1^\lambda, \mathsf{mpk})$$

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

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1^b, m_1^b), \ldots, c_n \leftarrow \mathsf{Enc}(\mathsf{ek}_n, x_n^b, m_n^b)$

$$b' \leftarrow \mathsf{A}_2^{\mathsf{Enc}(\mathsf{ek}_1,\cdot),\ldots,\mathsf{Enc}(\mathsf{ek}_n,\cdot),\mathsf{KGen}(\mathsf{msk},\cdot)}(c_1, \ldots, c_n, \alpha)$$

WIN if $b = b'$

CPA-1-sided security of Multi-input PE

$\Pr[\textbf{WIN}] \leq \mathsf{negl}(\lambda)$, and $\forall \mathsf{dk}_P$ we have

$$\forall i \in [n] \;\; P(\ldots, x_i^0, \ldots) = P(\ldots, x_i^1, \ldots) = 0$$

CPA-1-sided security of secret-key multi-input PE

$$((m_i^0)_{i\in[n]}, (m_i^1)_{i\in[n]}, (x_i^0)_{i\in[n]}, (x_i^1)_{i\in[n]}\alpha) \leftarrow \mathsf{A}_1^{\mathsf{Enc}(\mathsf{ek}_1,\cdot),\ldots,\mathsf{Enc}(\mathsf{ek}_n,\cdot),\mathsf{KGen}(\mathsf{msk},\cdot)}(1^\lambda, \mathsf{mpk})$$

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

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1^b, m_1^b), \ldots, c_n \leftarrow \mathsf{Enc}(\mathsf{ek}_n, x_n^b, m_n^b)$

$$b' \leftarrow \mathsf{A}_2^{\mathsf{Enc}(\mathsf{ek}_1,\cdot),\ldots,\mathsf{Enc}(\mathsf{ek}_n,\cdot),\mathsf{KGen}(\mathsf{msk},\cdot)}(c_1, \ldots, c_n, \alpha)$$

WIN if $b = b'$

CPA-1-sided security of Multi-input PE

$\Pr[\textbf{WIN}] \leq \mathsf{negl}(\lambda)$, and $\forall \mathsf{dk}_P$ we have

$$\forall i \in [n] \;\; P(\ldots, x_i^0, \ldots) = P(\ldots, x_i^1, \ldots) = 0$$

CPA-1-sided security of secret-key multi-input PE

$((m_i^0)_{i\in[n]}, (m_i^1)_{i\in[n]}, (x_i^0)_{i\in[n]}, (x_i^1)_{i\in[n]}\alpha) \leftarrow \mathsf{A}_1^{\mathsf{Enc}(\mathsf{ek}_1,\cdot),\ldots,\mathsf{Enc}(\mathsf{ek}_n,\cdot),\mathsf{KGen}(\mathsf{msk},\cdot)}(1^\lambda, \mathsf{mpk})$

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

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1^b, m_1^b), \ldots, c_n \leftarrow \mathsf{Enc}(\mathsf{ek}_n, x_n^b, m_n^b)$

$b' \leftarrow \mathsf{A}_2^{\mathsf{Enc}(\mathsf{ek}_1,\cdot),\ldots,\mathsf{Enc}(\mathsf{ek}_n,\cdot),\mathsf{KGen}(\mathsf{msk},\cdot)}(c_1, \ldots, c_n, \alpha)$

WIN if $b = b'$

CPA-1-sided security of Multi-input PE

$\Pr[\textbf{WIN}] \leq \mathsf{negl}(\lambda)$, and $\forall \mathsf{dk}_P$ we have

$\forall i \in [n] \;\; P(\ldots, x_i^0, \ldots) = P(\ldots, x_i^1, \ldots) = 0$

Main Result

(Informal) Theorem

Adaptively CPA-1-sided secure **secret-key** n-input PE from sub-exp LWE for $n = \text{poly}(\lambda)$ and

$$P(x_1, \ldots, x_n) = P_1(x_1) \wedge \ldots \wedge P_n(x_n)$$

Main Result

(Informal) Theorem

Adaptively CPA-1-sided secure **secret-key** n-input PE from sub-exp LWE for $n = \text{poly}(\lambda)$ and

$$P(x_1, \ldots, x_n) = P_1(x_1) \wedge \ldots \wedge P_n(x_n)$$

SKE + Adaptively (CPA-1-sided) secure PE + Lockable Obfuscation

Adaptively (CPA-1-sided) secure PE → Sub-exp LWE + complexity leveraging

Lockable Obfuscation → LWE

Main Result

(Informal) Theorem

Adaptively CPA-1-sided secure **secret-key** n-input PE

from sub- … ly(λ) and

$P(x_1, \ldots,$ … $\wedge P_n(x_n)$

Single decryption key,
i.e., single dk_P

SKE + Adaptively (CPA-1-sided) secure PE + Lockable Obfuscation

Sub-exp LWE + complexity leveraging

LWE

CPA-1-sided security of multi-input PE (corruption setting)

$$((m_i^0)_{i\in[n]}, (m_i^1)_{i\in[n]}, (x_i^0)_{i\in[n]}, (x_i^1)_{i\in[n]}\alpha) \leftarrow \mathsf{A}_1^{\mathsf{Enc}(\mathsf{ek}_1,\cdot),\ldots,\mathsf{Enc}(\mathsf{ek}_n,\cdot),\mathsf{KGen}(\mathsf{msk},\cdot),\mathsf{Corr}(\cdot)}(1^\lambda, \mathsf{mpk})$$

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

$$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1^b, m_1^b), \ldots, c_n \leftarrow \mathsf{Enc}(\mathsf{ek}_n, x_n^b, m_n^b)$$

$$b' \leftarrow \mathsf{A}_2^{\mathsf{Enc}(\mathsf{ek}_1,\cdot),\ldots,\mathsf{Enc}(\mathsf{ek}_n,\cdot),\mathsf{KGen}(\mathsf{msk},\cdot),\mathsf{Corr}(\cdot)}(c_1, \ldots, c_n, \alpha)$$

WIN if $b = b'$

CPA-1-sided security of Multi-input PE

$\Pr[\textbf{WIN}] \leq \mathsf{negl}(\lambda)$, and $\forall \mathsf{dk}_P$ we have

$$\forall i \in [n] \;\; P(\ldots, x_i^0, \ldots) = P(\ldots, x_i^1, \ldots) = 0$$

CPA-1-sided security of multi-input PE (corruption setting)

$((m_i^0)_{i\in[n]}, (m_i^1)_{i\in[n]}, (x_i^0)_{i\in[n]}, (x_i^1)_{i\in[n]}\alpha) \leftarrow \mathsf{A}_1^{\mathsf{Enc}(\mathsf{ek}_1,\cdot),\ldots,\mathsf{Enc}(\mathsf{ek}_n,\cdot),\mathsf{KGen}(\mathsf{msk},\cdot),\mathsf{Corr}(\cdot)}(1^\lambda, \mathsf{mpk})$

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

c_1 ... $\mathsf{Enc}(\mathsf{ek}_n, x_n^b, m_n^b)$

If j corrupted then we consider $\forall x_j$

$b' \leftarrow \mathsf{A}_2^{\mathsf{Enc}(\mathsf{ek}_1,\cdot),\ldots,\mathsf{Enc}(\mathsf{ek}_n,\cdot),\mathsf{KGen}(\mathsf{msk},\cdot),\mathsf{Corr}(\cdot)}(c_1,\ldots,c_n,\alpha)$

WIN if $b = b'$

CPA-1-sided security of Multi-input PE

$\Pr[\mathbf{WIN}] \leq \mathsf{negl}(\lambda)$, and $\forall \mathsf{dk}_P$ we have

$$\forall i \in [n] \;\; P(\ldots, x_i^0, \ldots) = P(\ldots, x_i^1, \ldots) = 0$$

Main Result

(Informal) Theorem

Adaptively CPA-1-sided secure n-input PE in the **corruption setting** from sub-exp LWE for $n = O(1)$ and $P(x_1, \ldots, x_n) = P_1(x_1) \wedge \ldots \wedge P_n(x_n)$

Main Result

(Informal) Theorem

Adaptively CPA-1-sided secure n-input PE in the **corruption setting** from sub-exp LWE for $n = O(1)$ and

$$P(x_1, \ldots, x_n) = P_1(x_1) \wedge \ldots \wedge P_n(x_n)$$

With Wildcard:

$\forall i \in [n], \exists x_i^\star$ such that $\forall P_i$ we have $P_i(x_i^\star) = 1$

Main Result

(Informal) Theorem

Adaptively CPA-1-sided secure n-input PE in the **corruption setting** from sub-exp LWE for $n = O(1)$ and

$$P(x_1, \ldots, x_n) = P_1(x_1) \wedge \ldots \wedge P_n(x_n)$$

With Wildcard:

$\forall i \in [n], \exists x_i^\star$ such that $\forall P_i$ we have $P_i(x_i^\star) = 1$

Main Result

(Informal) Theorem

Adaptively CPA-1-sided secure n-input PE in the **corruption setting** from sub-exp LWE for $n = O(1)$ and

$$P(x_1, \ldots, x_n) = P_1(x_1) \land \ldots \land P_n(x_n)$$

PKE + Adaptively secure PE + Lockable Obfuscation

Adaptively secure PE ↓ Sub-exp LWE + complexity leveraging

Lockable Obfuscation ↓ LWE

Main Result

(Informal) Theorem

Adaptively CPA-1-sided secure n-input PE in the **corruption setting** from sub-exp LWE for $n = O(1)$ and

$$P(x_1, \ldots, x_n) = P_1(x_1) \wedge \ldots \wedge P_n(x_n)$$

PKE + <u>Adaptively secure FE</u> + <u>Lockable Obfuscation</u>

ABE

Main Result

(Informal) Theorem

Adaptively CPA-1-sided secure n-input PE in the **corruption setting** for $n = O(1)$ and $P(x_1, \ldots,$ $\wedge P_n(x_n)$

Single decryption key,
i.e., single dk_P

PKE + Adaptively secure ~~PE~~ ABE + Lockable Obfuscation

Lockable Obfuscation

Obfuscation
$\widetilde{C} \leftarrow \mathsf{Obf}(1^\lambda, C, y, m)$

Lockable Obfuscation

Obfuscation

$$\widetilde{C} \leftarrow \mathsf{Obf}(1^\lambda, C, y, m)$$

Functionality

$$\widetilde{C}(x) = \begin{cases} m & \text{if } C(x) = y \\ \bot & \text{otherwise} \end{cases}$$

Lockable Obfuscation

Obfuscation

$$\widetilde{C} \leftarrow \mathsf{Obf}(1^\lambda, C, y, m)$$

Functionality

$$\widetilde{C}(x) = \begin{cases} m & \text{if } C(x) = y \\ \perp & \text{otherwise} \end{cases}$$

VBB security (informal)

$\widetilde{C}$ can be **VBB simulated** when the **lock** y is sampled at **random** and y **unknown** to the **adversary**

Construction of 2-input PE for $P(x_1, x_2) = P_1(x_1) \wedge P_2(x_2)$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

Construction of 2-input PE for $P(x_1, x_2) = P_1(x_1) \wedge P_2(x_2)$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

1. Let $\mathsf{ek}_1 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_1)$

Construction of 2-input PE for $P(x_1, x_2) = P_1(x_1) \wedge P_2(x_2)$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

1. Let $\mathsf{ek}_1 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_1)$
2. Sample locks $y_1^{\mathsf{out}}, y_1^{\mathsf{in}}$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

Construction of 2-input PE for $P(x_1, x_2) = P_1(x_1) \wedge P_2(x_2)$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

1. Let $\mathsf{ek}_1 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_1)$
2. Sample locks $y_1^{\mathsf{out}}, y_1^{\mathsf{in}}$
3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1, x_2^\star), (y_1^{\mathsf{out}}, y_1^{\mathsf{in}}))$

Construction of 2-input PE for $P(x_1, x_2) = P_1(x_1) \wedge P_2(x_2)$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

1. Let $\mathsf{ek}_1 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_1)$
2. Sample locks $y_1^{\mathsf{out}}, y_1^{\mathsf{in}}$
3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1, x_2^\star), (y_1^{\mathsf{out}}, y_1^{\mathsf{in}}))$
4. $w_1 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, c))$

Construction of 2-input PE for $P(x_1, x_2) = P_1(x_1) \wedge P_2(x_2)$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

1. Let $\mathsf{ek}_1 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_1)$
2. Sample locks $y_1^{\mathsf{out}}, y_1^{\mathsf{in}}$
3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1, x_2^\star), (y_1^{\mathsf{out}}, y_1^{\mathsf{in}}))$
4. $w_1 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, c))$
5. $\widetilde{C}_1^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{in}}, \mathsf{sk}_1)$

Construction of 2-input PE for $P(x_1, x_2) = P_1(x_1) \wedge P_2(x_2)$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

1. Let $\mathsf{ek}_1 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_1)$
2. Sample locks $y_1^{\mathsf{out}}, y_1^{\mathsf{in}}$
3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1, x_2^\star), (y_1^{\mathsf{out}}, y_1^{\mathsf{in}}))$
4. $w_1 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, c))$
5. $\widetilde{C}_1^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{in}}, \mathsf{sk}_1)$
6. $\widetilde{C}_1^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{out}}, m_1)$

Construction of 2-input PE for $P(x_1, x_2) = P_1(x_1) \wedge P_2(x_2)$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

