

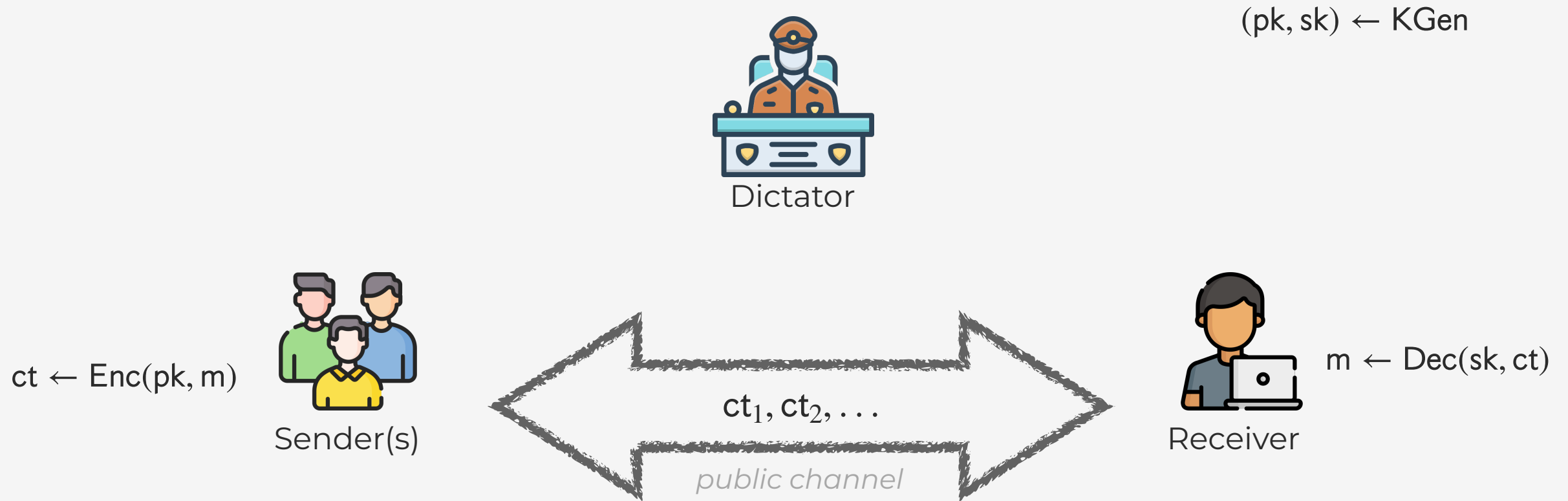
A Unified Treatment of Anamorphic Encryption

Wonseok Choi, Daniel Collins, **Xiangyu Liu**, Roy Stracovsky, Vassilis Zikas

@CRYPTO 2026

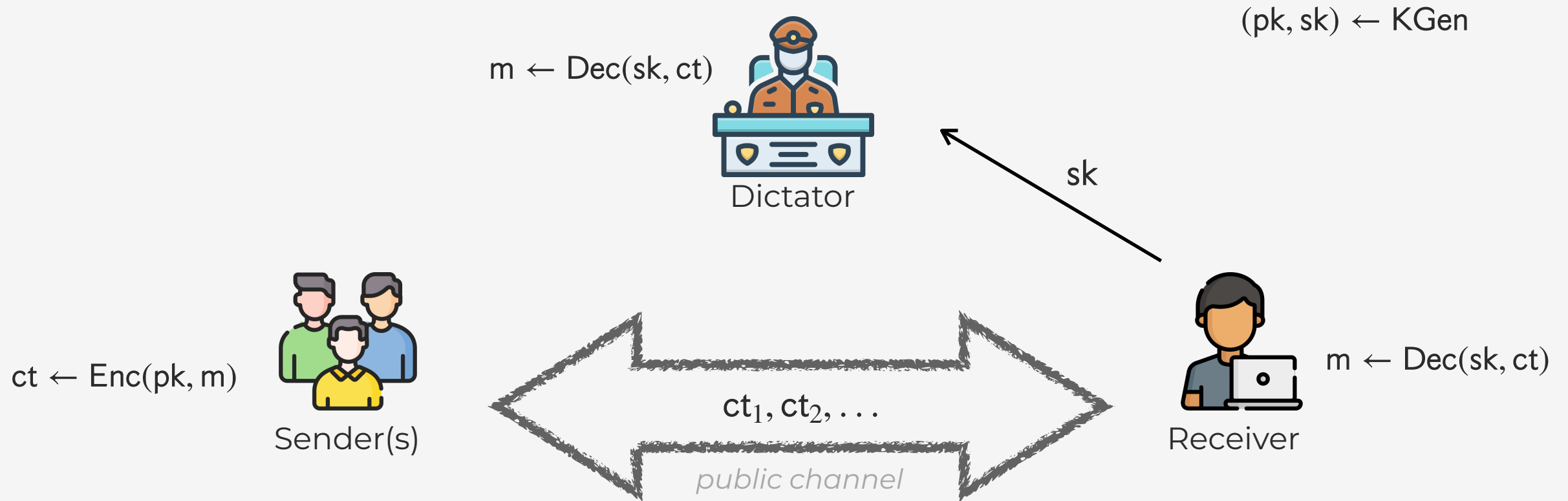


Here Comes the Dictator



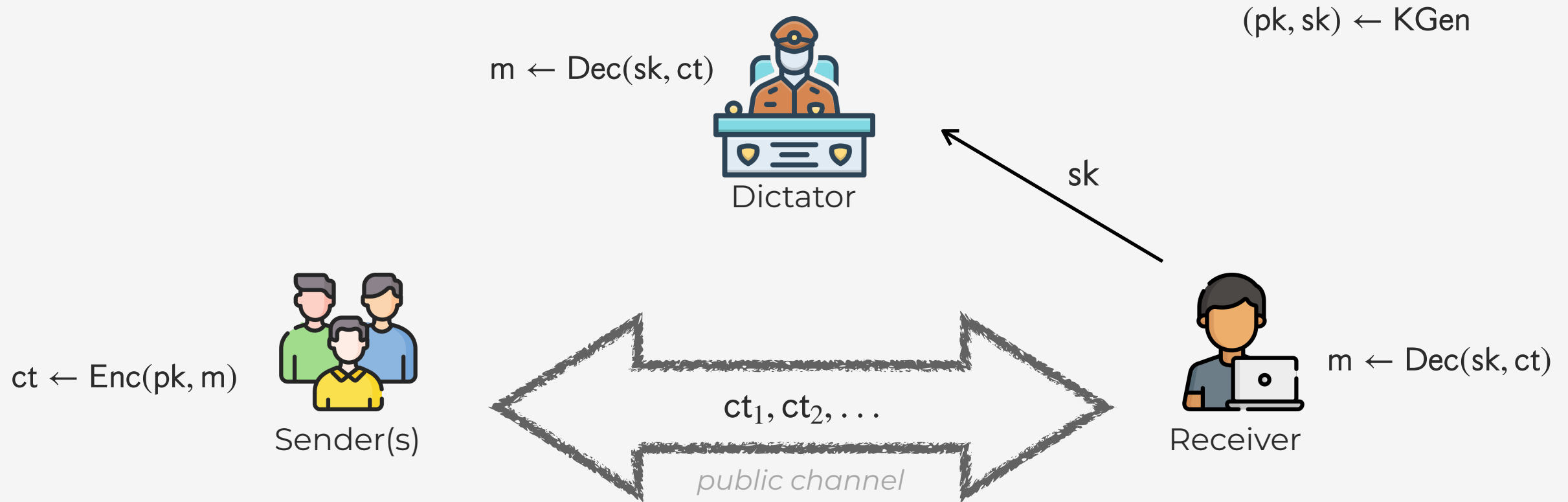


Here Comes the Dictator





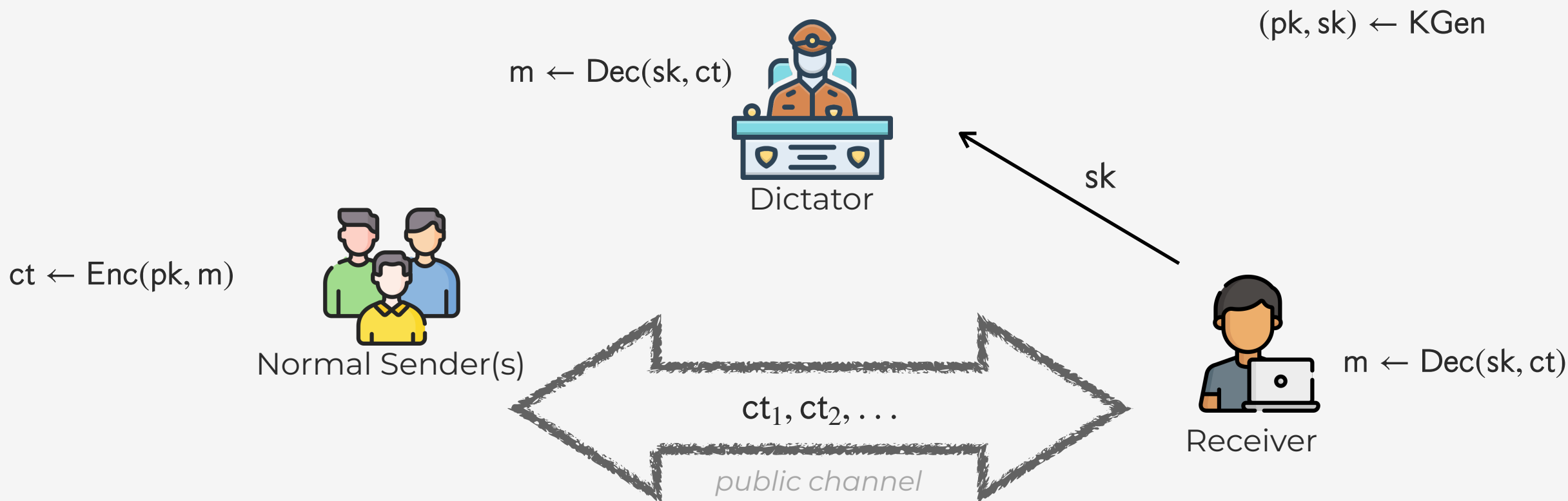
Here Comes the Dictator



Do we still have private communication?



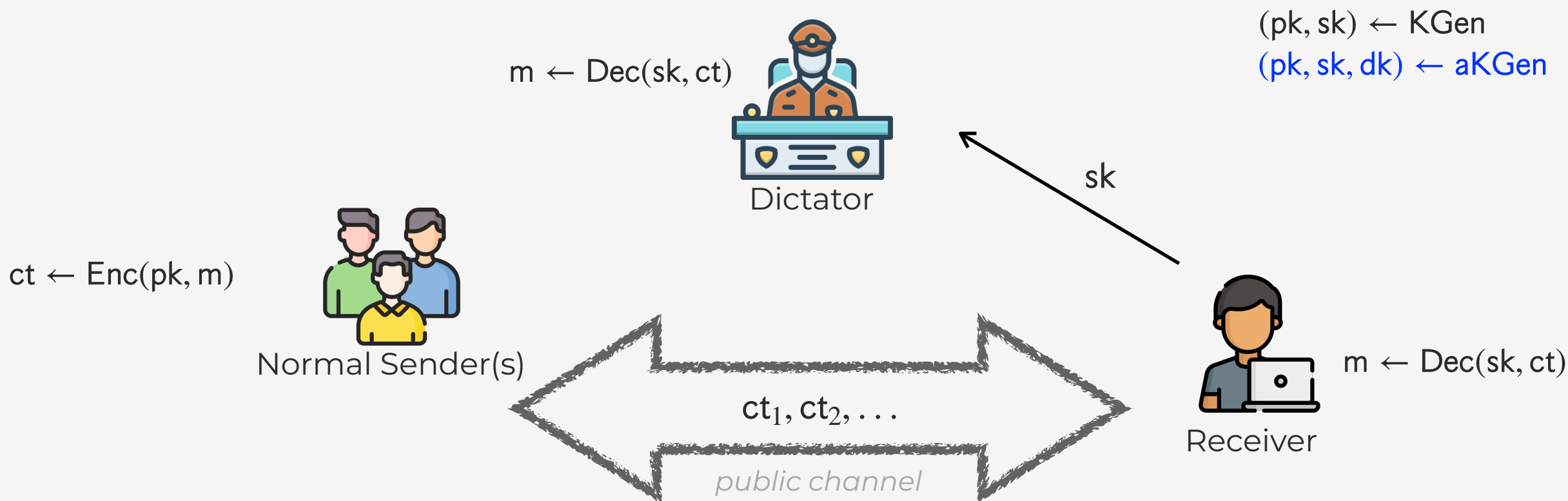
(Receiver) Anamorphic Encryption (AME)



[PPY22] Giuseppe Persiano, Duong Hieu Phan, and Moti Yung. Anamorphic encryption: Private communication against a dictator. (EUROCRYPT)



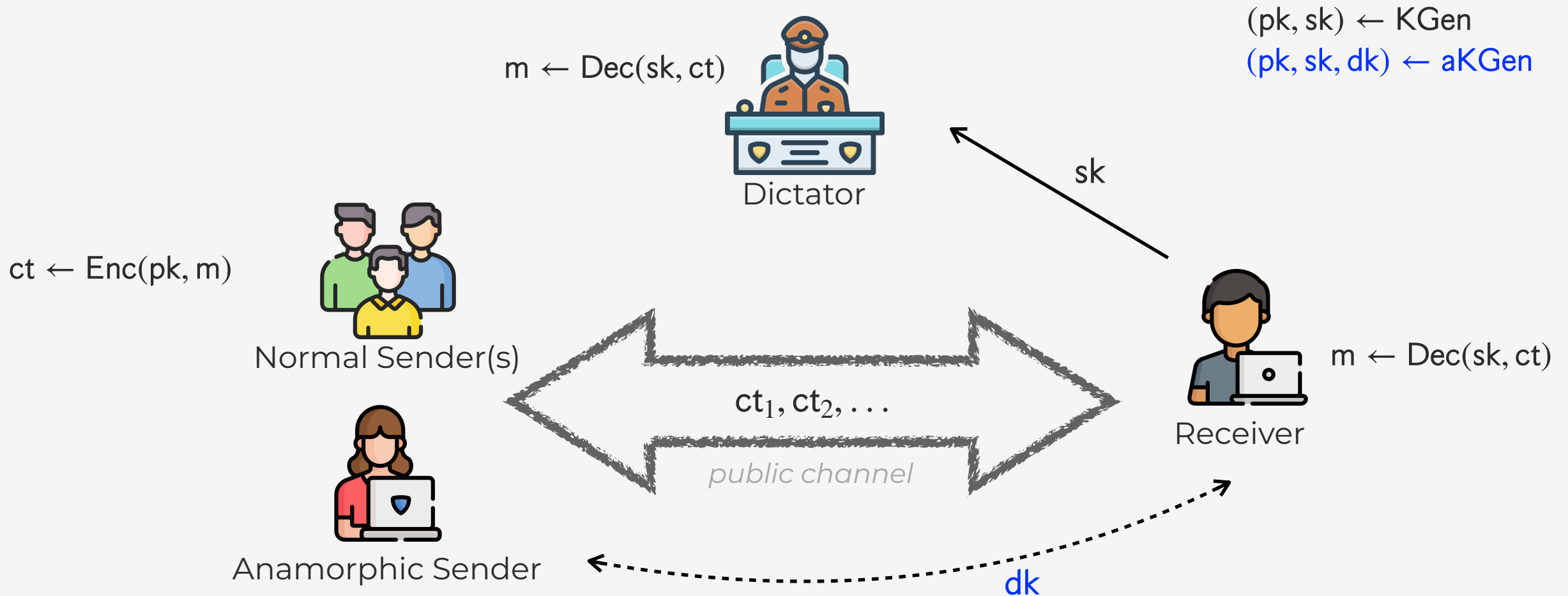
(Receiver) Anamorphic Encryption (AME)



[PPY22] Giuseppe Persiano, Duong Hieu Phan, and Moti Yung. Anamorphic encryption: Private communication against a dictator. (EUROCRYPT)



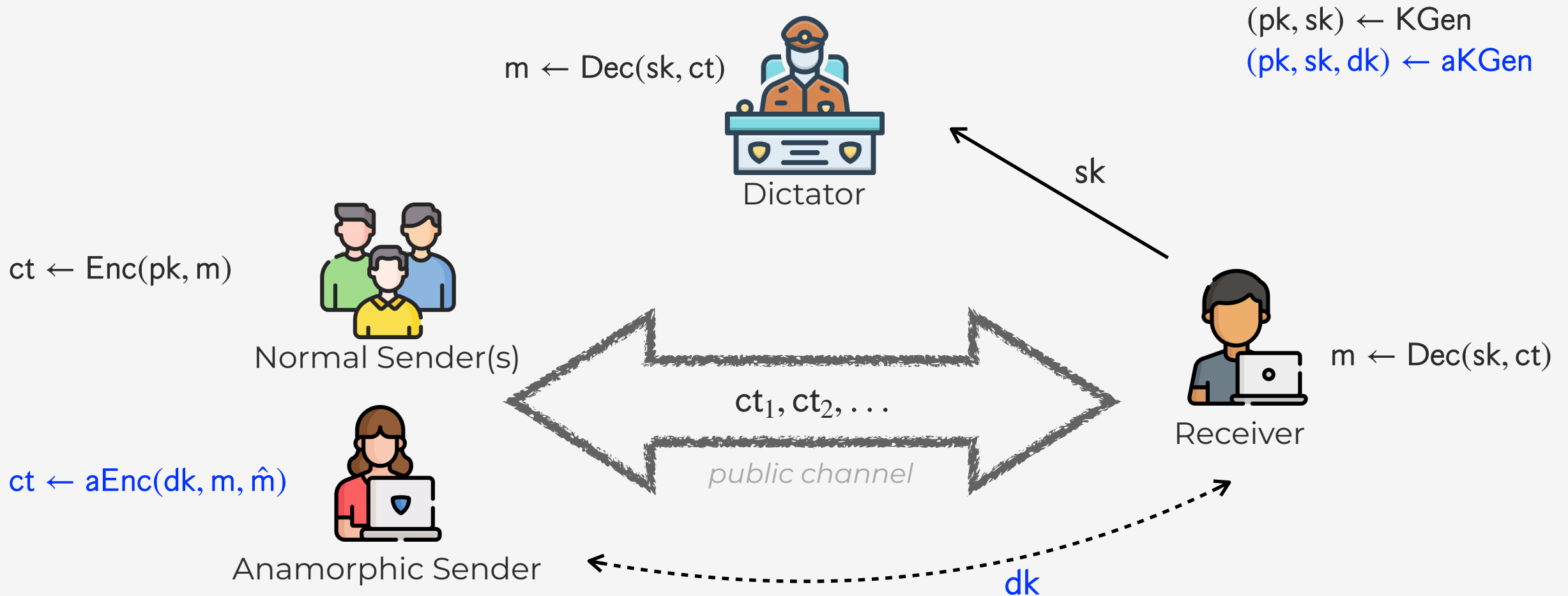
(Receiver) Anamorphic Encryption (AME)



[PPY22] Giuseppe Persiano, Duong Hieu Phan, and Moti Yung. Anamorphic encryption: Private communication against a dictator. (EUROCRYPT)



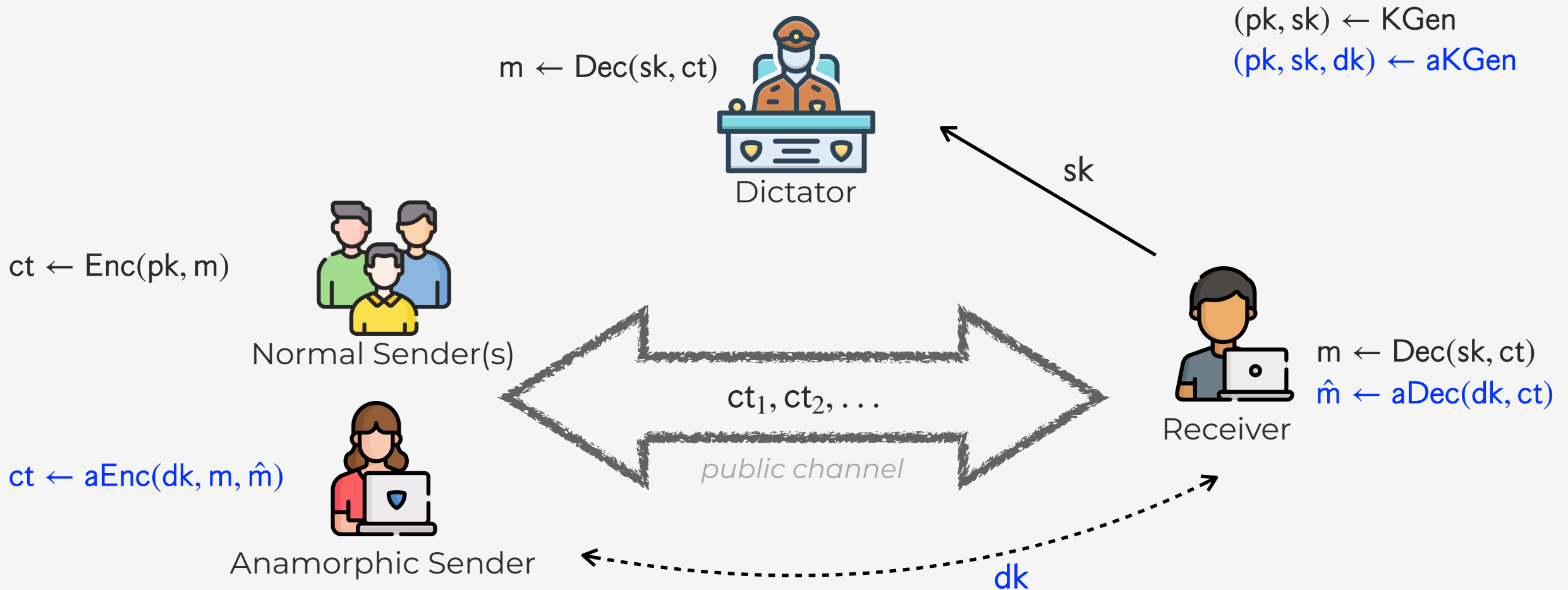
(Receiver) Anamorphic Encryption (AME)



[PPY22] Giuseppe Persiano, Duong Hieu Phan, and Moti Yung. Anamorphic encryption: Private communication against a dictator. (EUROCRYPT)



(Receiver) Anamorphic Encryption (AME)



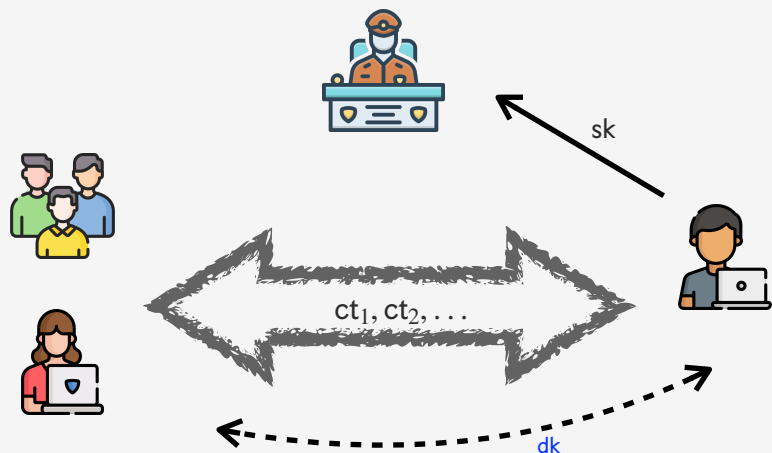
[PPY22] Giuseppe Persiano, Duong Hieu Phan, and Moti Yung. Anamorphic encryption: Private communication against a dictator. (EUROCRYPT)



(Receiver) Anamorphic Encryption (AME)