1. Let $\mathsf{ek}_1 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_1)$
2. Sample locks $y_1^{\mathsf{out}}, y_1^{\mathsf{in}}$
3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1, x_2^\star), (y_1^{\mathsf{out}}, y_1^{\mathsf{in}}))$
4. $w_1 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, c))$
5. $\widetilde{C}_1^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{in}}, \mathsf{sk}_1)$
6. $\widetilde{C}_1^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{out}}, m_1)$
7. Return $c_1 = (\widetilde{C}_1^{\mathsf{in}}, \widetilde{C}_1^{\mathsf{out}})$

Construction of 2-input PE for $P(x_1, x_2) = P_1(x_1) \wedge P_2(x_2)$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

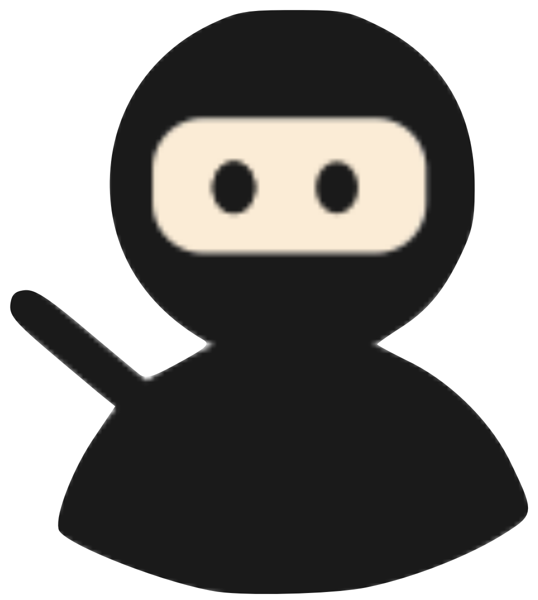
ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$	$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$
1. Let $\mathsf{ek}_1 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_1)$	1. Let $\mathsf{ek}_2 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_2)$
2. Sample locks $y_1^{\mathsf{out}}, y_1^{\mathsf{in}}$	2. Sample locks $y_2^{\mathsf{out}}, y_2^{\mathsf{in}}$
3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1, x_2^\star), (y_1^{\mathsf{out}}, y_1^{\mathsf{in}}))$	3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1^\star, x_2), (y_2^{\mathsf{out}}, y_2^{\mathsf{in}}))$
4. $w_1 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, c))$	4. $w_2 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, c))$
5. $\widetilde{C}_1^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C_{w_1, \mathsf{sk}_1}^{\mathsf{in}}, y_1^{\mathsf{in}}, \mathsf{sk}_1)$	5. $\widetilde{C}_2^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C_{w_2, \mathsf{sk}_2}^{\mathsf{in}}, y_2^{\mathsf{in}}, \mathsf{sk}_2)$
6. $\widetilde{C}_1^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C_{w_1, \mathsf{sk}_1}^{\mathsf{out}}, y_1^{\mathsf{out}}, m_1)$	6. $\widetilde{C}_2^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C_{w_2, \mathsf{sk}_2}^{\mathsf{out}}, y_2^{\mathsf{out}}, m_2)$
7. Return $c_1 = (\widetilde{C}_1^{\mathsf{in}}, \widetilde{C}_1^{\mathsf{out}})$	7. Return $c_2 = (\widetilde{C}_2^{\mathsf{in}}, \widetilde{C}_2^{\mathsf{out}})$

Construction of 2-input PE for $P(x_1, x_2) = P_1(x_1) \wedge P_2(x_2)$

Receiver



dk_P

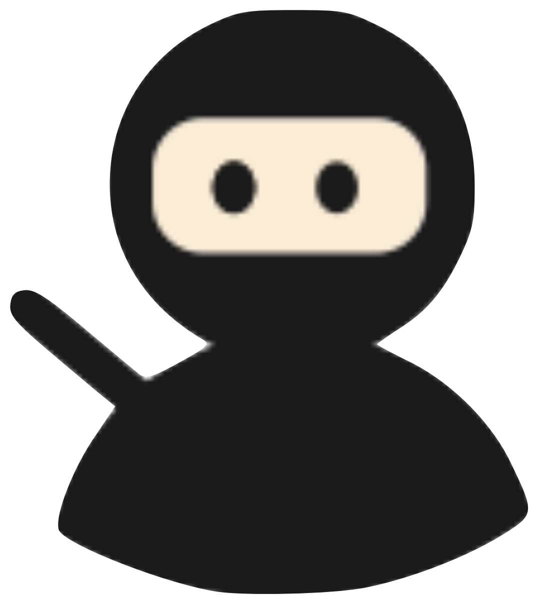
$m \leftarrow \mathsf{Dec}(\mathsf{dk}_P, c_1, c_2)$

Construction of 2-input PE for $P(x_1, x_2) = P_1(x_1) \wedge P_2(x_2)$

$\mathsf{dk}_P \leftarrow \mathsf{KGen}(\mathsf{msk}, P)$

Return $\mathsf{KGen}_{\mathsf{PE}}(\mathsf{msk}, P)$

Receiver



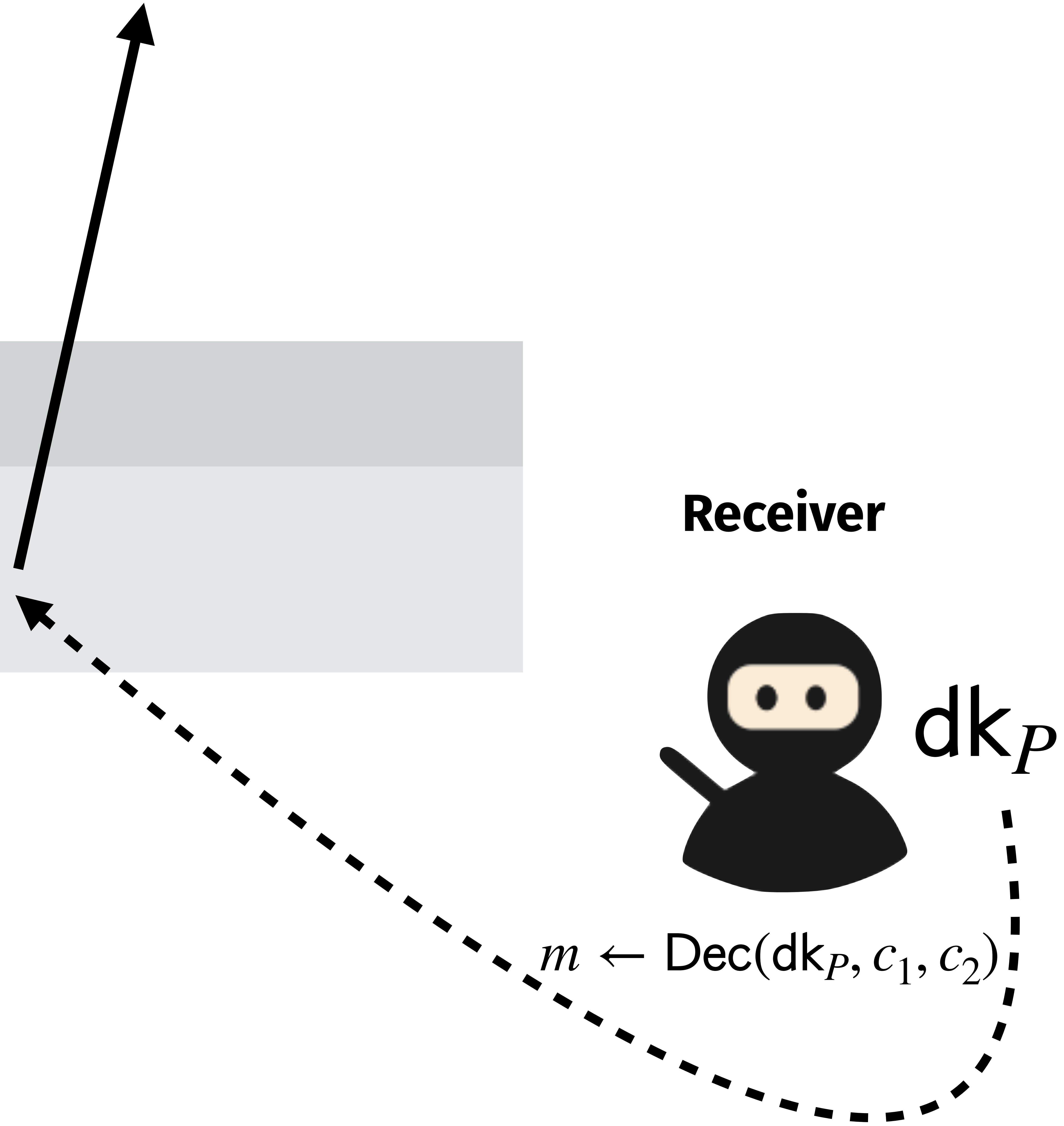
dk_P

$m \leftarrow \mathsf{Dec}(\mathsf{dk}_P, c_1, c_2)$

Construction of 2-input PE for $P(x_1, x_2) = P_1(x_1) \wedge P_2(x_2)$

$\mathsf{dk}_P \leftarrow \mathsf{KGen}(\mathsf{msk}, P)$

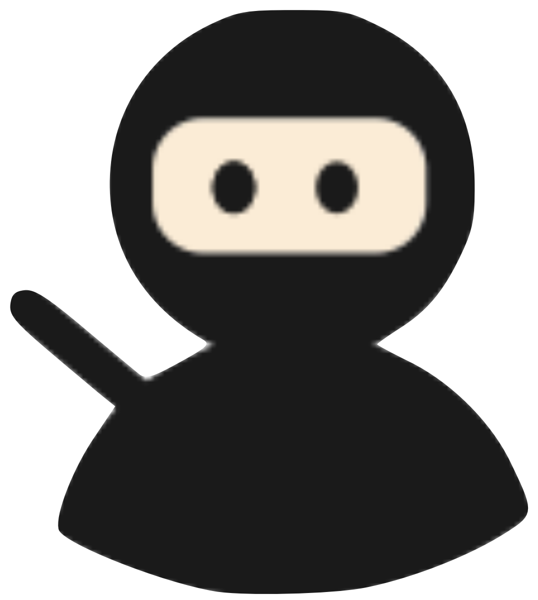
Return $\mathsf{KGen}_{\mathsf{PE}}(\mathsf{msk}, P)$



Construction of 2-input PE for $P(x_1, x_2) = P_1(x_1) \wedge P_2(x_2)$

$$(m_1, m_2) = \mathsf{Dec}(\mathsf{dk}_P, c_1, c_2)$$

Receiver



dk_P

$m \leftarrow \mathsf{Dec}(\mathsf{dk}_P, c_1, c_2)$

Construction of 2-input PE for $P(x_1, x_2) = P_1(x_1) \wedge P_2(x_2)$

$$(m_1, m_2) = \mathsf{Dec}(\mathsf{dk}_P, c_1, c_2)$$

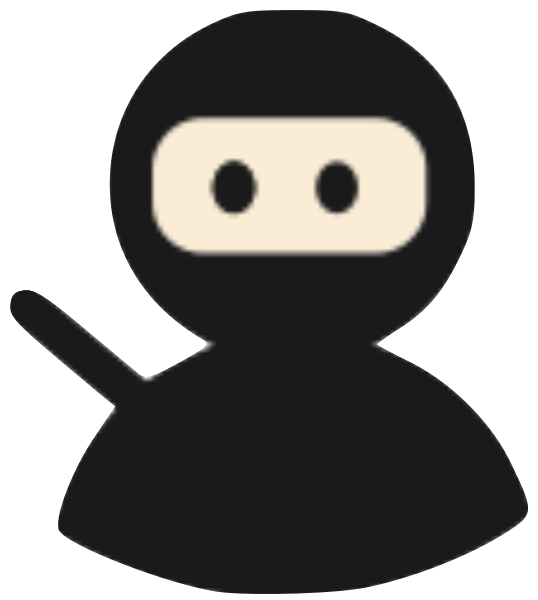
1. Let $c_1 = (\widetilde{C}_1^{\mathsf{in}}, \widetilde{C}_1^{\mathsf{out}})$ and $c_2 = (\widetilde{C}_2^{\mathsf{in}}, \widetilde{C}_2^{\mathsf{out}})$

Construction of 2-input PE for $P(x_1, x_2) = P_1(x_1) \wedge P_2(x_2)$

$$(m_1, m_2) = \mathsf{Dec}(\mathsf{dk}_P, c_1, c_2)$$

1. Let $c_1 = (\widetilde{C}_1^{\text{in}}, \widetilde{C}_1^{\text{out}})$ and $c_2 = (\widetilde{C}_2^{\text{in}}, \widetilde{C}_2^{\text{out}})$
2. $m_1 = \widetilde{C}_1^{\text{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\text{in}})$

Receiver



dk_P

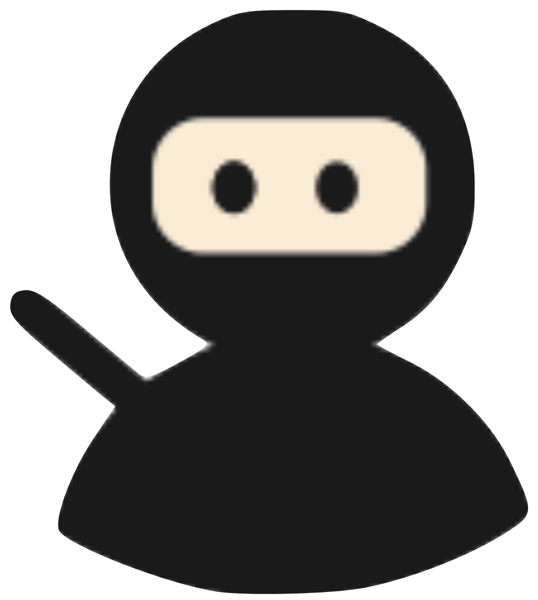
$m \leftarrow \mathsf{Dec}(\mathsf{dk}_P, c_1, c_2)$

Construction of 2-input PE for $P(x_1, x_2) = P_1(x_1) \wedge P_2(x_2)$

$$(m_1, m_2) = \mathsf{Dec}(\mathsf{dk}_P, c_1, c_2)$$

1. Let $c_1 = (\widetilde{C}_1^{\mathsf{in}}, \widetilde{C}_1^{\mathsf{out}})$ and $c_2 = (\widetilde{C}_2^{\mathsf{in}}, \widetilde{C}_2^{\mathsf{out}})$
2. $m_1 = \widetilde{C}_1^{\mathsf{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\mathsf{in}})$
3. $m_2 = \widetilde{C}_2^{\mathsf{out}}(\mathsf{dk}_P, \widetilde{C}_1^{\mathsf{in}})$

Receiver



dk_P

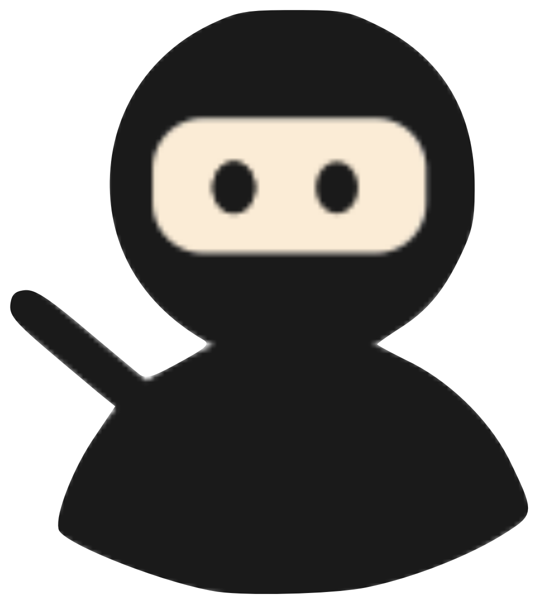
$m \leftarrow \mathsf{Dec}(\mathsf{dk}_P, c_1, c_2)$

Construction of 2-input PE for $P(x_1, x_2) = P_1(x_1) \wedge P_2(x_2)$

$$(m_1, m_2) = \mathsf{Dec}(\mathsf{dk}_P, c_1, c_2)$$

1. Let $c_1 = (\widetilde{C}_1^{\text{in}}, \widetilde{C}_1^{\text{out}})$ and $c_2 = (\widetilde{C}_2^{\text{in}}, \widetilde{C}_2^{\text{out}})$
2. $m_1 = \widetilde{C}_1^{\text{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\text{in}})$
3. $m_2 = \widetilde{C}_2^{\text{out}}(\mathsf{dk}_P, \widetilde{C}_1^{\text{in}})$
4. Return (m_1, m_2)

Receiver



dk_P

$m \leftarrow \mathsf{Dec}(\mathsf{dk}_P, c_1, c_2)$

Construction of 2-input PE for $P(x_1, x_2) = P_1(x_1) \wedge P_2(x_2)$

$(m_1, m_2) = \mathsf{Dec}(\mathsf{dk}_P, c_1, c_2)$

1. Let $c_1 = (\widetilde{C}_1^{\mathsf{in}}, \widetilde{C}_1^{\mathsf{out}})$ and $c_2 = (\widetilde{C}_2^{\mathsf{in}}, \widetilde{C}_2^{\mathsf{out}})$
2. $m_1 = \widetilde{C}_1^{\mathsf{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\mathsf{in}})$
3. $m_2 = \widetilde{C}_2^{\mathsf{out}}(\mathsf{dk}_P, \widetilde{C}_1^{\mathsf{in}})$
4. Return (m_1, m_2)

Definition of $\widetilde{C}_i^{\mathsf{in}}$ and $\widetilde{C}_i^{\mathsf{out}}$ and correctness

???

Receiver

$m \leftarrow \mathsf{Dec}(\mathsf{dk}_P, c_1, c_2)$

Decryption: Computation of $m_1 = \widetilde{C}_1^{\mathsf{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\mathsf{in}})$

Decryption: Computation of $m_1 = \widetilde{C}_1^{\mathsf{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\mathsf{in}})$

$\widetilde{C}_1^{\mathsf{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\mathsf{in}})$

Decryption: Computation of $m_1 = \widetilde{C}_1^{\mathsf{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\mathsf{in}})$

$\widetilde{C}_1^{\mathsf{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\mathsf{in}})$

$\widetilde{C}_1^{\mathsf{out}}$

1. Lock y_1^{out}
2. Message m_1
3. Hardcoded info
    1. Secret key sk_1
    2. Cipher w_1

Decryption: Computation of $m_1 = \widetilde{C}_1^{\mathsf{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\mathsf{in}})$

$\widetilde{C}_1^{\mathsf{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\mathsf{in}})$

$\widetilde{C}_2^{\mathsf{in}}(\mathsf{dk}_P, \mathsf{sk}_1)$

$\widetilde{C}_1^{\mathsf{out}}$

1. Lock y_1^{out}
2. Message m_1
3. Hardcoded info
   1. Secret key sk_1
   2. Cipher w_1

Decryption: Computation of $m_1 = \widetilde{C}_1^{\text{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\text{in}})$

$\widetilde{C}_1^{\text{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\text{in}})$

$\widetilde{C}_2^{\text{in}}(\mathsf{dk}_P, \mathsf{sk}_1)$

$\widetilde{C}_1^{\text{out}}$

1. Lock y_1^{out}
2. Message m_1
3. Hardcoded info
    1. Secret key sk_1
    2. Cipher w_1

$\widetilde{C}_2^{\text{in}}$

1. Lock y_2^{in}
2. Message sk_2
3. Hardcoded info
    1. Secret key sk_2
    2. Cipher w_2

Decryption: Computation of $m_1 = \widetilde{C}_1^{\mathsf{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\mathsf{in}})$

$\widetilde{C}_1^{\mathsf{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\mathsf{in}})$

$\widetilde{C}_2^{\mathsf{in}}(\mathsf{dk}_P, \mathsf{sk}_1)$

1. Obtain c by decrypting w_2 using $\mathsf{sk}_1, \mathsf{sk}_2$
2. Obtain $y_2^{\mathsf{out}}, y_2^{\mathsf{in}}$ by decrypting c using dk_P
3. Return y_2^{in}

$\widetilde{C}_1^{\mathsf{out}}$

1. Lock y_1^{out}
2. Message m_1
3. Hardcoded info
    1. Secret key sk_1
    2. Cipher w_1

$\widetilde{C}_2^{\mathsf{in}}$

1. Lock y_2^{in}
2. Message sk_2
3. Hardcoded info
    1. Secret key sk_2
    2. Cipher w_2

Decryption: Computation of $m_1 = \widetilde{C}_1^{\text{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\text{in}})$

$\widetilde{C}_1^{\text{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\text{in}})$

$\widetilde{C}_2^{\text{in}}(\mathsf{dk}_P, \mathsf{sk}_1)$

1. Obtain c by decrypting w_2 using $\mathsf{sk}_1, \mathsf{sk}_2$
2. Obtain $y_2^{\text{out}}, y_2^{\text{in}}$ by decrypting c using dk_P
3. Return y_2^{in}

$P_2(x_2) = 1$
since 1-slot has wildcard $x_1^\star$

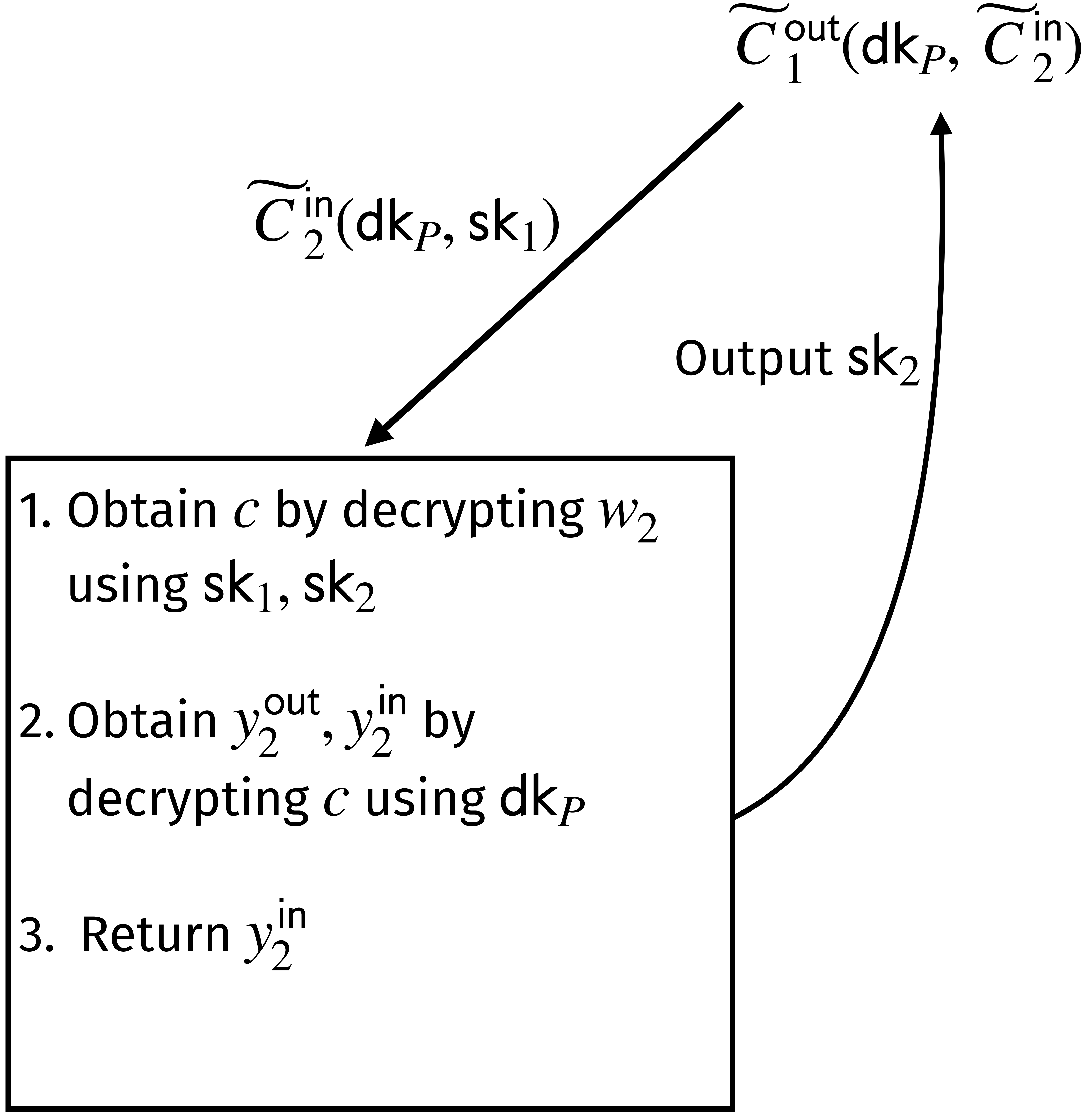
$\widetilde{C}_1^{\text{out}}$

1. Lock y_1^{out}
2. Message m_1
3. Hardcoded info
   1. Secret key sk_1
   2. Cipher w_1

$\widetilde{C}_2^{\text{in}}$

1. Lock y_2^{in}
2. Message sk_2
3. Hardcoded info
   1. Secret key sk_2
   2. Cipher w_2

Decryption: Computation of $m_1 = \widetilde{C}_1^{\mathsf{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\mathsf{in}})$



$\widetilde{C}_1^{\mathsf{out}}$

1. Lock y_1^{out}
2. Message m_1
3. Hardcoded info
   1. Secret key sk_1
   2. Cipher w_1

$\widetilde{C}_2^{\mathsf{in}}$

1. Lock y_2^{in}
2. Message sk_2
3. Hardcoded info
   1. Secret key sk_2
   2. Cipher w_2

Decryption: Computation of $m_1 = \widetilde{C}_1^{\mathsf{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\mathsf{in}})$

$\widetilde{C}_1^{\mathsf{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\mathsf{in}})$

$\widetilde{C}_2^{\mathsf{in}}(\mathsf{dk}_P, \mathsf{sk}_1)$

1. Obtain c by decrypting w_2 using $\mathsf{sk}_1, \mathsf{sk}_2$
2. Obtain $y_2^{\mathsf{out}}, y_2^{\mathsf{in}}$ by decrypting c using dk_P
3. Return y_2^{in}

Output sk_2

1. Obtain c by decrypting w_1 using $\mathsf{sk}_1, \mathsf{sk}_2$
2. Obtain $y_1^{\mathsf{out}}, y_1^{\mathsf{in}}$ by decrypting c using dk_P
3. Return y_1^{out}

$\widetilde{C}_1^{\mathsf{out}}$

1. Lock y_1^{out}
2. Message m_1
3. Hardcoded info
    1. Secret key sk_1
    2. Cipher w_1

$\widetilde{C}_2^{\mathsf{in}}$

1. Lock y_2^{in}
2. Message sk_2
3. Hardcoded info
    1. Secret key sk_2
    2. Cipher w_2

Decryption: Computation of $m_1 = \widetilde{C}_1^{\text{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\text{in}})$