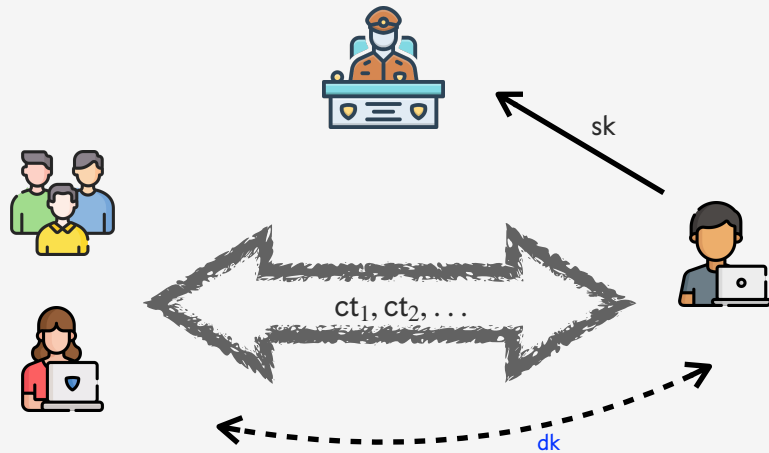


(Receiver) Anamorphic Encryption (AME)





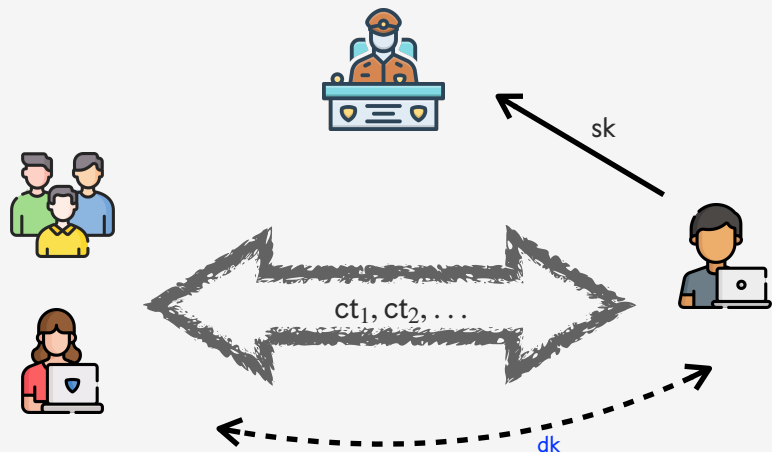
(Receiver) Anamorphic Encryption (AME)



PKE scheme $\begin{cases} (pk, sk) \leftarrow \text{KGen} \\ ct \leftarrow \text{Enc}(pk, m) \\ m \leftarrow \text{Dec}(sk, ct) \end{cases}$



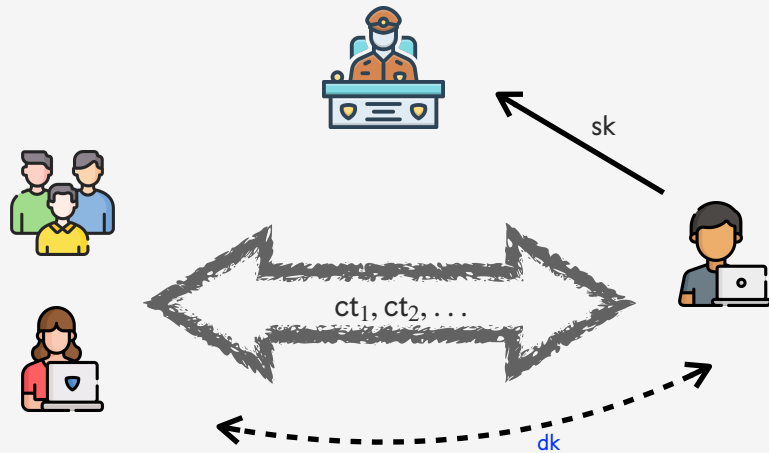
(Receiver) Anamorphic Encryption (AME)



$$\text{AME} \begin{cases} \text{PKE scheme} & \begin{cases} (pk, sk) \leftarrow \text{KGen} \\ ct \leftarrow \text{Enc}(pk, m) \\ m \leftarrow \text{Dec}(sk, ct) \end{cases} \\ \text{AME triplet} & \begin{cases} (pk, sk, dk) \leftarrow \text{aKGen} \\ ct \leftarrow \text{aEnc}(pk, dk, m, \hat{m}) \\ \hat{m} \leftarrow \text{aDec}(sk, dk, ct) \end{cases} \end{cases}$$



(Receiver) Anamorphic Encryption (AME)

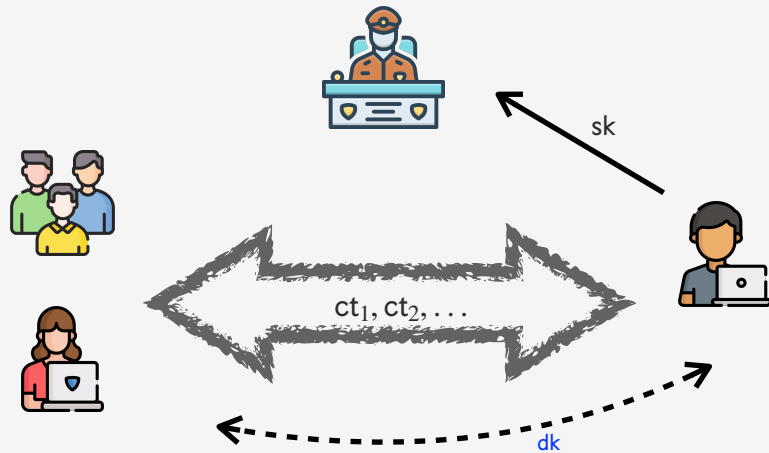


$$\text{AME} \begin{cases} \text{PKE scheme} & \begin{cases} (pk, sk) \leftarrow \text{KGen} \\ ct \leftarrow \text{Enc}(pk, m) \\ m \leftarrow \text{Dec}(sk, ct) \end{cases} \\ \text{AME triplet} & \begin{cases} (pk, sk, dk) \leftarrow \text{aKGen} \\ ct \leftarrow \text{aEnc}(pk, dk, m, \hat{m}) \\ \hat{m} \leftarrow \text{aDec}(sk, dk, ct) \end{cases} \end{cases}$$

[PPY22] AME is secure, if



(Receiver) Anamorphic Encryption (AME)



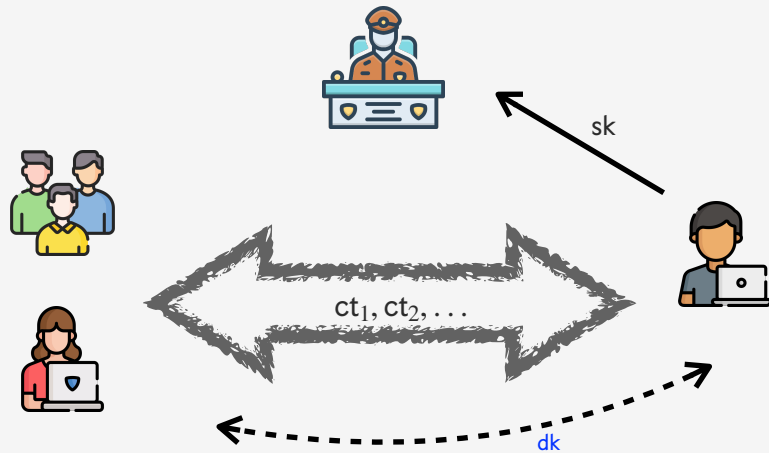
$$\text{AME} \begin{cases} \text{PKE scheme} & \begin{cases} (pk, sk) \leftarrow \text{KGen} \\ ct \leftarrow \text{Enc}(pk, m) \\ m \leftarrow \text{Dec}(sk, ct) \end{cases} \\ \text{AME triplet} & \begin{cases} (pk, sk, dk) \leftarrow \text{aKGen} \\ ct \leftarrow \text{aEnc}(pk, dk, m, \hat{m}) \\ \hat{m} \leftarrow \text{aDec}(sk, dk, ct) \end{cases} \end{cases}$$

[PPY22] AME is secure, if

1. (underlying) PKE is IND-CPA;



(Receiver) Anamorphic Encryption (AME)



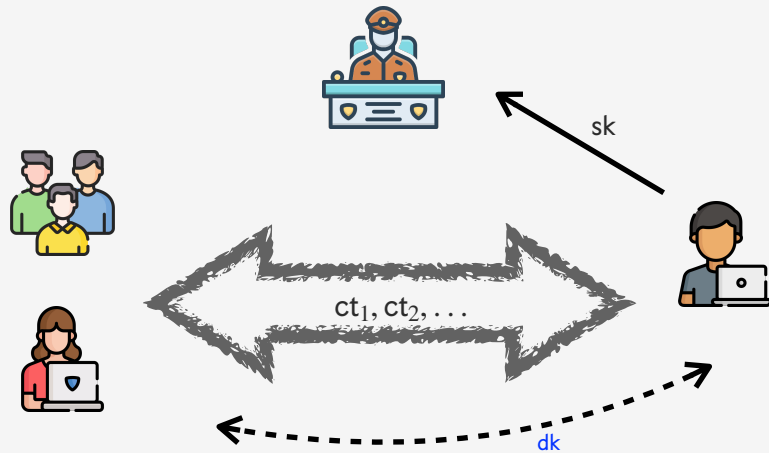
$$\text{AME} \begin{cases} \text{PKE scheme} \begin{cases} (pk, sk) \leftarrow \text{KGen} \\ ct \leftarrow \text{Enc}(pk, m) \\ m \leftarrow \text{Dec}(sk, ct) \end{cases} \\ \text{AME triplet} \begin{cases} (pk, sk, dk) \leftarrow \text{aKGen} \\ ct \leftarrow \text{aEnc}(pk, dk, m, \hat{m}) \\ \hat{m} \leftarrow \text{aDec}(sk, dk, ct) \end{cases} \end{cases}$$

[PPY22] AME is secure, if

1. (underlying) PKE is IND-CPA;
2. $(\text{aEnc}, \text{aDec})$ forms a symmetric encryption on $\hat{m}$;



(Receiver) Anamorphic Encryption (AME)



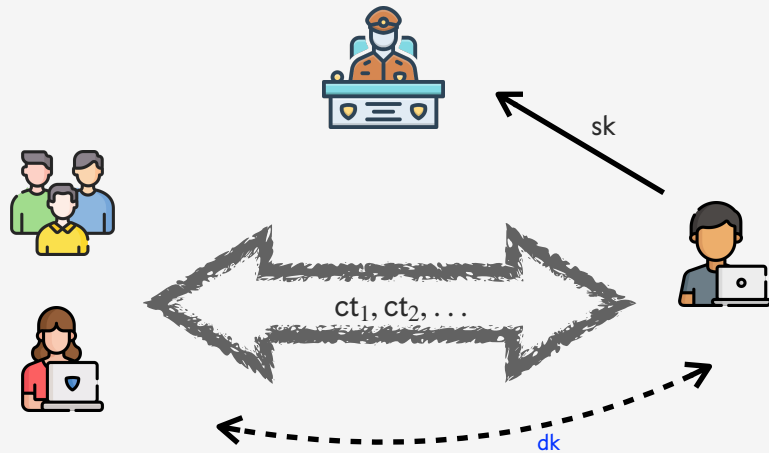
$$\text{AME} \begin{cases} \text{PKE scheme} \begin{cases} (pk, sk) \leftarrow \text{KGen} \\ ct \leftarrow \text{Enc}(pk, m) \\ m \leftarrow \text{Dec}(sk, ct) \end{cases} \\ \text{AME triplet} \begin{cases} (pk, sk, dk) \leftarrow \text{aKGen} \\ ct \leftarrow \text{aEnc}(pk, dk, m, \hat{m}) \\ \hat{m} \leftarrow \text{aDec}(sk, dk, ct) \end{cases} \end{cases}$$

[PPY22] AME is secure, if

1. (underlying) PKE is IND-CPA;
2. $(\text{aEnc}, \text{aDec})$ forms a symmetric encryption on $\hat{m}$;
3. the adversary (dictator) cannot detect the use of AME, i.e.,