$\widetilde{C}_1^{\text{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\text{in}})$

$\widetilde{C}_2^{\text{in}}(\mathsf{dk}_P, \mathsf{sk}_1)$

1. Obtain c by decrypting w_2 using $\mathsf{sk}_1, \mathsf{sk}_2$
2. Obtain $y_2^{\text{out}}, y_2^{\text{in}}$ by decrypting c using dk_P
3. Return y_2^{in}

Output sk_2

1. Obtain c by decrypting w_1 using $\mathsf{sk}_1, \mathsf{sk}_2$
2. Obtain $y_1^{\text{out}}, y_1^{\text{in}}$ by decrypting c using dk_P
3. Return y_1^{out}

$P_1(x_1) = 1$
since 2-slot has wildcard $x_2^\star$

$\widetilde{C}_1^{\text{out}}$

1. Lock y_1^{out}
2. Message m_1
3. Hardcoded info
   1. Secret key sk_1
   2. Cipher w_1

$\widetilde{C}_2^{\text{in}}$

1. Lock y_2^{in}
2. Message sk_2
3. Hardcoded info
   1. Secret key sk_2
   2. Cipher w_2

Decryption: Computation of $m_1 = \widetilde{C}_1^{\text{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\text{in}})$

$\widetilde{C}_1^{\text{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\text{in}}) = m_1$

$\widetilde{C}_2^{\text{in}}(\mathsf{dk}_P, \mathsf{sk}_1)$

1. Obtain c by decrypting w_2 using $\mathsf{sk}_1, \mathsf{sk}_2$
2. Obtain $y_2^{\text{out}}, y_2^{\text{in}}$ by decrypting c using dk_P
3. Return y_2^{in}

Output sk_2

1. Obtain c by decrypting w_1 using $\mathsf{sk}_1, \mathsf{sk}_2$
2. Obtain $y_1^{\text{out}}, y_1^{\text{in}}$ by decrypting c using dk_P
3. Return y_1^{out}

$P_1(x_1) = 1$
since 2-slot has wildcard $x_2^\star$

$\widetilde{C}_1^{\text{out}}$

1. Lock y_1^{out}
2. Message m_1
3. Hardcoded info
   1. Secret key sk_1
   2. Cipher w_1

$\widetilde{C}_2^{\text{in}}$

1. Lock y_2^{in}
2. Message sk_2
3. Hardcoded info
   1. Secret key sk_2
   2. Cipher w_2

Decryption for $n > 2$

$\mathbb{C}^{\mathsf{out}}_{c,\mathsf{sk},i}(\mathbb{C}_1, \ldots, \mathbb{C}_{n-1}, \mathsf{dk}_{\mathbb{P}})$

Initialize: $c_n = c, \mathsf{sk}'_i = \mathsf{sk}, \forall j \in [n] \setminus \{i\},\ \mathsf{sk}'_j = \bot$

// Execute each circuit received in input in order to retrieve the related secret key.

For t **from** 1 **to** $n-1$ **do:**

$\mathsf{Eval}_3(\mathbb{C}_t, (\mathbb{C}_{t+1}, \ldots, \mathbb{C}_{n-1}, \overbrace{\bot, \ldots, \bot}^{t-1}, \mathsf{sk}'_1, \ldots, \mathsf{sk}'_n, \mathsf{dk}_{\mathbb{P}})) = r$

If $r = \bot$: **return** $\bot$

Else: $\mathsf{sk}'_h = \overline{\mathsf{sk}}$ **where** $r = (\overline{\mathsf{sk}}, h)$ // Save the secret key returned by $\mathbb{C}_t$.

end for.

// At this point, all secret keys are known.

For j **from** n **to** 1 **do:** $\mathsf{Dec}_{2,j}(\mathsf{sk}'_j, c_j) = c_{j-1}$

$\mathsf{Dec}_1(\mathsf{dk}_{\mathbb{P}}, c_0) = v$

If $v = \bot$: **return** $\bot$

Else: **return** y_i^{out} **where** $v = (y_i^{\mathsf{in}}, y_i^{\mathsf{out}})$

$\mathbb{C}^{\mathsf{in}}_{c,\mathsf{sk},i}(\mathbb{C}_1, \ldots, \mathbb{C}_{n-2}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n, \mathsf{dk}_{\mathbb{P}})$

Initialize:

$c_n = c, \mathsf{sk}'_i = \mathsf{sk}, \mathbb{C}_{n-1} = \bot, k = \bot, \forall j \in [n] \setminus \{i\},\ \mathsf{sk}'_j = \mathsf{sk}_j$

If $\exists w \in [n-2]$ **such that** $\mathbb{C}_w \neq \bot$ **and** $\mathbb{C}_{w+1} = \bot$: $k = w$

end initialize.

If $k \neq \bot$ **do:** // If k = $\bot$, no circuit to execute.

// Execute each circuit received in input in order to retrieve the related secret key.

For $t \in [k]$ **do:**

$\mathsf{Eval}_3(\mathbb{C}_t, (\mathbb{C}_{t+1}, \ldots, \mathbb{C}_k, \overbrace{\bot, \ldots, \bot}^{n-2+t-k}, \mathsf{sk}'_1, \ldots, \mathsf{sk}'_n, \mathsf{dk}_{\mathbb{P}})) = r$

If $r = \bot$: **return** $\bot$

Else: $\mathsf{sk}'_h = \overline{\mathsf{sk}}$ **where** $r = (\overline{\mathsf{sk}}, h)$ // Save the secret key returned by $\mathbb{C}_t$.

end for.

end if.

// At this point, all secret keys are known.

For j **from** n **to** 1 **do:** $\mathsf{Dec}_{2,j}(\mathsf{sk}'_j, c_j) = c_{j-1}$

$\mathsf{Dec}_1(\mathsf{dk}_{\mathbb{P}}, c_0) = v$

If $v = \bot$: **return** $\bot$

Else: **return** y_i^{in} **where** $v = (y_i^{\mathsf{in}}, y_i^{\mathsf{out}})$

We support $n \in O(1)$

n-ary tree of height $n \Rightarrow$ Decryption running time is $O(n^n)$

Other considerations

CPA-2-sided security

Replace **CPA-1-sided** secure PE with **CPA-2-sided** secure PE $\Rightarrow$
CPA-2-sided secure multi-key/multi-input PE

Applications

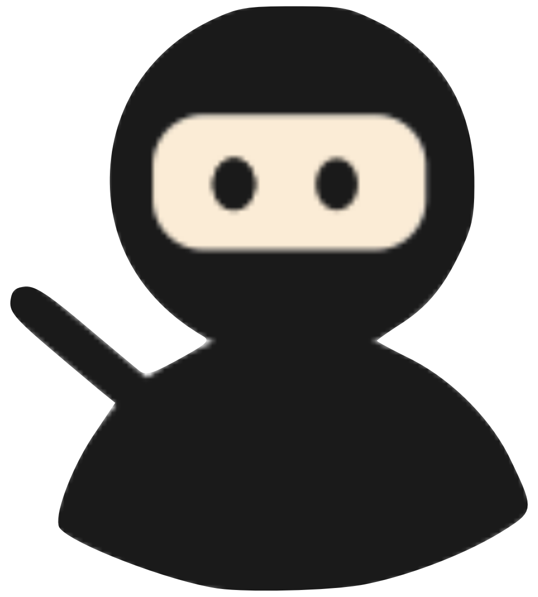
1. **CPA-1-sided** secure 2-key PE $\Rightarrow$ Matchmaking Encryption with **mismatch sec.** [AFNV19]
2. **CPA-1-sided** secure n-input PE for $n = O(1)$ in the **corruption setting** $\Rightarrow$

(Ind. based) **CPA-1-sided reusable (without session ids)** $(n-1)$**-robust**

non-interactive MPC for $(n \in O(1))$-parties where

$$f_P((x_1, m_1), \ldots, (x_n, m_n)) = \begin{cases} (m_1, \ldots, m_n) & \text{if } P_1(x_1) = 1 \wedge \ldots \wedge P_n(x_n) = 1 \\ \perp & \text{otherwise.} \end{cases}$$

Thank You!



https://eprint.iacr.org/2022/806

CPA-1-sided security of 2-input PE

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

CPA-1-sided security of 2-input PE

CPA-1-sided validity (<u>no corruptions</u>) : $P(x_1, x_2) = 0 \implies P_1(x_1) = 0 \vee P_2(x_2) = 0$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

CPA-1-sided security of 2-input PE

CPA-1-sided validity **(no corruptions)** **:** $P(x_1, x_2) = 0 \implies P_1(x_1) = 0 \lor P_2(x_2) = 0$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

CPA-1-sided security of 2-input PE

CPA-1-sided validity **(no corruptions)** : $P(x_1, x_2) = 0 \implies P_1(x_1) = 0 \vee P_2(x_2) = 0$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

1. Let $\mathsf{ek}_1 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_1)$
2. Sample locks $y_1^{\mathsf{out}}, y_1^{\mathsf{in}}$
3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1, x_2^\star), (y_1^{\mathsf{out}}, y_1^{\mathsf{in}}))$
4. $w_1 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$
5. $\widetilde{C}_1^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{in}}, \mathsf{sk}_1)$
6. $\widetilde{C}_1^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{out}}, m_1)$
7. Return $c_1 = (\widetilde{C}_1^{\mathsf{in}}, \widetilde{C}_1^{\mathsf{out}})$

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

1. Let $\mathsf{ek}_2 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_2)$
2. Sample locks $y_2^{\mathsf{out}}, y_2^{\mathsf{in}}$
3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1^\star, x_2), (y_2^{\mathsf{out}}, y_2^{\mathsf{in}}))$
4. $w_2 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$
5. $\widetilde{C}_2^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_2, \mathsf{sk}_2}, y_2^{\mathsf{in}}, \mathsf{sk}_2)$
6. $\widetilde{C}_2^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_2, \mathsf{sk}_2}, y_2^{\mathsf{out}}, m_2)$
7. Return $c_2 = (\widetilde{C}_2^{\mathsf{in}}, \widetilde{C}_2^{\mathsf{out}})$

CPA-1-sided security of 2-input PE

CPA-1-sided validity (<u>no corruptions</u>): $P(x_1, x_2) = 0 \implies P_1(x_1) = 0 \vee P_2(x_2) = 0$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$	$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$
1. Let $\mathsf{ek}_1 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_1)$	1. Let $\mathsf{ek}_2 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_2)$
2. Sample locks $y_1^{\mathsf{out}}, y_1^{\mathsf{in}}$	2. Sample locks $y_2^{\mathsf{out}}, y_2^{\mathsf{in}}$
3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1, x_2^\star), (y_1^{\mathsf{out}}, y_1^{\mathsf{in}}))$	3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1^\star, x_2), (y_2^{\mathsf{out}}, y_2^{\mathsf{in}}))$
4. $w_1 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$	4. $w_2 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$
5. $\widetilde{C}_1^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{in}}, \mathsf{sk}_1)$	5. $\widetilde{C}_2^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_2, \mathsf{sk}_2}, y_2^{\mathsf{in}}, \mathsf{sk}_2)$
6. $\widetilde{C}_1^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{out}}, m_1)$	6. $\widetilde{C}_2^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_2, \mathsf{sk}_2}, y_2^{\mathsf{out}}, m_2)$
7. Return $c_1 = (\widetilde{C}_1^{\mathsf{in}}, \widetilde{C}_1^{\mathsf{out}})$	7. Return $c_2 = (\widetilde{C}_2^{\mathsf{in}}, \widetilde{C}_2^{\mathsf{out}})$

CPA-1-sided security of 2-input PE

CPA-1-sided validity (no corruptions): $P(x_1, x_2) = 0 \implies P_1(x_1) = 0 \vee P_2(x_2) = 0$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$	$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$
1. Let $\mathsf{ek}_1 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_1)$	1. Let $\mathsf{ek}_2 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_2)$
2. Sample locks $y_1^{\mathsf{out}}, y_1^{\mathsf{in}}$	2. Sample locks $y_2^{\mathsf{out}}, y_2^{\mathsf{in}}$
3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1, x_2^\star), (y_1^{\mathsf{out}}, y_1^{\mathsf{in}}))$	3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1^\star, x_2), (y_2^{\mathsf{out}}, y_2^{\mathsf{in}}))$
4. $w_1 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$	4. $w_2 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$
5. $\widetilde{C}_1^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{in}}, \mathsf{sk}_1)$	5. $\widetilde{C}_2^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_2, \mathsf{sk}_2}, y_2^{\mathsf{in}}, \mathsf{sk}_2)$
6. $\widetilde{C}_1^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{out}}, m_1)$	6. $\widetilde{C}_2^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_2, \mathsf{sk}_2}, y_2^{\mathsf{out}}, m_2)$
7. Return $c_1 = (\widetilde{C}_1^{\mathsf{in}}, \widetilde{C}_1^{\mathsf{out}})$	7. Return $c_2 = (\widetilde{C}_2^{\mathsf{in}}, \widetilde{C}_2^{\mathsf{out}})$

CPA-1-sided security of 2-input PE

CPA-1-sided validity (no corruptions): $P(x_1, x_2) = 0 \implies P_1(x_1) = 0 \vee P_2(x_2) = 0$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$	$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$
1. Let $\mathsf{ek}_1 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_1)$	1. Let $\mathsf{ek}_2 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_2)$
2. Sample locks $y_1^{\mathsf{out}}, y_1^{\mathsf{in}}$	2. Sample locks $y_2^{\mathsf{out}}, y_2^{\mathsf{in}}$
3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1, x_2^\star), (y_1^{\mathsf{out}}, y_1^{\mathsf{in}}))$	3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1^\star, x_2), (y_2^{\mathsf{out}}, y_2^{\mathsf{in}}))$
4. $w_1 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$	4. $w_2 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$
Simulate 5. $\widetilde{C}_1^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{in}}, \mathsf{sk}_1)$	5. $\widetilde{C}_2^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_2, \mathsf{sk}_2}, y_2^{\mathsf{in}}, \mathsf{sk}_2)$
Simulate 6. $\widetilde{C}_1^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{out}}, m_1)$	6. $\widetilde{C}_2^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_2, \mathsf{sk}_2}, y_2^{\mathsf{out}}, m_2)$
7. Return $c_1 = (\widetilde{C}_1^{\mathsf{in}}, \widetilde{C}_1^{\mathsf{out}})$	7. Return $c_2 = (\widetilde{C}_2^{\mathsf{in}}, \widetilde{C}_2^{\mathsf{out}})$

CPA-1-sided security of 2-input PE

CPA-1-sided validity (no corruptions): $P(x_1, x_2) = 0 \implies P_1(x_1) = 0 \vee P_2(x_2) = 0$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$	$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$
1. Let $\mathsf{ek}_1 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_1)$	1. Let $\mathsf{ek}_2 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_2)$
2. Sample locks $y_1^{\mathsf{out}}, y_1^{\mathsf{in}}$	2. Sample locks $y_2^{\mathsf{out}}, y_2^{\mathsf{in}}$
3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1, x_2^\star), (y_1^{\mathsf{out}}, y_1^{\mathsf{in}}))$	3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1^\star, x_2), (y_2^{\mathsf{out}}, y_2^{\mathsf{in}}))$
4. $w_1 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$	4. $w_2 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$
Simulate 5. $\widetilde{C}_1^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{in}}, \mathsf{sk}_1)$	5. $\widetilde{C}_2^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_2, \mathsf{sk}_2}, y_2^{\mathsf{in}}, \mathsf{sk}_2)$
Simulate 6. $\widetilde{C}_1^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{out}}, m_1)$	6. $\widetilde{C}_2^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_2, \mathsf{sk}_2}, y_2^{\mathsf{out}}, m_2)$
7. Return $c_1 = (\widetilde{C}_1^{\mathsf{in}}, \widetilde{C}_1^{\mathsf{out}})$	7. Return $c_2 = (\widetilde{C}_2^{\mathsf{in}}, \widetilde{C}_2^{\mathsf{out}})$

CPA-1-sided security of 2-input PE

CPA-1-sided validity (no corruptions): $P(x_1, x_2) = 0 \implies P_1(x_1) = 0 \vee P_2(x_2) = 0$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$	$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$
1. Let $\mathsf{ek}_1 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_1)$	1. Let $\mathsf{ek}_2 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_2)$
2. Sample locks $y_1^{\mathsf{out}}, y_1^{\mathsf{in}}$	2. Sample locks $y_2^{\mathsf{out}}, y_2^{\mathsf{in}}$
3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1, x_2^\star), (y_1^{\mathsf{out}}, y_1^{\mathsf{in}}))$	3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1^\star, x_2), (y_2^{\mathsf{out}}, y_2^{\mathsf{in}}))$
4. $w_1 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$	4. $w_2 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$
Simulate 5. $\widetilde{C}_1^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{in}}, \mathsf{sk}_1)$	5. $\widetilde{C}_2^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_2, \mathsf{sk}_2}, y_2^{\mathsf{in}}, \mathsf{sk}_2)$
Simulate 6. $\widetilde{C}_1^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{out}}, m_1)$	6. $\widetilde{C}_2^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_2, \mathsf{sk}_2}, y_2^{\mathsf{out}}, m_2)$
7. Return $c_1 = (\widetilde{C}_1^{\mathsf{in}}, \widetilde{C}_1^{\mathsf{out}})$	7. Return $c_2 = (\widetilde{C}_2^{\mathsf{in}}, \widetilde{C}_2^{\mathsf{out}})$

CPA-1-sided security of 2-input PE

CPA-1-sided validity (no corruptions): $P(x_1, x_2) = 0 \implies P_1(x_1) = 0 \vee P_2(x_2) = 0$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$	$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$
1. Let $\mathsf{ek}_1 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_1)$	1. Let $\mathsf{ek}_2 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_2)$
2. Sample locks $y_1^{\mathsf{out}}, y_1^{\mathsf{in}}$	2. Sample locks $y_2^{\mathsf{out}}, y_2^{\mathsf{in}}$
3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1, x_2^\star), (y_1^{\mathsf{out}}, y_1^{\mathsf{in}}))$	3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1^\star, x_2), (y_2^{\mathsf{out}}, y_2^{\mathsf{in}}))$
4. $w_1 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$	4. $w_2 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$
Simulate 5. $\widetilde{C}_1^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_1,\mathsf{sk}_1}, y_1^{\mathsf{in}}, \mathsf{sk}_1)$	**Simulate** 5. $\widetilde{C}_2^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_2,\mathsf{sk}_2}, y_2^{\mathsf{in}}, \mathsf{sk}_2)$
Simulate 6. $\widetilde{C}_1^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_1,\mathsf{sk}_1}, y_1^{\mathsf{out}}, m_1)$	**Simulate** 6. $\widetilde{C}_2^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_2,\mathsf{sk}_2}, y_2^{\mathsf{out}}, m_2)$
7. Return $c_1 = (\widetilde{C}_1^{\mathsf{in}}, \widetilde{C}_1^{\mathsf{out}})$	7. Return $c_2 = (\widetilde{C}_2^{\mathsf{in}}, \widetilde{C}_2^{\mathsf{out}})$

CPA-1-sided security of 2-input PE

CPA-1-sided validity (#2 corrupted) : $P(x_1, x_2) = 0 \implies P_1(x_1) = 0$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

1. Let $\mathsf{ek}_1 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_1)$
2. Sample locks $y_1^{\mathsf{out}}, y_1^{\mathsf{in}}$
3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1, x_2^\star), (y_1^{\mathsf{out}}, y_1^{\mathsf{in}}))$
4. $w_1 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$
5. $\widetilde{C}_1^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_1,\mathsf{sk}_1}, y_1^{\mathsf{in}}, \mathsf{sk}_1)$
6. $\widetilde{C}_1^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_1,\mathsf{sk}_1}, y_1^{\mathsf{out}}, m_1)$
7. Return $c_1 = (\widetilde{C}_1^{\mathsf{in}}, \widetilde{C}_1^{\mathsf{out}})$

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

1. Let $\mathsf{ek}_2 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_2)$
2. Sample locks $y_2^{\mathsf{out}}, y_2^{\mathsf{in}}$
3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1^\star, x_2), (y_2^{\mathsf{out}}, y_2^{\mathsf{in}}))$
4. $w_2 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$
5. $\widetilde{C}_2^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_2,\mathsf{sk}_2}, y_2^{\mathsf{in}}, \mathsf{sk}_2)$
6. $\widetilde{C}_2^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_2,\mathsf{sk}_2}, y_2^{\mathsf{out}}, m_2)$
7. Return $c_2 = (\widetilde{C}_2^{\mathsf{in}}, \widetilde{C}_2^{\mathsf{out}})$

CPA-1-sided security of 2-input PE

CPA-1-sided validity (#2 corrupted) : $P(x_1, x_2) = 0 \implies P_1(x_1) = 0$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$	$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$
1. Let $\mathsf{ek}_1 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_1)$	1. Let $\mathsf{ek}_2 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_2)$
2. Sample locks $y_1^{\mathsf{out}}, y_1^{\mathsf{in}}$	2. Sample locks $y_2^{\mathsf{out}}, y_2^{\mathsf{in}}$
3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1, x_2^\star), (y_1^{\mathsf{out}}, y_1^{\mathsf{in}}))$	3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1^\star, x_2), (y_2^{\mathsf{out}}, y_2^{\mathsf{in}}))$
4. $w_1 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$	4. $w_2 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$
5. $\widetilde{C}_1^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_1,\mathsf{sk}_1}, y_1^{\mathsf{in}}, \mathsf{sk}_1)$	5. $\widetilde{C}_2^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_2,\mathsf{sk}_2}, y_2^{\mathsf{in}}, \mathsf{sk}_2)$
6. $\widetilde{C}_1^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_1,\mathsf{sk}_1}, y_1^{\mathsf{out}}, m_1)$	6. $\widetilde{C}_2^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_2,\mathsf{sk}_2}, y_2^{\mathsf{out}}, m_2)$
7. Return $c_1 = (\widetilde{C}_1^{\mathsf{in}}, \widetilde{C}_1^{\mathsf{out}})$	7. Return $c_2 = (\widetilde{C}_2^{\mathsf{in}}, \widetilde{C}_2^{\mathsf{out}})$

CPA-1-sided security of 2-input PE

CPA-1-sided validity (#2 corrupted): $P(x_1, x_2) = 0 \implies P_1(x_1) = 0$

Sender #1

ek$_1$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek$_2$

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$	$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$
1. Let $\mathsf{ek}_1 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_1)$	1. Let $\mathsf{ek}_2 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_2)$
2. Sample locks $y_1^{\mathsf{out}}, y_1^{\mathsf{in}}$	2. Sample locks $y_2^{\mathsf{out}}, y_2^{\mathsf{in}}$
3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1, x_2^\star), (y_1^{\mathsf{out}}, y_1^{\mathsf{in}}))$	3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1^\star, x_2), (y_2^{\mathsf{out}}, y_2^{\mathsf{in}}))$
4. $w_1 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$	4. $w_2 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$
5. $\widetilde{C}_1^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{in}}, \mathsf{sk}_1)$	5. $\widetilde{C}_2^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_2, \mathsf{sk}_2}, y_2^{\mathsf{in}}, \mathsf{sk}_2)$
6. $\widetilde{C}_1^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{out}}, m_1)$	6. $\widetilde{C}_2^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_2, \mathsf{sk}_2}, y_2^{\mathsf{out}}, m_2)$
7. Return $c_1 = (\widetilde{C}_1^{\mathsf{in}}, \widetilde{C}_1^{\mathsf{out}})$	7. Return $c_2 = (\widetilde{C}_2^{\mathsf{in}}, \widetilde{C}_2^{\mathsf{out}})$

CPA-1-sided security of 2-input PE

CPA-1-sided validity (#2 corrupted) : $P(x_1, x_2) = 0 \Longrightarrow P_1(x_1) = 0$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$	$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$
1. Let $\mathsf{ek}_1 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_1)$	1. Let $\mathsf{ek}_2 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_2)$
2. Sample locks $y_1^{\mathsf{out}}, y_1^{\mathsf{in}}$ (crossed out)	2. Sample locks $y_2^{\mathsf{out}}, y_2^{\mathsf{in}}$
3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1, x_2^\star), (y_1^{\mathsf{out}}, y_1^{\mathsf{in}}))$ (locks crossed out)	3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1^\star, x_2), (y_2^{\mathsf{out}}, y_2^{\mathsf{in}}))$
4. $w_1 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$	4. $w_2 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$
Simulate 5. $\widetilde{C}_1^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{in}}, \mathsf{sk}_1)$	5. $\widetilde{C}_2^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_2, \mathsf{sk}_2}, y_2^{\mathsf{in}}, \mathsf{sk}_2)$
Simulate 6. $\widetilde{C}_1^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{out}}, m_1)$	6. $\widetilde{C}_2^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_2, \mathsf{sk}_2}, y_2^{\mathsf{out}}, m_2)$
7. Return $c_1 = (\widetilde{C}_1^{\mathsf{in}}, \widetilde{C}_1^{\mathsf{out}})$	7. Return $c_2 = (\widetilde{C}_2^{\mathsf{in}}, \widetilde{C}_2^{\mathsf{out}})$

CPA-1-sided security of 2-input PE

CPA-1-sided validity (#2 corrupted): $P(x_1, x_2) = 0 \implies P_1(x_1) = 0$

Sender #1

ek$_1$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek$_2$

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$	$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$
1. Let $\mathsf{ek}_1 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_1)$	1. Let $\mathsf{ek}_2 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_2)$
2. Sample locks $y_1^{\mathsf{out}}, y_1^{\mathsf{in}}$	2. Sample locks $y_2^{\mathsf{out}}, y_2^{\mathsf{in}}$
3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1, x_2^\star), (y_1^{\mathsf{out}}, y_1^{\mathsf{in}}))$	3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1^\star, x_2), (y_2^{\mathsf{out}}, y_2^{\mathsf{in}}))$
4. $w_1 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$	4. $w_2 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$
Simulate 5. $\widetilde{C}_1^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{in}}, \mathsf{sk}_1)$	5. $\widetilde{C}_2^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_2, \mathsf{sk}_2}, y_2^{\mathsf{in}}, \mathsf{sk}_2)$
Simulate 6. $\widetilde{C}_1^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{out}}, m_1)$	6. $\widetilde{C}_2^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_2, \mathsf{sk}_2}, y_2^{\mathsf{out}}, m_2)$
7. Return $c_1 = (\widetilde{C}_1^{\mathsf{in}}, \widetilde{C}_1^{\mathsf{out}})$	7. Return $c_2 = (\widetilde{C}_2^{\mathsf{in}}, \widetilde{C}_2^{\mathsf{out}})$