(Receiver) Anamorphic Encryption (AME)



$$\text{AME} \begin{cases} \text{PKE scheme} & \begin{cases} (pk, sk) \leftarrow \text{KGen} \\ ct \leftarrow \text{Enc}(pk, m) \\ m \leftarrow \text{Dec}(sk, ct) \end{cases} \\ \text{AME triplet} & \begin{cases} (pk, sk, dk) \leftarrow \text{aKGen} \\ ct \leftarrow \text{aEnc}(pk, dk, m, \hat{m}) \\ \hat{m} \leftarrow \text{aDec}(sk, dk, ct) \end{cases} \end{cases}$$

[PPY22] AME is secure, if

1. (underlying) PKE is IND-CPA;
2. $(\text{aEnc}, \text{aDec})$ forms a symmetric encryption on $\hat{m}$;
3. the adversary (dictator) cannot detect the use of AME, i.e.,

Normal Mode

(pk, sk) from KGen
 $\{ct \text{ from } \text{Enc}(m)\}$



Anamorphic Mode

(pk, sk) from aKGen
 $\{ct \text{ from } \text{aEnc}(m, \hat{m})\}$

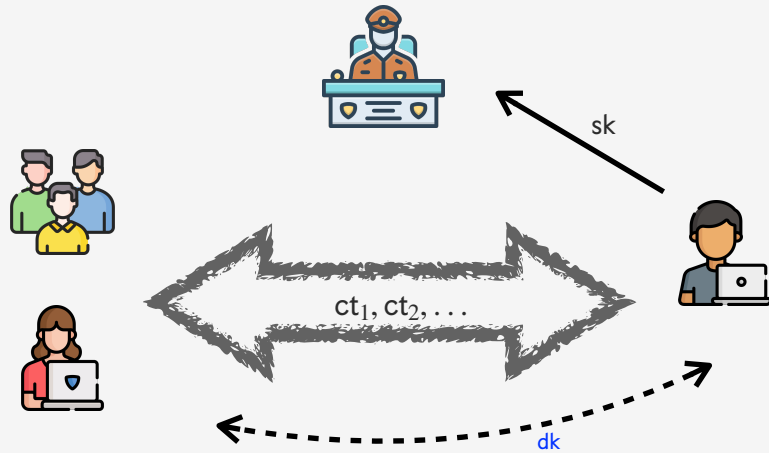


Follow-up Works

- [BGH+24] *Fabio Banfi, Konstantin Gegier, Martin Hirt, Ueli Maurer, and Guilherme Rito. Anamorphic encryption, revisited.*
- [CGM24] *Dario Catalano, Emanuele Giunta, and Francesco Migliaro. Anamorphic encryption: new constructions and homomorphic realizations.*
- [KPP+23] *Miroslaw Kutylowski, Giuseppe Persiano, Duong Hieu Phan, Moti Yung, and Marcin Zawada. The self-anti-censorship nature of encryption: On the prevalence of anamorphic cryptography.*
- [PPY24] *Giuseppe Persiano, Duong Hieu Phan, and Moti Yung. Public-key anamorphism in (CCA-secure) publickey encryption and beyond.*
- [CGM25] *Dario Catalano, Emanuele Giunta, and Francesco Migliaro. Generic anamorphic encryption, revisited: New limitations and constructions.*
- [DL26] *Amit Deo and Benoît Libert. Fully asymmetric anamorphic homomorphic encryption from LWE.*
- [BPRS26] *Shalini Banerjee, Tapas Pal, Andy Rupp, and Daniel Slamanig. Simple public key anamorphic encryption and signature using multi-message extensions.*
- *and more...*

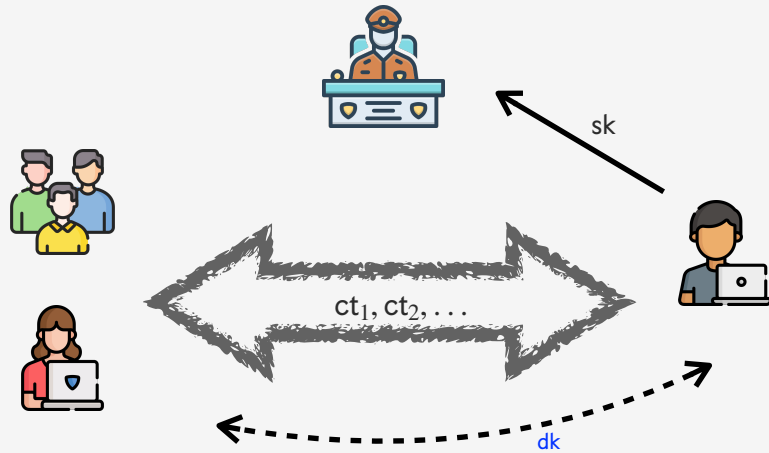


[BGH+24]: Stateful AME Extension, Robustness





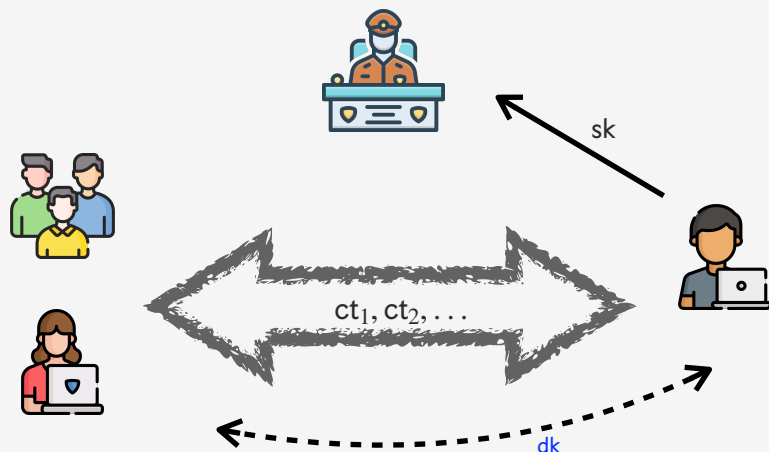
[BGH+24]: Stateful AME Extension, Robustness



PKE scheme $\begin{cases} (pk, sk) \leftarrow \text{KGen} \\ ct \leftarrow \text{Enc}(pk, m) \\ m \leftarrow \text{Dec}(sk, ct) \end{cases}$



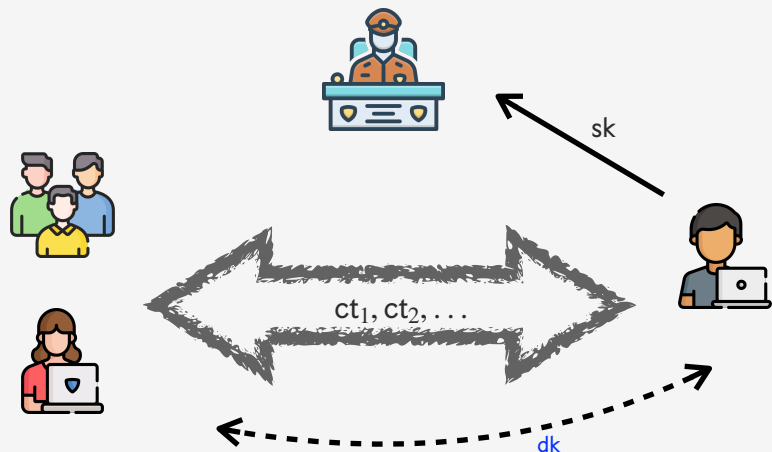
[BGH+24]: Stateful AME Extension, Robustness



$$\text{AME} \begin{cases} \text{PKE scheme} & \begin{cases} (pk, sk) \leftarrow \text{KGen} \\ ct \leftarrow \text{Enc}(pk, m) \\ m \leftarrow \text{Dec}(sk, ct) \end{cases} \\ \text{AME Extension} & \begin{cases} dk \leftarrow \text{aKGen}(pk, sk) \\ (ct, st') \leftarrow \text{aEnc}(pk, dk, m, \hat{m}; st) \\ (\hat{m}, st') \leftarrow \text{aDec}(sk, dk, ct; st) \end{cases} \end{cases}$$



[BGH+24]: Stateful AME Extension, Robustness



$$\text{AME} \begin{cases} \text{PKE scheme} & \begin{cases} (pk, sk) \leftarrow \text{KGen} \\ ct \leftarrow \text{Enc}(pk, m) \\ m \leftarrow \text{Dec}(sk, ct) \end{cases} \\ \text{AME Extension} & \begin{cases} dk \leftarrow \text{aKGen}(pk, sk) \\ (ct, st') \leftarrow \text{aEnc}(pk, dk, m, \hat{m}; st) \\ (\hat{m}, st') \leftarrow \text{aDec}(sk, dk, ct; st) \end{cases} \end{cases}$$

Normal Mode

(pk, sk) from KGen
 $\{ct \text{ from } \text{Enc}(m)\}$

Anamorphic Mode

(pk, sk) from aKGen
 $\{ct \text{ from } \text{aEnc}(m, \hat{m})\}$

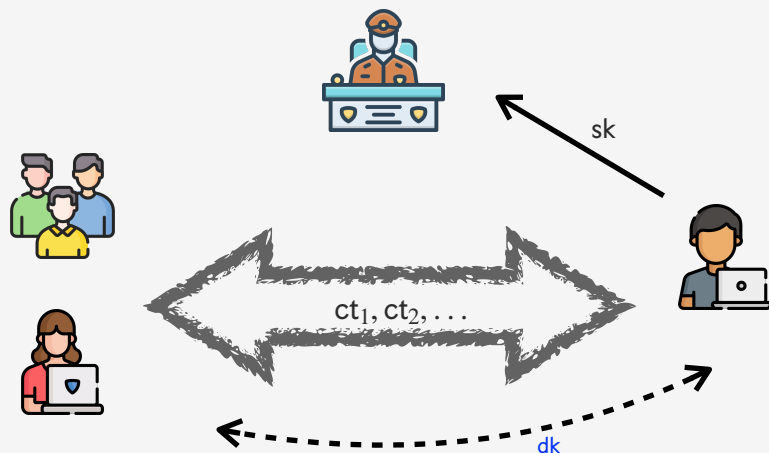
$\approx$

Robustness

$$\text{aDec}(dk, \text{Enc}(m)) \approx \perp$$

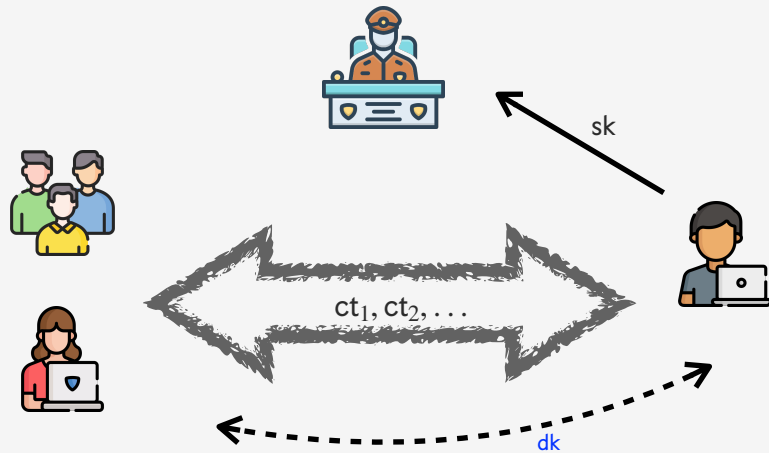


[CGM24]: Asymmetric AME





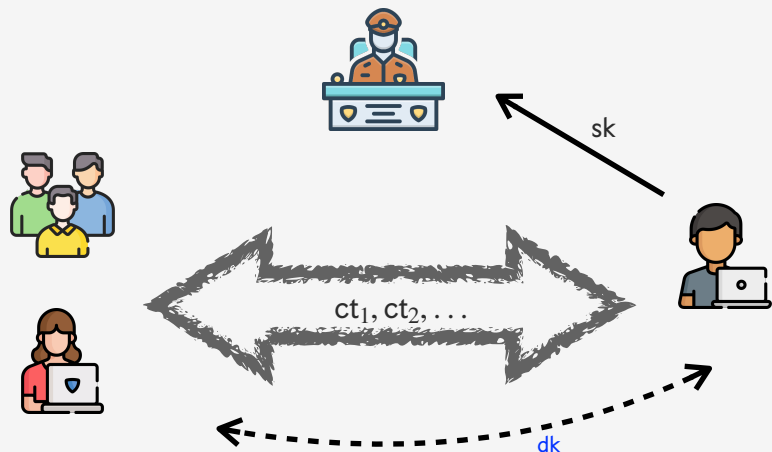
[CGM24]: Asymmetric AME



PKE scheme $\begin{cases} (pk, sk) \leftarrow \text{KGen} \\ ct \leftarrow \text{Enc}(pk, m) \\ m \leftarrow \text{Dec}(sk, ct) \end{cases}$