CPA-1-sided security of 2-input PE

CPA-1-sided validity (#2 corrupted) : $P(x_1, x_2) = 0 \implies P_1(x_1) = 0$

Sender #1

ek_1

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$

Sender #2

ek_2

$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$

$c_1 \leftarrow \mathsf{Enc}(\mathsf{ek}_1, x_1, m_1)$	$c_2 \leftarrow \mathsf{Enc}(\mathsf{ek}_2, x_2, m_2)$
1. Let $\mathsf{ek}_1 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_1)$	1. Let $\mathsf{ek}_2 = (\mathsf{mpk}, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_2)$
2. Sample locks $y_1^{\mathsf{out}}, y_1^{\mathsf{in}}$	2. Sample locks $y_2^{\mathsf{out}}, y_2^{\mathsf{in}}$
3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1, x_2^\star), (y_1^{\mathsf{out}}, y_1^{\mathsf{in}}))$	3. $c \leftarrow \mathsf{Enc}_{\mathsf{PE}}(\mathsf{mpk}, (x_1^\star, x_2), (y_2^{\mathsf{out}}, y_2^{\mathsf{in}}))$
4. $w_1 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$	4. $w_2 \leftarrow \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_1, \mathsf{Enc}_{\mathsf{pke}}(\mathsf{pk}_2, v))$
Simulate 5. $\widetilde{C}_1^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{in}}, \mathsf{sk}_1)$	**Simulate** 5. $\widetilde{C}_2^{\mathsf{in}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{in}}_{w_2, \mathsf{sk}_2}, y_2^{\mathsf{in}}, \mathsf{sk}_2)$
Simulate 6. $\widetilde{C}_1^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_1, \mathsf{sk}_1}, y_1^{\mathsf{out}}, m_1)$	**Simulate** 6. $\widetilde{C}_2^{\mathsf{out}} \leftarrow \mathsf{Obf}(1^\lambda, C^{\mathsf{out}}_{w_2, \mathsf{sk}_2}, y_2^{\mathsf{out}}, m_2)$
7. Return $c_1 = (\widetilde{C}_1^{\mathsf{in}}, \widetilde{C}_1^{\mathsf{out}})$	7. Return $c_2 = (\widetilde{C}_2^{\mathsf{in}}, \widetilde{C}_2^{\mathsf{out}})$

Execute $\widetilde{C}_1^{\mathsf{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\mathsf{in}}, \widetilde{C}_3^{\mathsf{in}})$

$\widetilde{C}_1^{\mathsf{out}}$

1. Lock y_1^{out}
2. Message m_1
3. Hardcoded info
    1. Secret key sk_1
    2. Cipher w_1

$\widetilde{C}_i^{\mathsf{in}}$ for $i \in [3]$

1. Lock y_i^{in}
2. Message sk_i
3. Hardcoded info
    1. Secret key sk_i
    2. Cipher w_i

Execute $\widetilde{C}_1^{\text{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\text{in}}, \widetilde{C}_3^{\text{in}})$

Execute $\widetilde{C}_2^{\text{in}}(\mathsf{dk}_P, \mathsf{sk}_1, \widetilde{C}_3^{\text{in}})$

$\widetilde{C}_1^{\text{out}}$

1. Lock y_1^{out}
2. Message m_1
3. Hardcoded info
    1. Secret key sk_1
    2. Cipher w_1

$\widetilde{C}_i^{\text{in}}$ for $i \in [3]$

1. Lock y_i^{in}
2. Message sk_i
3. Hardcoded info
    1. Secret key sk_i
    2. Cipher w_i

Execute $\widetilde{C}_1^{\mathsf{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\mathsf{in}}, \widetilde{C}_3^{\mathsf{in}})$

Execute $\widetilde{C}_2^{\mathsf{in}}(\mathsf{dk}_P, \mathsf{sk}_1, \widetilde{C}_3^{\mathsf{in}})$

Execute $\widetilde{C}_3^{\mathsf{in}}(\mathsf{dk}_P, \mathsf{sk}_1, \mathsf{sk}_3)$

$\widetilde{C}_1^{\mathsf{out}}$

1. Lock y_1^{out}
2. Message m_1
3. Hardcoded info
    1. Secret key sk_1
    2. Cipher w_1

$\widetilde{C}_i^{\mathsf{in}}$ for $i \in [3]$

1. Lock y_i^{in}
2. Message sk_i
3. Hardcoded info
    1. Secret key sk_i
    2. Cipher w_i

Execute $\widetilde{C}_1^{\mathsf{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\mathsf{in}}, \widetilde{C}_3^{\mathsf{in}})$

Execute $\widetilde{C}_2^{\mathsf{in}}(\mathsf{dk}_P, \mathsf{sk}_1, \widetilde{C}_3^{\mathsf{in}})$

Execute $\widetilde{C}_3^{\mathsf{in}}(\mathsf{dk}_P, \mathsf{sk}_1, \mathsf{sk}_3)$

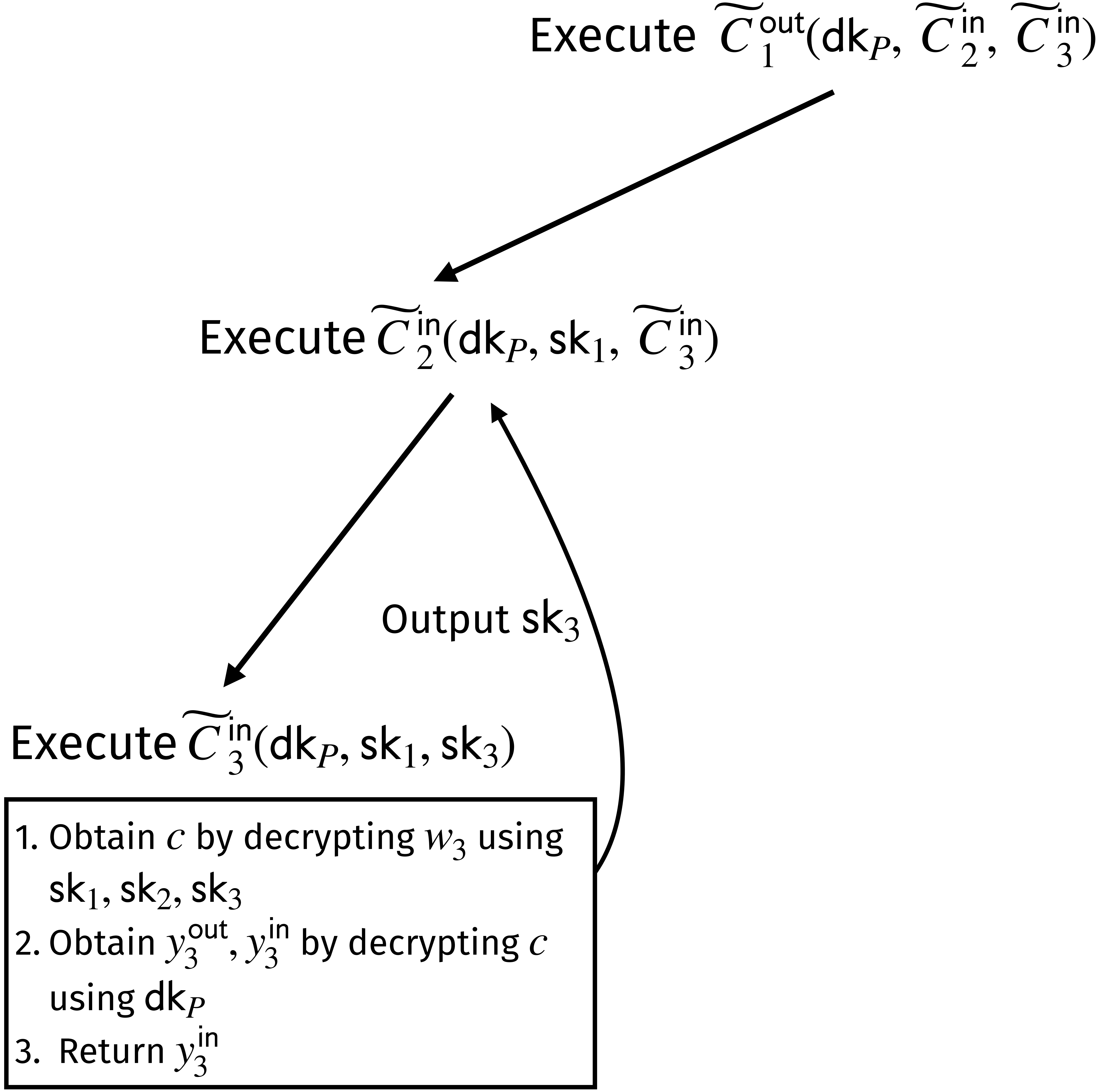
1. Obtain c by decrypting w_3 using $\mathsf{sk}_1, \mathsf{sk}_2, \mathsf{sk}_3$
2. Obtain $y_3^{\mathsf{out}}, y_3^{\mathsf{in}}$ by decrypting c using dk_P
3. Return y_3^{in}

$\widetilde{C}_1^{\mathsf{out}}$

1. Lock y_1^{out}
2. Message m_1
3. Hardcoded info
    1. Secret key sk_1
    2. Cipher w_1

$\widetilde{C}_i^{\mathsf{in}}$ for $i \in [3]$

1. Lock y_i^{in}
2. Message sk_i
3. Hardcoded info
    1. Secret key sk_i
    2. Cipher w_i

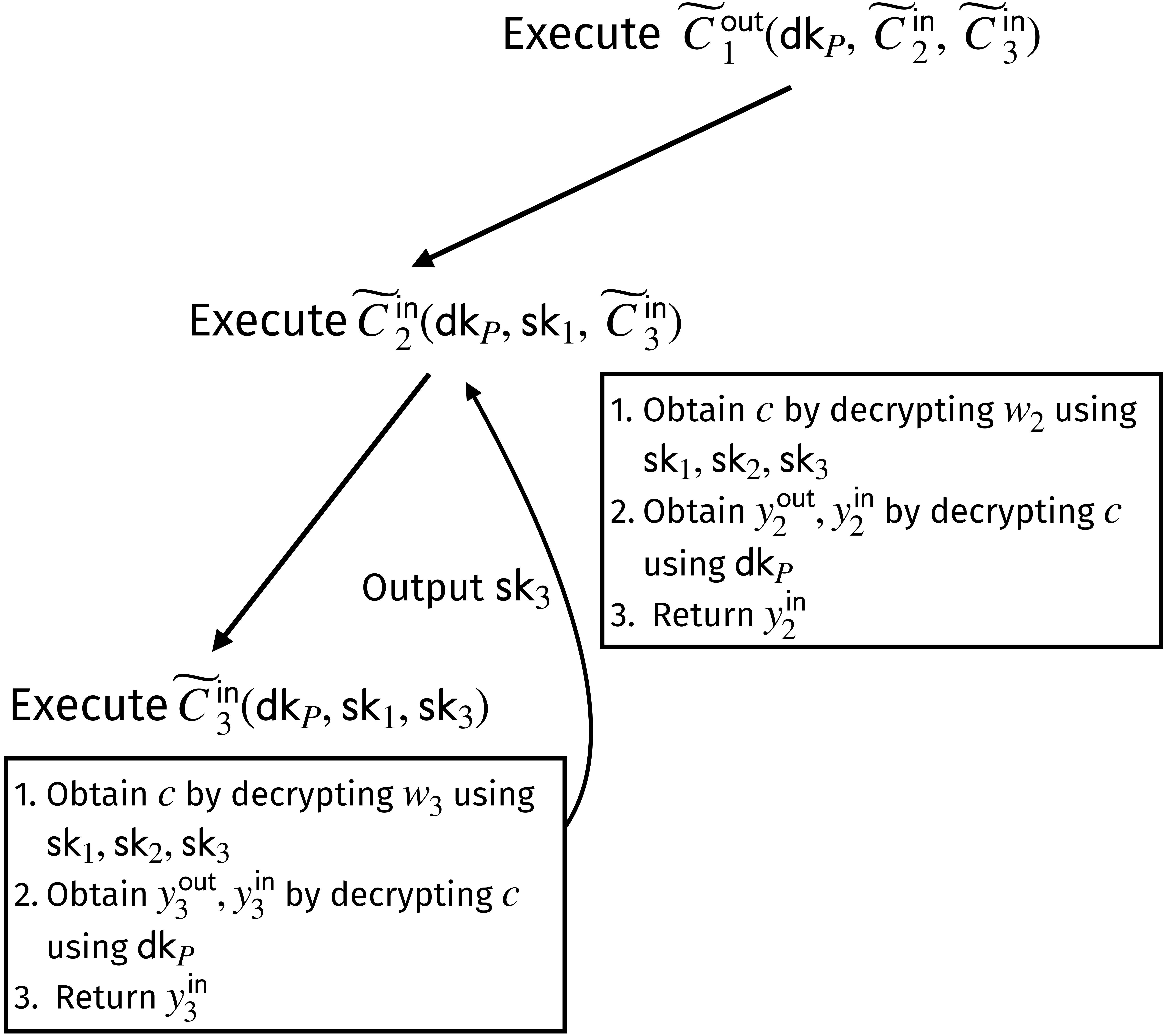


$\widetilde{C}_1^{\text{out}}$

1. Lock y_1^{out}
2. Message m_1
3. Hardcoded info
    1. Secret key sk_1
    2. Cipher w_1

$\widetilde{C}_i^{\text{in}}$ for $i \in [3]$

1. Lock y_i^{in}
2. Message sk_i
3. Hardcoded info
    1. Secret key sk_i
    2. Cipher w_i

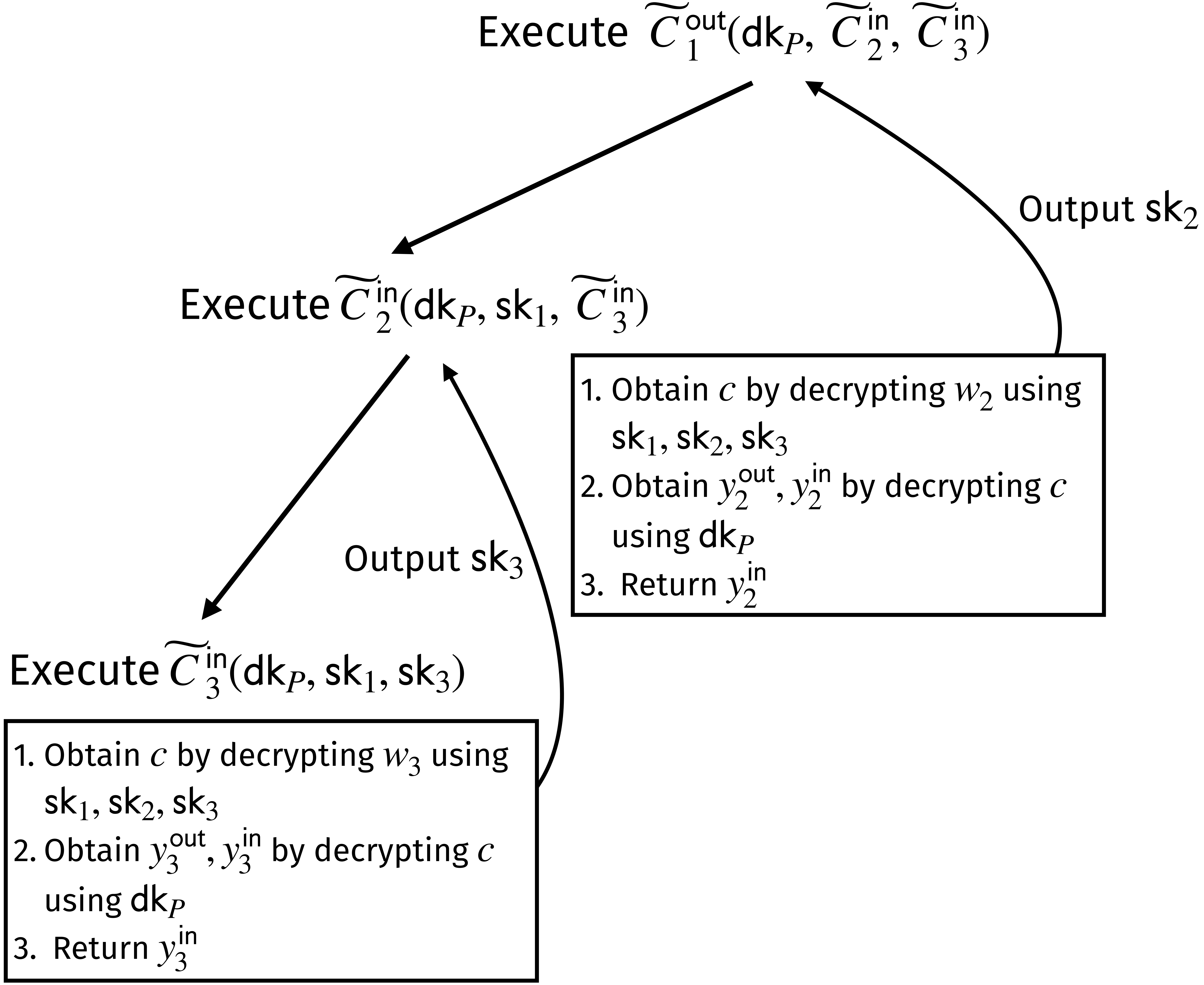


$\widetilde{C}_1^{\mathsf{out}}$

1. Lock y_1^{out}
2. Message m_1
3. Hardcoded info
   1. Secret key sk_1
   2. Cipher w_1

$\widetilde{C}_i^{\mathsf{in}}$ for $i \in [3]$

1. Lock y_i^{in}
2. Message sk_i
3. Hardcoded info
   1. Secret key sk_i
   2. Cipher w_i

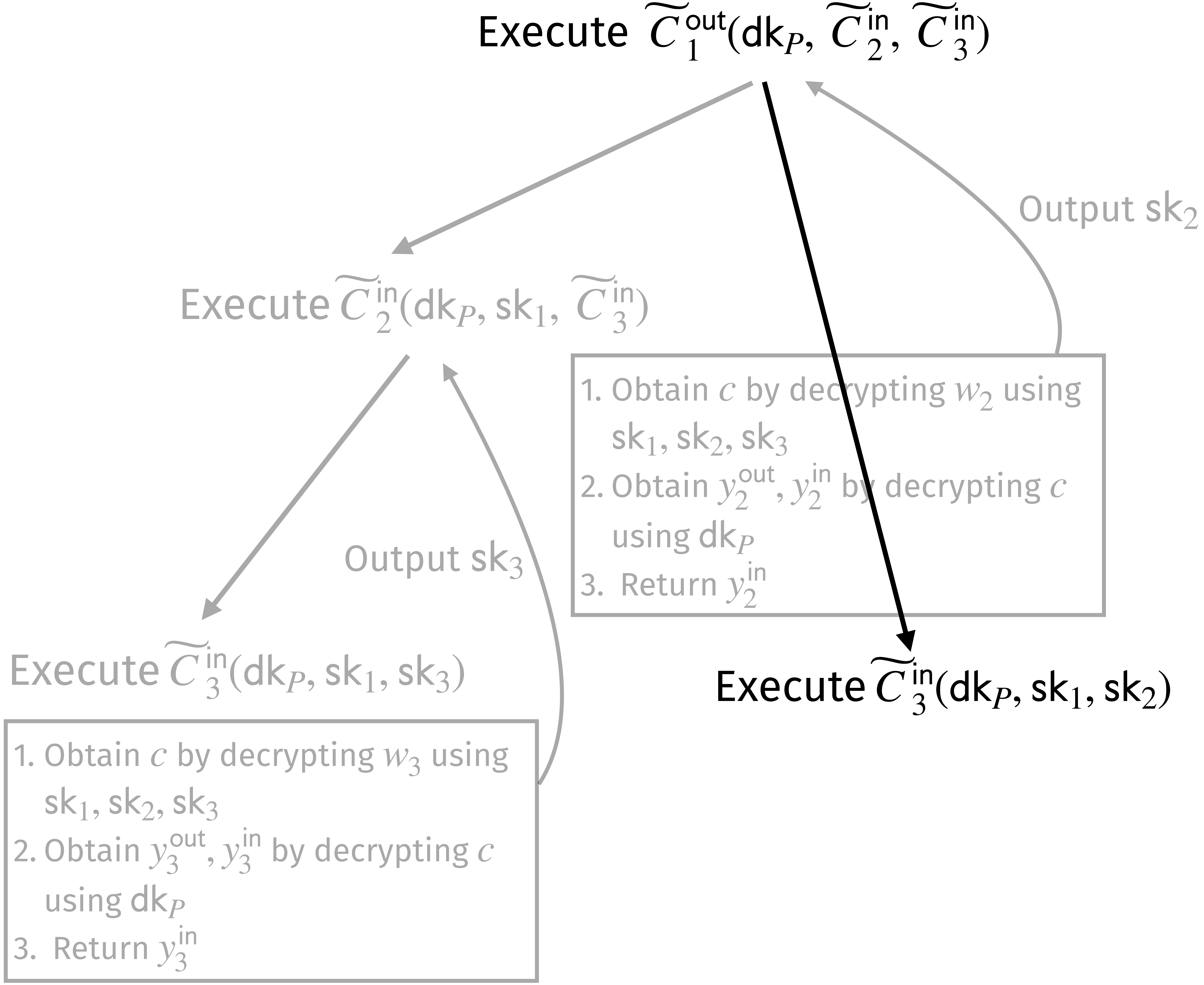


$\widetilde{C}_1^{\text{out}}$

1. Lock y_1^{out}
2. Message m_1
3. Hardcoded info
    1. Secret key sk_1
    2. Cipher w_1

$\widetilde{C}_i^{\text{in}}$ for $i \in [3]$

1. Lock y_i^{in}
2. Message sk_i
3. Hardcoded info
    1. Secret key sk_i
    2. Cipher w_i

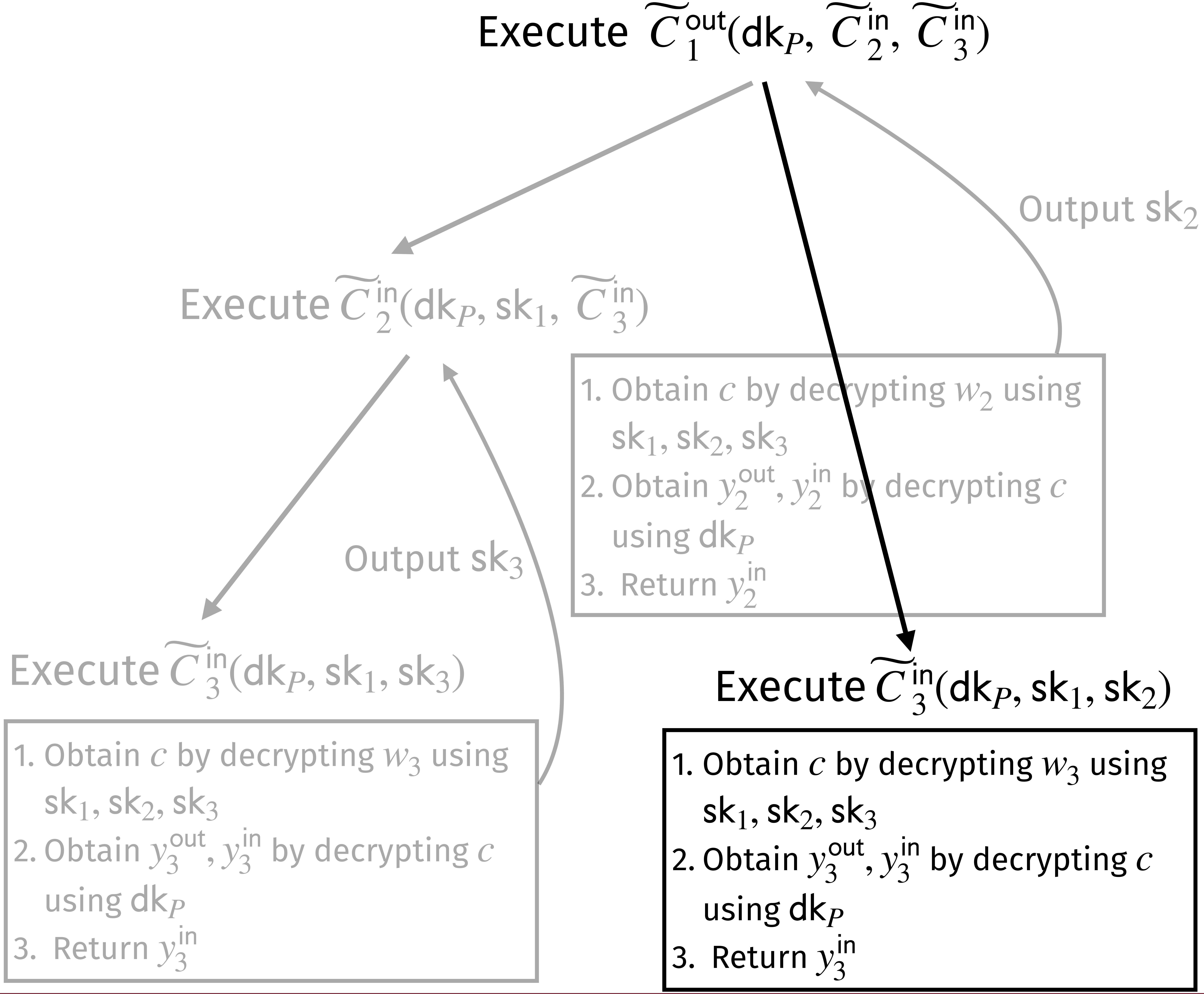


$\widetilde{C}_1^{\text{out}}$

1. Lock y_1^{out}
2. Message m_1
3. Hardcoded info
   1. Secret key sk_1
   2. Cipher w_1

$\widetilde{C}_i^{\text{in}}$ for $i \in [3]$

1. Lock y_i^{in}
2. Message sk_i
3. Hardcoded info
   1. Secret key sk_i
   2. Cipher w_i

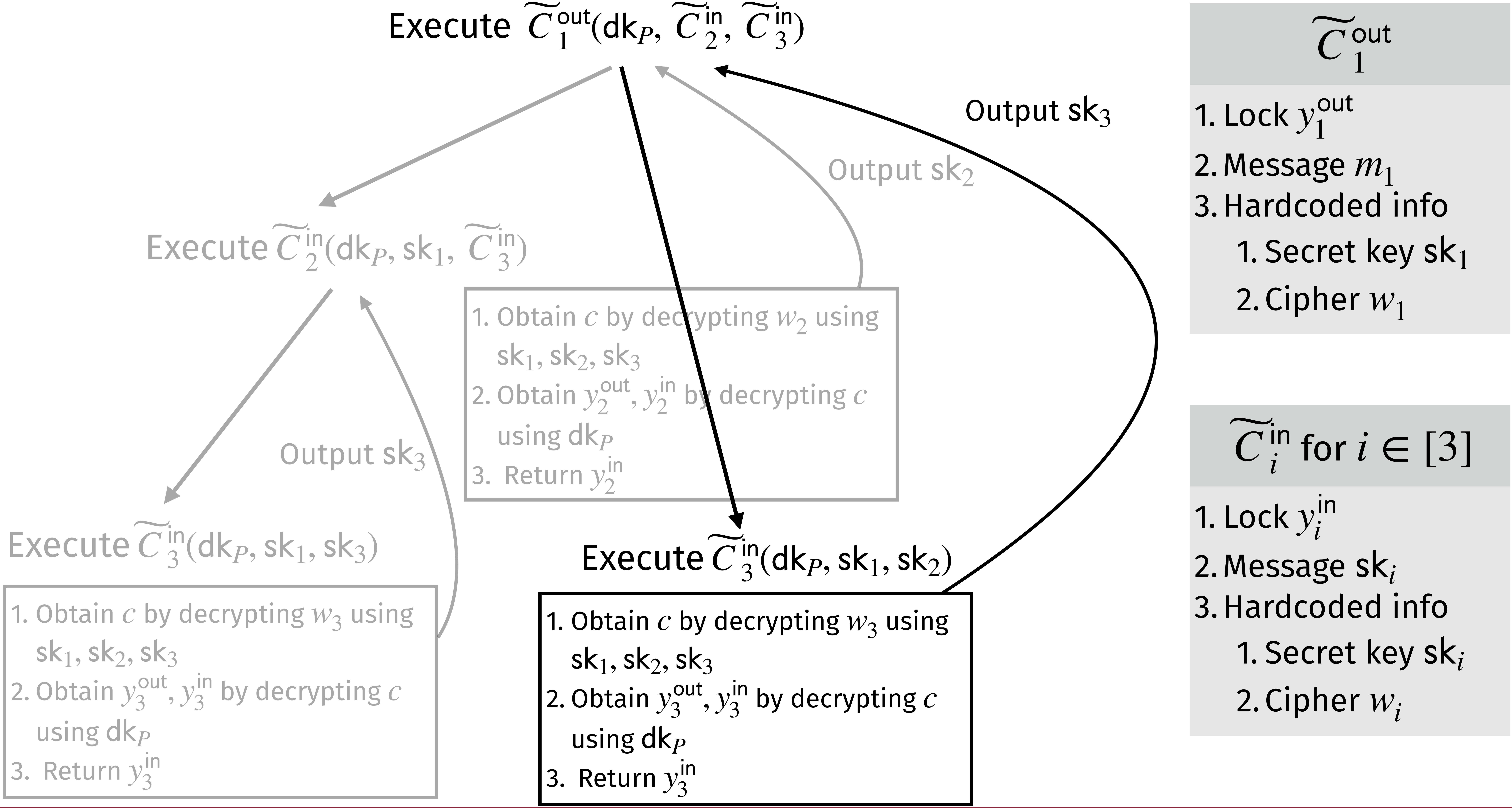


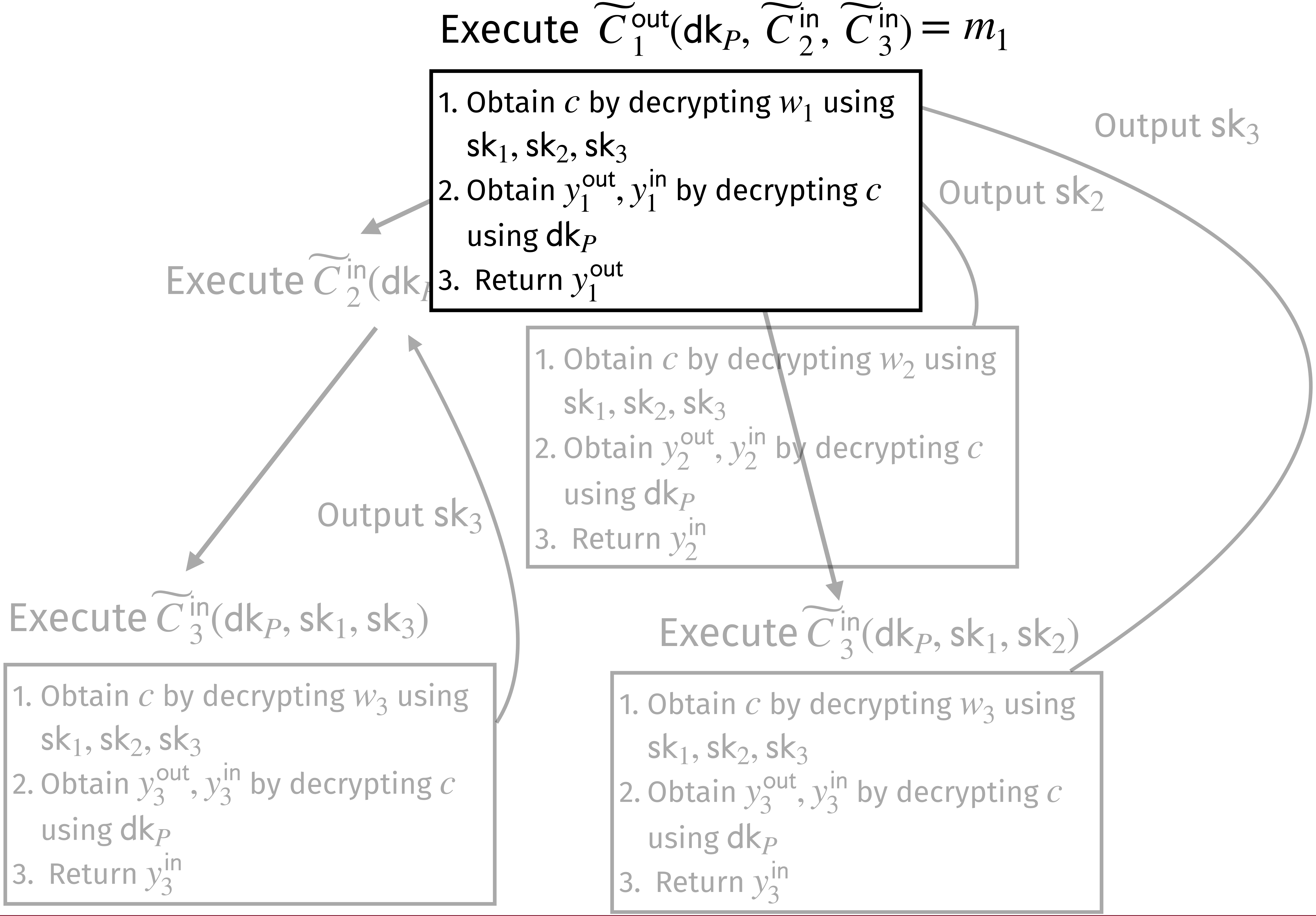
$\widetilde{C}_1^{\text{out}}$

1. Lock y_1^{out}
2. Message m_1
3. Hardcoded info
   1. Secret key sk_1
   2. Cipher w_1