[CGM24]: Asymmetric AME



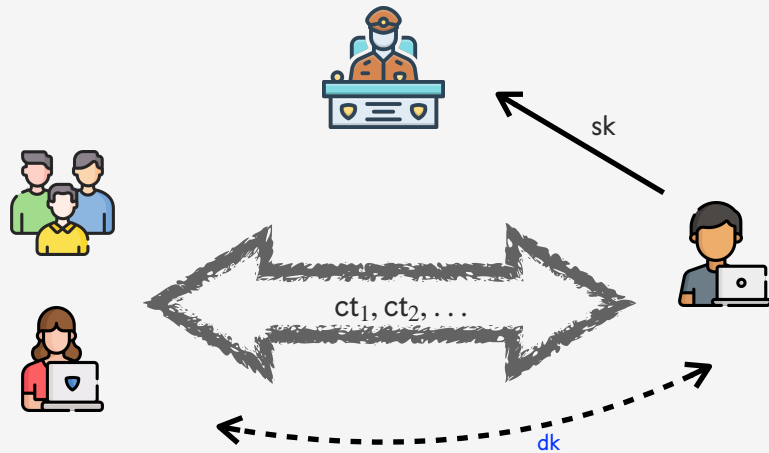
AME {

PKE scheme $\begin{cases} (pk, sk) \leftarrow \text{KGen} \\ ct \leftarrow \text{Enc}(pk, m) \\ m \leftarrow \text{Dec}(sk, ct) \end{cases}$

Asymmetric
AME Triplet $\begin{cases} (pk, sk, dk, tk) \leftarrow \text{aKGen} \\ ct \leftarrow \text{aEnc}(pk, dk, m, \hat{m}) \\ \hat{m} \leftarrow \text{aDec}(sk, dk, tk, ct) \end{cases}$



[CGM24]: Asymmetric AME



$$\text{AME} \begin{cases} \text{PKE scheme} & \begin{cases} (pk, sk) \leftarrow \text{KGen} \\ ct \leftarrow \text{Enc}(pk, m) \\ m \leftarrow \text{Dec}(sk, ct) \end{cases} \\ \text{Asymmetric AME Triplet} & \begin{cases} (pk, sk, dk, tk) \leftarrow \text{aKGen} \\ ct \leftarrow \text{aEnc}(pk, dk, m, \hat{m}) \\ \hat{m} \leftarrow \text{aDec}(sk, dk, tk, ct) \end{cases} \end{cases}$$

Normal Mode

(pk, sk) from KGen
 $\{ct \text{ from } \text{Enc}(m)\}$

Anamorphic Mode

(pk, sk) from aKGen
 $\{ct \text{ from } \text{aEnc}(m, \hat{m})\}$

$\approx$

Fully Asymmetric Property

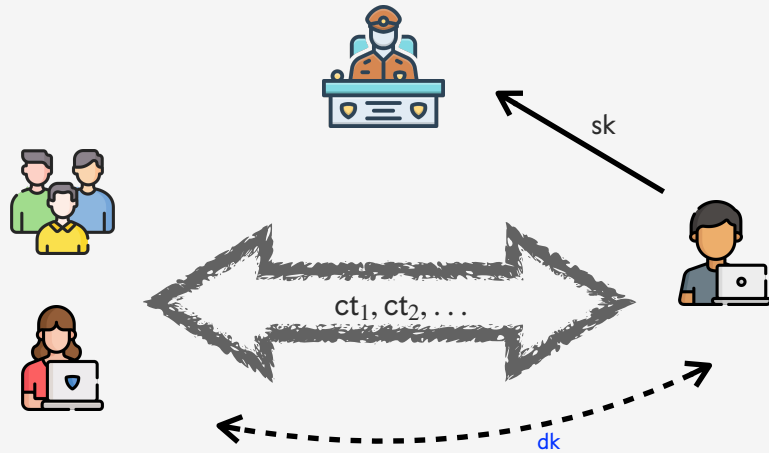
- security against anamorphic sender(s)

given (pk, dk)

$$\text{aEnc}(pk, dk, m_0, \hat{m}_0) \approx \text{aEnc}(pk, dk, m_1, \hat{m}_1)$$

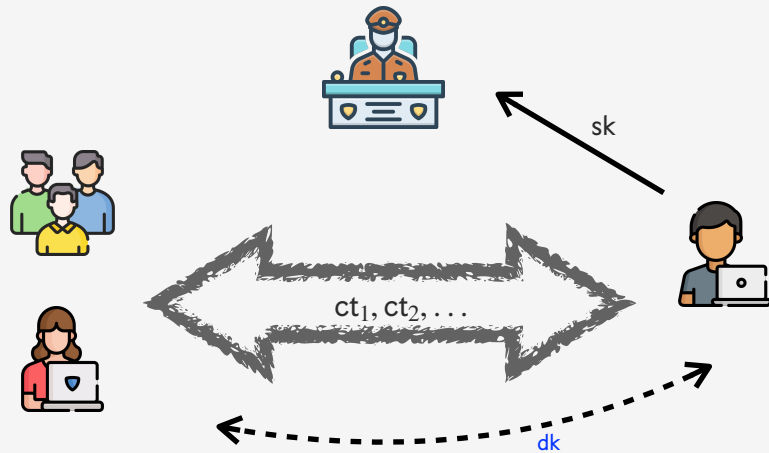


[BPRS26]: μ -Message Asymmetric AME





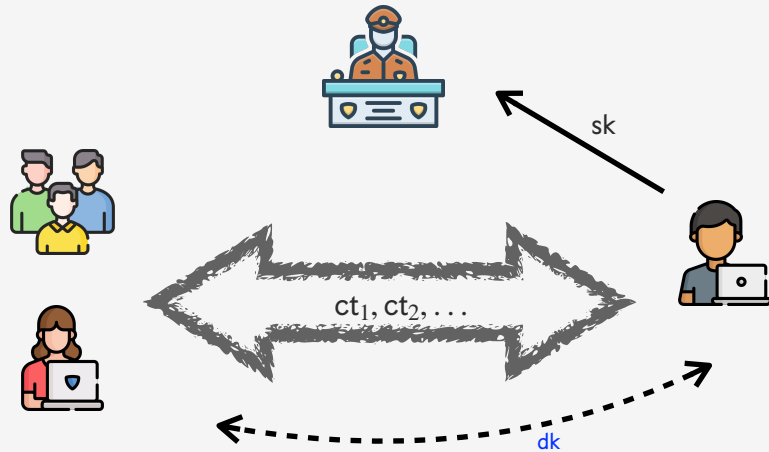
[BPRS26]: μ -Message Asymmetric AME



$$\text{AME} \begin{cases} \text{PKE scheme} & \begin{cases} (pk, sk) \leftarrow \text{KGen} \\ ct \leftarrow \text{Enc}(pk, m) \\ m \leftarrow \text{Dec}(sk, ct) \end{cases} \\ \text{AME Triplet} & \begin{cases} (pk, sk, dk, tk) \leftarrow \text{aKGen} \\ ct \leftarrow \text{aEnc}(pk, dk, m, \hat{m}) \\ \hat{m} \leftarrow \text{aDec}(sk, tk, ct) \end{cases} \end{cases}$$



[BPRS26]: μ -Message Asymmetric AME



$$\text{AME} \begin{cases} \text{PKE scheme} & \begin{cases} (pk, sk) \leftarrow \text{KGen} \\ ct \leftarrow \text{Enc}(pk, m) \\ m \leftarrow \text{Dec}(sk, ct) \end{cases} \\ \text{AME Triplet} & \begin{cases} (pk, sk, dk, tk) \leftarrow \text{aKGen} \\ ct \leftarrow \text{aEnc}(pk, dk, m, \hat{m}) \\ \hat{m} \leftarrow \text{aDec}(sk, tk, ct) \end{cases} \end{cases}$$

Normal Mode

(pk, sk) from KGen
 $\{ct \text{ from } \text{Enc}(m)\}$

Anamorphic Mode

(pk, sk) from aKGen
 $\{ct \text{ from } \text{aEnc}(m, \hat{m})\}$

$\approx$

Strong IND-CPA

- security against collusion

given (pk, sk, dk)

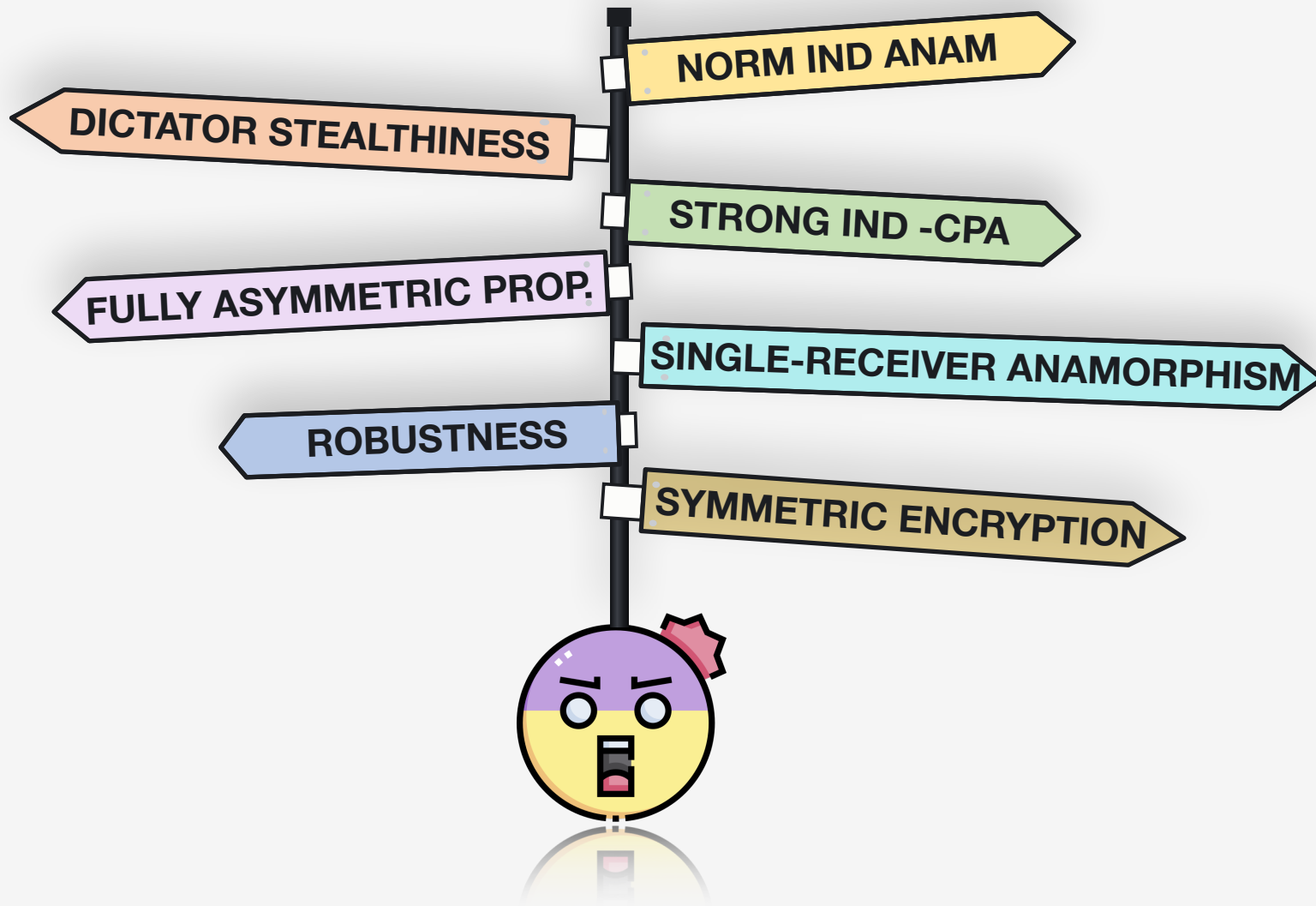
$$\text{aEnc}(pk, dk, m, \hat{m}_0) \approx \text{aEnc}(pk, dk, m, \hat{m}_1)$$



Get Lost in AME

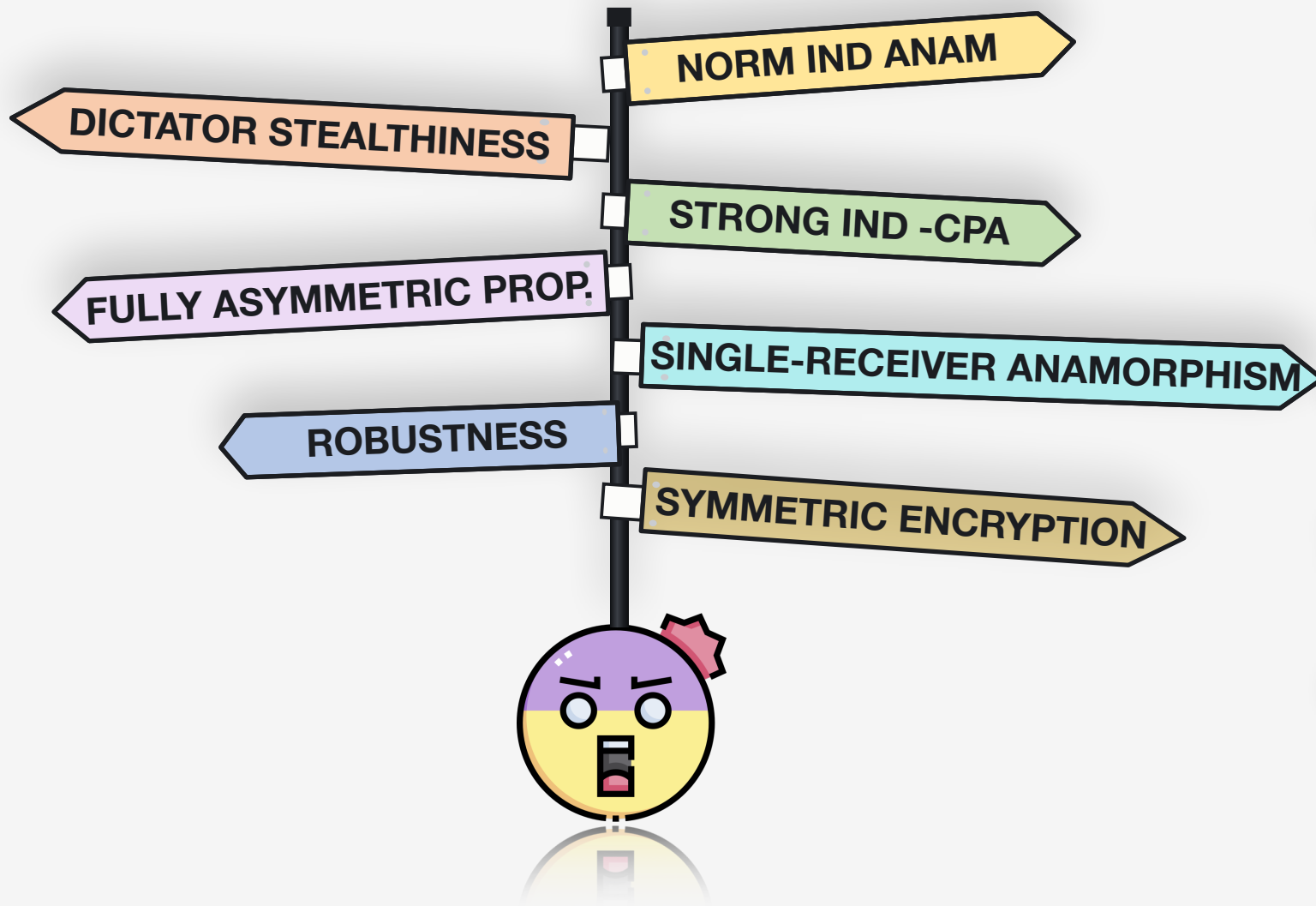


Get Lost in AME





Get Lost in AME



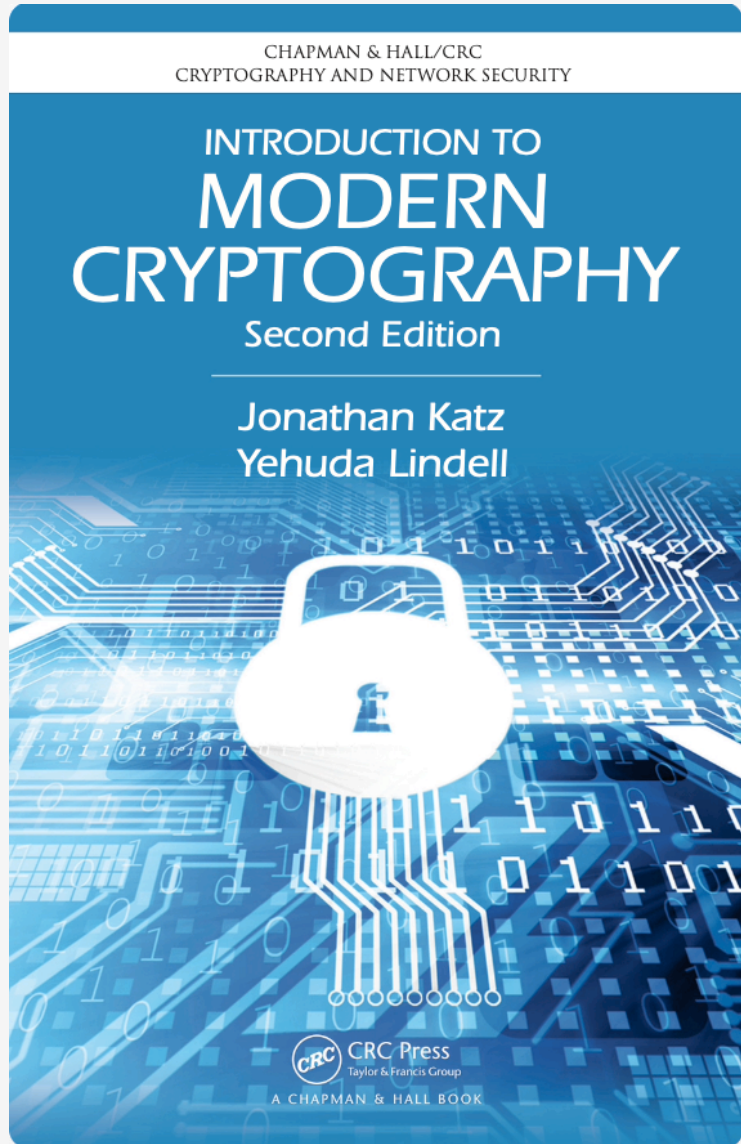
WHAT IS THE
CORRECT
SECURITY NOTION
OF AME?



Back to Our First Lesson

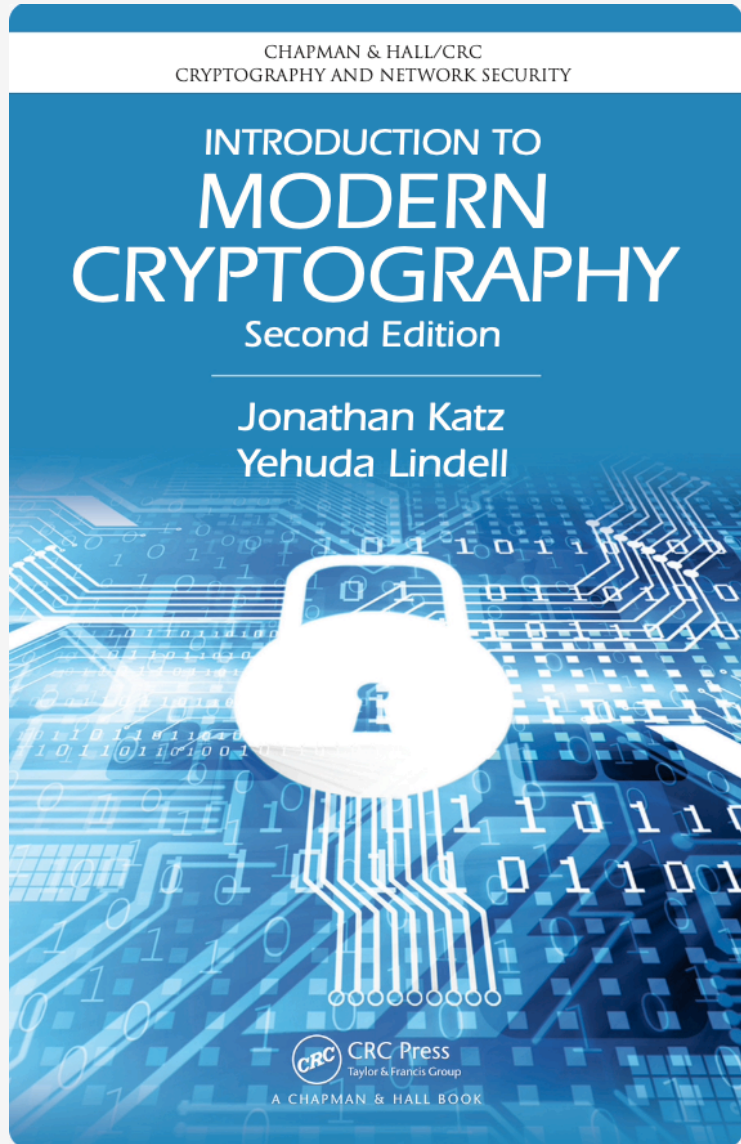


Back to Our First Lesson





Back to Our First Lesson



Contents

Preface	xv
I Introduction and Classical Cryptography	
1 Introduction	3
1.1 Cryptography and Modern Cryptography	3
1.2 The Setting of Private-Key Encryption	4
1.3 Historical Ciphers and Their Cryptanalysis	8
1.4 Principles of Modern Cryptography	16
1.4.1 Principle 1 – Formal Definitions	17
1.4.2 Principle 2 – Precise Assumptions	20
1.4.3 Principle 3 – Proofs of Security	22
1.4.4 Provable Security and Real-World Security	22
References and Additional Reading	23
Exercises	24



Back to Our First Lesson

CHAPMAN & HALL/CRC
CRYPTOGRAPHY AND NETWORK SECURITY

INTRODUCTION TO MODERN

1.4.1 Principle 1 – Formal Definitions

One of the key contributions of modern cryptography has been the recognition that formal definitions of security are *essential* for the proper design, study, evaluation, and usage of cryptographic primitives. Put bluntly:

If you don't understand what you want to achieve, how can you possibly know when (or if) you have achieved it?

Formal definitions provide such understanding by giving a clear description of what threats are in scope and what security guarantees are desired. As such, definitions can help guide the design of cryptographic schemes. Indeed, it is much better to formalize what is required *before* the design process begins, rather than to come up with a definition *post facto* once the design is complete. The latter approach risks having the design phase end when the designers' patience is exhausted (rather than when the goal has been met), or may result in a construction achieving *more* than is needed at the expense of efficiency.