$\widetilde{C}_i^{\text{in}}$ for $i \in [3]$

1. Lock y_i^{in}
2. Message sk_i
3. Hardcoded info
   1. Secret key sk_i
   2. Cipher w_i


Execute $\widetilde{C}_1^{\mathsf{out}}(\mathsf{dk}_P, \widetilde{C}_2^{\mathsf{in}}, \widetilde{C}_3^{\mathsf{in}})$
Output sk_3
Output sk_2
Execute $\widetilde{C}_2^{\mathsf{in}}(\mathsf{dk}_P, \mathsf{sk}_1, \widetilde{C}_3^{\mathsf{in}})$
1. Obtain c by decrypting w_2 using $\mathsf{sk}_1, \mathsf{sk}_2, \mathsf{sk}_3$
2. Obtain $y_2^{\mathsf{out}}, y_2^{\mathsf{in}}$ by decrypting c using dk_P
3. Return y_2^{in}
Output sk_3
Execute $\widetilde{C}_3^{\mathsf{in}}(\mathsf{dk}_P, \mathsf{sk}_1, \mathsf{sk}_3)$
1. Obtain c by decrypting w_3 using $\mathsf{sk}_1, \mathsf{sk}_2, \mathsf{sk}_3$
2. Obtain $y_3^{\mathsf{out}}, y_3^{\mathsf{in}}$ by decrypting c using dk_P
3. Return y_3^{in}
Execute $\widetilde{C}_3^{\mathsf{in}}(\mathsf{dk}_P, \mathsf{sk}_1, \mathsf{sk}_2)$
1. Obtain c by decrypting w_3 using $\mathsf{sk}_1, \mathsf{sk}_2, \mathsf{sk}_3$
2. Obtain $y_3^{\mathsf{out}}, y_3^{\mathsf{in}}$ by decrypting c using dk_P
3. Return y_3^{in}
$\widetilde{C}_1^{\mathsf{out}}$
1. Lock y_1^{out}
2. Message m_1
3. Hardcoded info
1. Secret key sk_1
2. Cipher w_1
$\widetilde{C}_i^{\mathsf{in}}$ for $i \in [3]$
1. Lock y_i^{in}
2. Message sk_i
3. Hardcoded info
1. Secret key sk_i
2. Cipher w_i

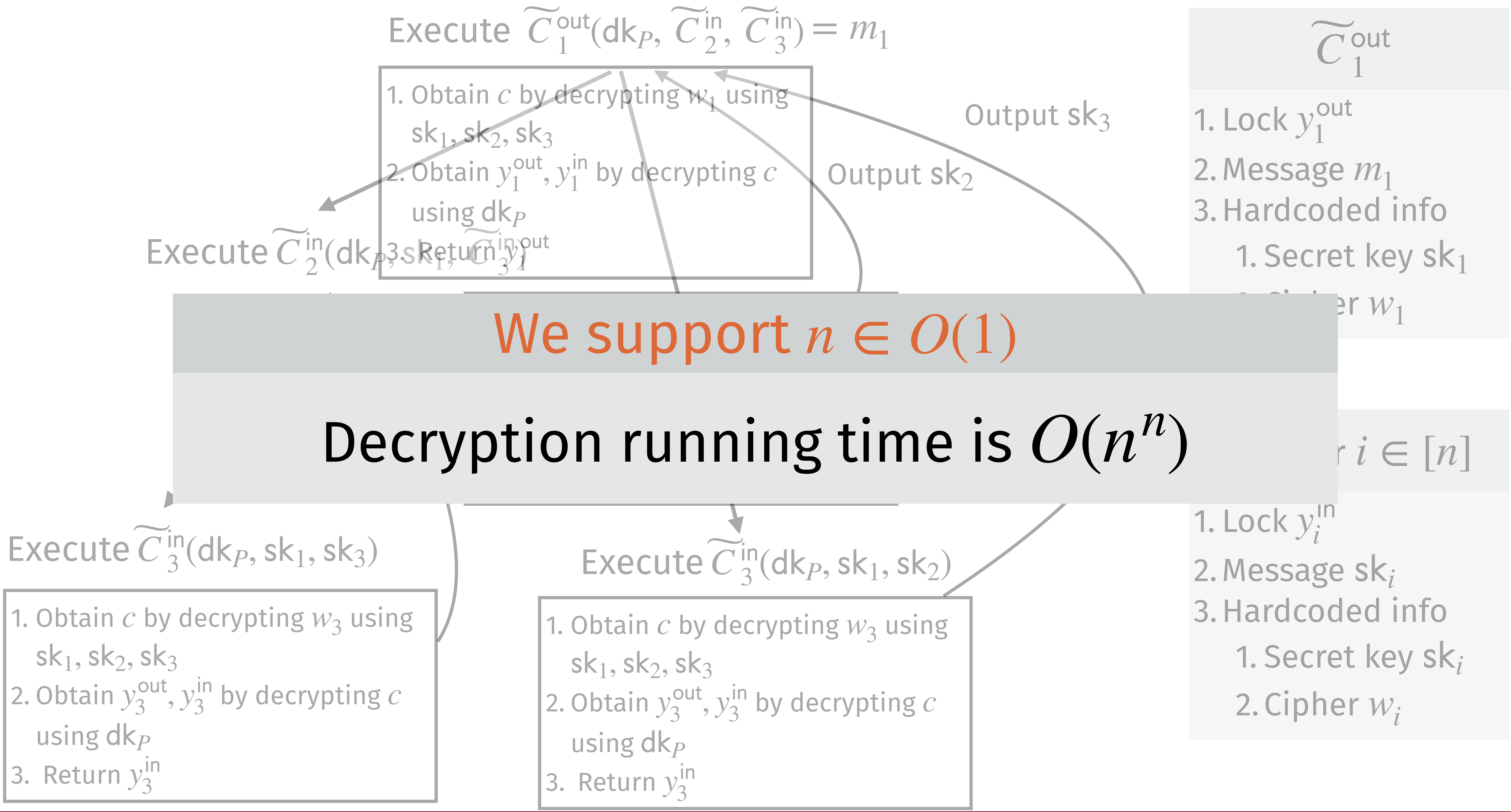


$\widetilde{C}_1^{\text{out}}$

1. Lock y_1^{out}
2. Message m_1
3. Hardcoded info
    1. Secret key sk_1
    2. Cipher w_1

$\widetilde{C}_i^{\text{in}}$ for $i \in [3]$

1. Lock y_i^{in}
2. Message sk_i
3. Hardcoded info
    1. Secret key sk_i
    2. Cipher w_i



We support $n \in O(1)$

Decryption running time is $O(n^n)$