Contents

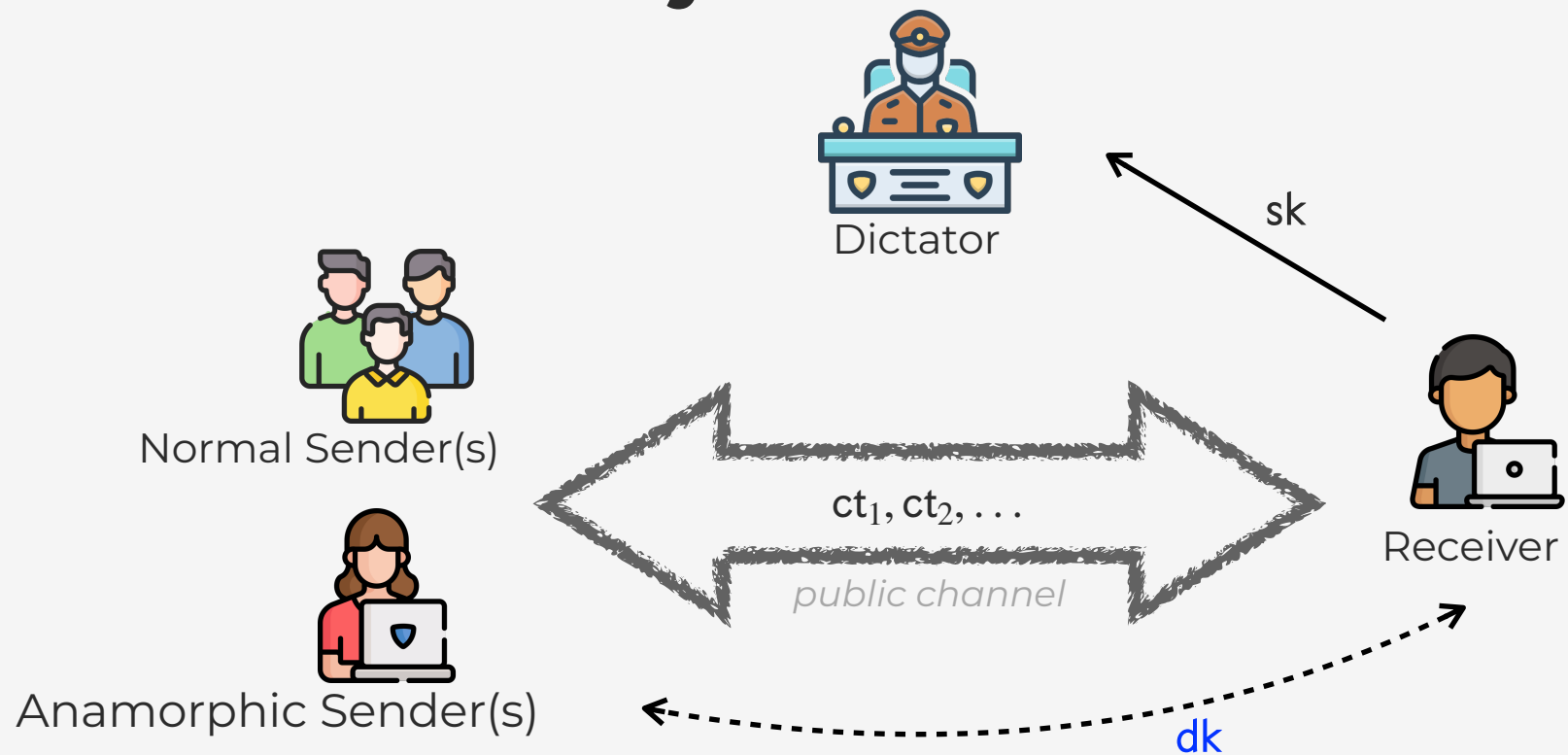
Preface	xv
I Introduction and Classical Cryptography	
1 Introduction	3
1.1 Cryptography and Modern Cryptography	3
1.2 The Setting of Private-Key Encryption	4
1.3 Historical Ciphers and Their Cryptanalysis	8
1.4 Principles of Modern Cryptography	16
1.4.1 Principle 1 – Formal Definitions	17
1.4.2 Principle 2 – Precise Assumptions	20
1.4.3 Principle 3 – Proofs of Security	22
1.4.4 Provable Security and Real-World Security	22
References and Additional Reading	23
Exercises	24



Our Formalization: Syntax

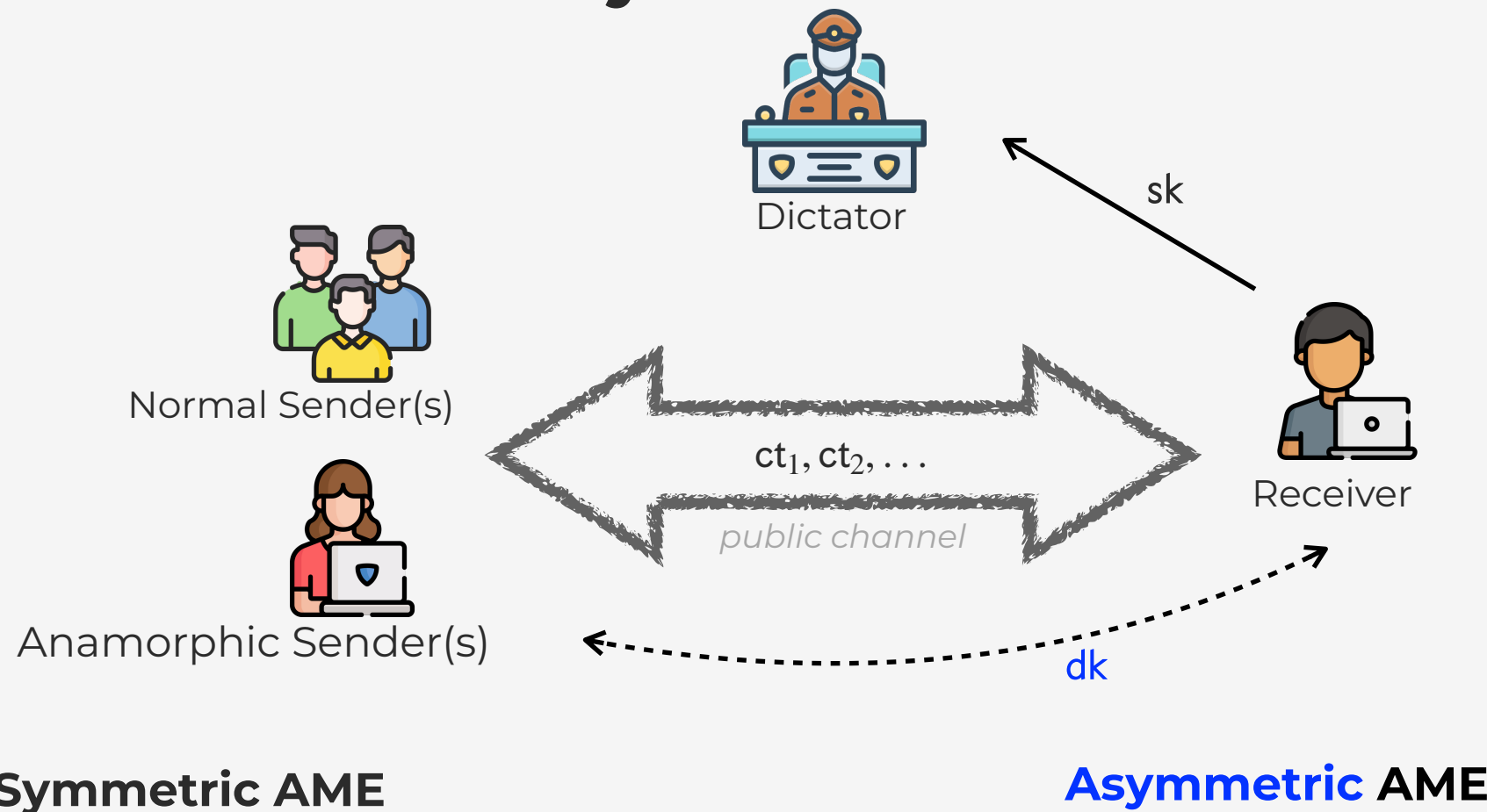


Our Formalization: Syntax



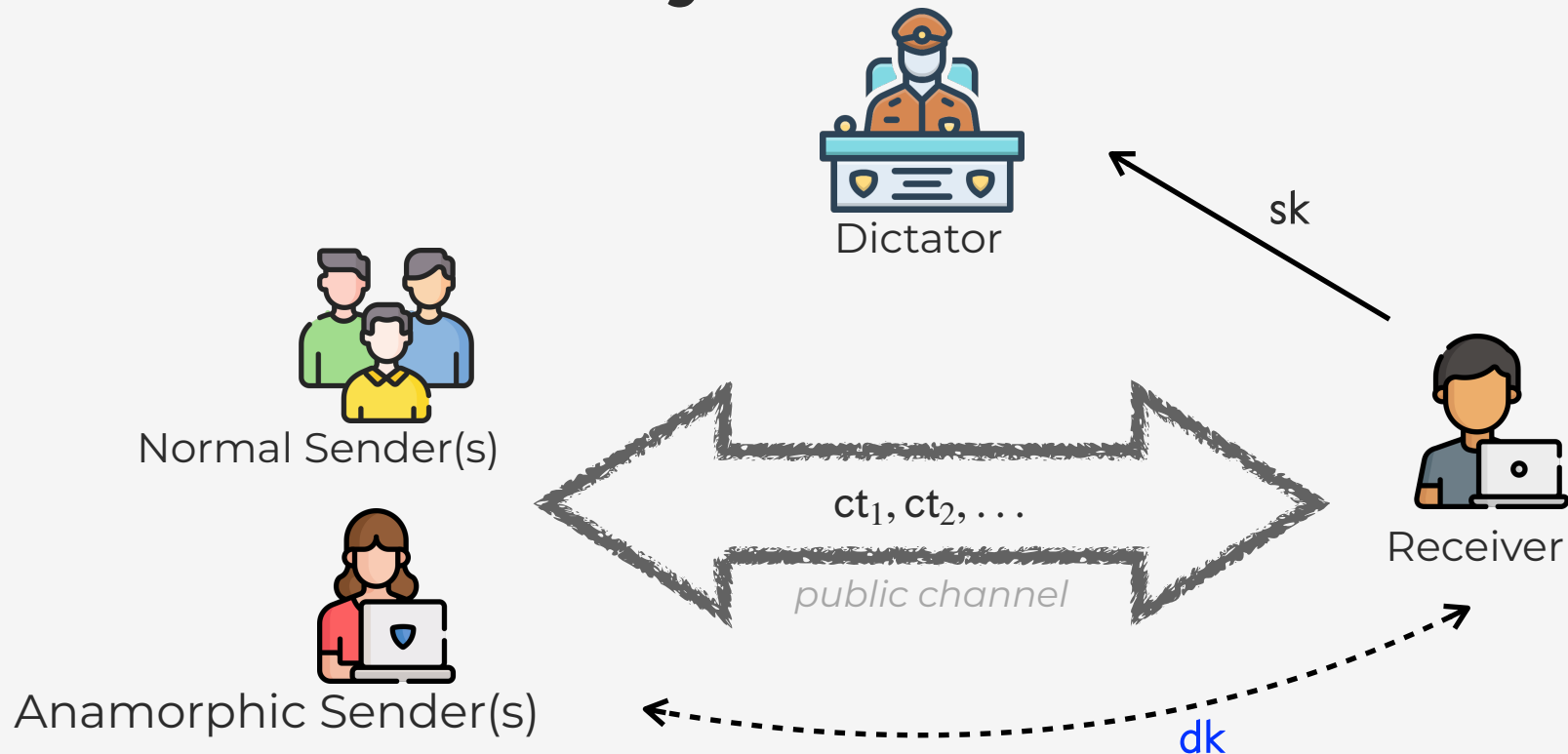


Our Formalization: Syntax





Our Formalization: Syntax



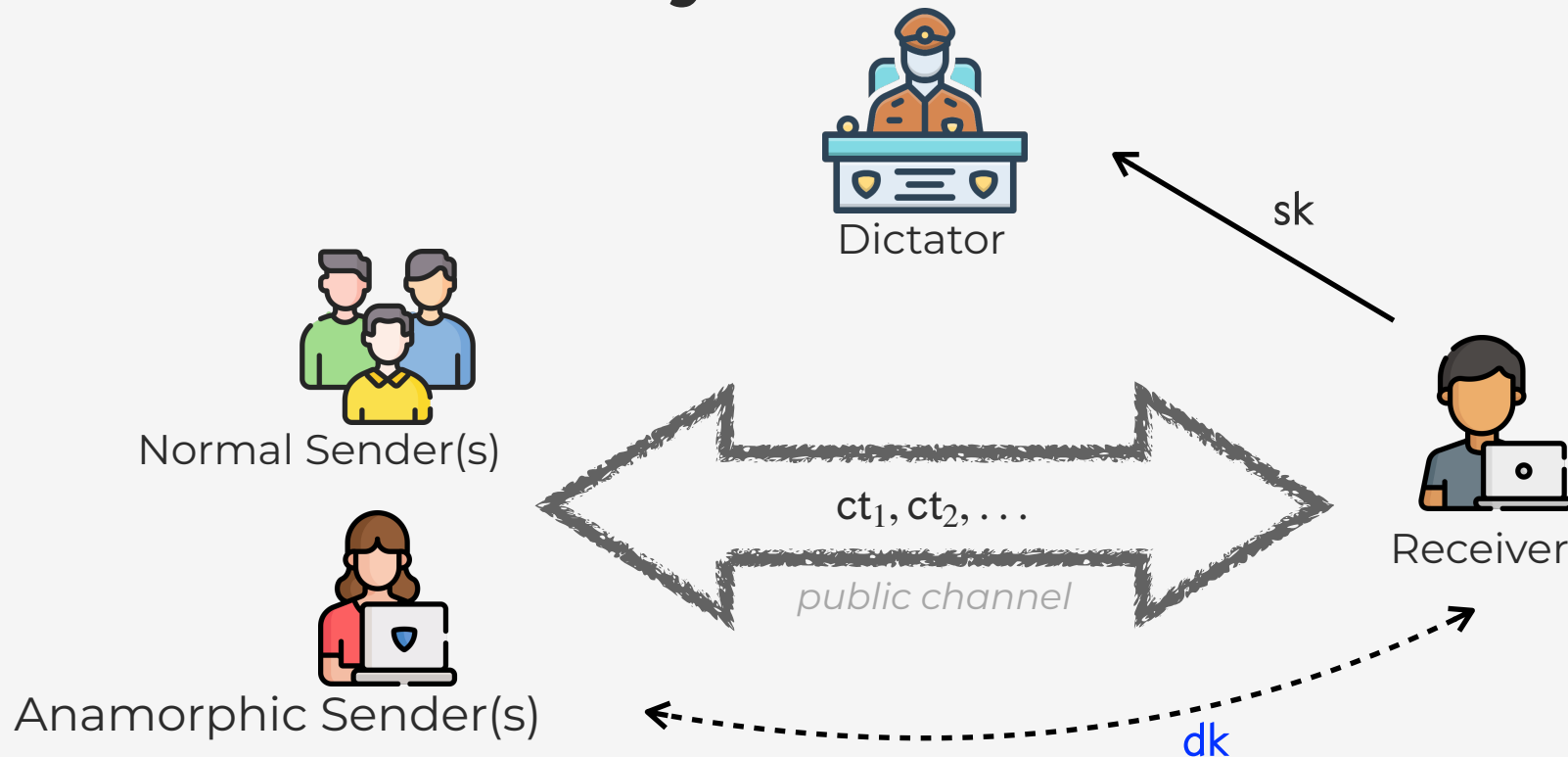
Symmetric AME

$$\begin{cases} (pk, sk, dk) \leftarrow \text{aKGen} \\ ct \leftarrow \text{aEnc}(pk, dk, m, \hat{m}) \\ \hat{m} \leftarrow \text{aDec}(sk, dk, ct) \end{cases}$$

Asymmetric AME



Our Formalization: Syntax



Symmetric AME

$$\begin{cases} (pk, sk, dk) \leftarrow \text{aKGen} \\ ct \leftarrow \text{aEnc}(pk, dk, m, \hat{m}) \\ \hat{m} \leftarrow \text{aDec}(sk, dk, ct) \end{cases}$$

Asymmetric AME

$$\begin{cases} (pk, sk, \textcolor{blue}{sdk}, \textcolor{blue}{rdk}) \leftarrow \text{aKGen} \\ ct \leftarrow \text{aEnc}(pk, \textcolor{blue}{sdk}, m, \hat{m}) \\ \hat{m} \leftarrow \text{aDec}(sk, \textcolor{blue}{rdk}, ct) \end{cases}$$



Our Formalization: Security



Our Formalization: Security

- What is the key setting, KGen or aKGen?



Our Formalization: Security

- What is the key setting, KGen or aKGen?
- What key(s) does the adversary get?



Our Formalization: Security

- What is the key setting, KGen or aKGen?
- What key(s) does the adversary get?
- What oracles does the adversary obtain? What are the inputs and outputs?



Our Formalization: Security

- What is the key setting, KGen or aKGen?
- What key(s) does the adversary get?
- What oracles does the adversary obtain? What are the inputs and outputs?
- What is the adversary's attack goal?



Our Formalization: Security

$\text{SEC}_{\Pi, \Sigma, b}(\mathcal{A})$:	$\text{CHAL}(\text{④-Input})$	$\text{AUX}_1(\dots)$
1 : if $b = 0$:	1 : if $b = 0$:	1 : ...
2 : $(\text{pk}, \text{sk}, \dots) \leftarrow \text{①-lKGen}$	2 : $\text{ct} \leftarrow \text{⑤-lChal}$	$\text{AUX}_2(\dots)$
3 : else :	3 : else :	1 : ...
4 : $(\text{pk}, \text{sk}, \dots) \leftarrow \text{②-rKGen}$	4 : $\text{ct} \leftarrow \text{⑥-rChal}$	
5 : return $\mathcal{A}^{\text{CHAL}, \text{AUX}_1, \text{AUX}_2, \dots}(\text{pk}, \text{③-Key})$	5 : return ct	



Our Formalization: Security

$\text{SEC}_{\Pi, \Sigma, b}(\mathcal{A})$:	$\text{CHAL}(\textcircled{4}\text{-Input})$	$\text{AUX}_1(\dots)$
1 : if $b = 0$:	1 : if $b = 0$:	1 : ...
2 : $(\text{pk}, \text{sk}, \dots) \leftarrow \textcircled{1}\text{-lKGen}$	2 : $\text{ct} \leftarrow \textcircled{5}\text{-lChal}$	$\text{AUX}_2(\dots)$
3 : else :	3 : else :	1 : ...
4 : $(\text{pk}, \text{sk}, \dots) \leftarrow \textcircled{2}\text{-rKGen}$	4 : $\text{ct} \leftarrow \textcircled{6}\text{-rChal}$	
5 : return $\mathcal{A}^{\text{CHAL}, \text{AUX}_1, \text{AUX}_2, \dots}(\text{pk}, \textcircled{3}\text{-Key})$	5 : return ct	

- What is the key setting, KGen or aKGen?
- What key(s) does the adversary get?



Our Formalization: Security

	①-lKGen	②-rKGen	③-Key	
			sym	asym
Dictator Stealthiness (STE)	KGen	aKGen	sk	sk
n/a	KGen	KGen	sk	sk
Dictator Indistinguishability (DIND)	aKGen	aKGen	sk	sk
Sender Indistinguishability (SIND)	aKGen	aKGen	dk	sdk
Colluder Indistinguishability (CIND)	aKGen	aKGen	(sk, dk)	(sk, sdk)

- What is the key setting, KGen or aKGen?
- What key(s) does the adversary get?



Our Formalization: Security

$\text{SEC}_{\Pi, \Sigma, b}(\mathcal{A})$:	$\text{CHAL}(\textcircled{4}\text{-Input})$	$\text{AUX}_1(\dots)$
1 : if $b = 0$:	1 : if $b = 0$:	1 : ...
2 : $(\text{pk}, \text{sk}, \dots) \leftarrow \textcircled{1}\text{-lKGen}$	2 : $\text{ct} \leftarrow \textcircled{5}\text{-lChal}$	$\text{AUX}_2(\dots)$
3 : else :	3 : else :	1 : ...
4 : $(\text{pk}, \text{sk}, \dots) \leftarrow \textcircled{2}\text{-rKGen}$	4 : $\text{ct} \leftarrow \textcircled{6}\text{-rChal}$	
5 : return $\mathcal{A}^{\text{CHAL}, \text{AUX}_1, \text{AUX}_2, \dots}(\text{pk}, \textcircled{3}\text{-Key})$	5 : return ct	



Our Formalization: Security

$\text{SEC}_{\Pi, \Sigma, b}(\mathcal{A})$:	$\text{CHAL}(\textcircled{4}\text{-Input})$	$\text{AUX}_1(\dots)$
1 : if $b = 0$:	1 : if $b = 0$:	1 : ...
2 : $(\text{pk}, \text{sk}, \dots) \leftarrow \textcircled{1}\text{-lKGen}$	2 : $\text{ct} \leftarrow \textcircled{5}\text{-lChal}$	$\text{AUX}_2(\dots)$
3 : else :	3 : else :	1 : ...
4 : $(\text{pk}, \text{sk}, \dots) \leftarrow \textcircled{2}\text{-rKGen}$	4 : $\text{ct} \leftarrow \textcircled{6}\text{-rChal}$	
5 : return $\mathcal{A}^{\text{CHAL}, \text{AUX}_1, \text{AUX}_2, \dots}(\text{pk}, \textcircled{3}\text{-Key})$	5 : return ct	

- What oracles does the adversary obtain? What are the inputs and outputs?
- What is the adversary's attack goal?



Our Formalization: Security

	④-Input	⑤-lChal	⑥-rChal
xCPA1	$(m, \hat{m})$	$\text{Enc}(m)$	$\text{aEnc}(m, \hat{m})$
xCPA2	$(m_0, m_1, \hat{m})$	$\text{Enc}(m_0)$	$\text{aEnc}(m_1, \hat{m})$
nCPA	(m_0, m_1)	$\text{Enc}(m_0)$	$\text{Enc}(m_1)$
aCPA1	$(m_0, m_1, \hat{m})$	$\text{aEnc}(m_0, \hat{m})$	$\text{aEnc}(m_1, \hat{m})$
aCPA2	$(m, \hat{m}_0, \hat{m}_1)$	$\text{aEnc}(m, \hat{m}_0)$	$\text{aEnc}(m, \hat{m}_1)$
aCPA3	$(m_0, m_1, \hat{m}_0, \hat{m}_1)$	$\text{aEnc}(m_0, \hat{m}_0)$	$\text{aEnc}(m_1, \hat{m}_1)$

- What oracles does the adversary obtain? What are the inputs and outputs?
- What is the adversary's attack goal?



Our Formalization: Security

STE (sk)	DIND (sk)	SIND (dk or sdk)	CIND (sk, sdk)
xCPA1			
xCPA2			
nCPA			
aCPA1			
aCPA2			
aCPA3			



Our Formalization: Security

	STE (sk)	DIND (sk)	SIND (dk or sdk)	CIND (sk, sdk)
xCPA1	STE-xCPA1 [PPY22]	DIND-xCPA1 [PPY22]	SIND-xCPA1	CIND-xCPA1
xCPA2	Impossible	Impossible	SIND-xCPA2	Impossible
nCPA	Impossible	Impossible	SIND-nCPA	Impossible
aCPA1	Not Well-Defined	Impossible	SIND-aCPA1	Impossible
aCPA2	Not Well-Defined	DIND-aCPA2 [PPY22]	SIND-aCPA2	CIND-aCPA2 [BPRS25]
aCPA3	Not Well-Defined	Impossible	SIND-aCPA3 [CGM24]	Impossible



Our Formalization: Security

	STE (sk)	DIND (sk)	SIND (dk or sdk)	CIND (sk, sdk)
xCPA1	STE-xCPA1 [PPY22]	DIND-xCPA1 [PPY22]	SIND-xCPA1	CIND-xCPA1
xCPA2	Impossible	Impossible	SIND-xCPA2	Impossible
nCPA	Impossible	Impossible	SIND-nCPA	Impossible
aCPA1	Not Well-Defined	Impossible	SIND-aCPA1	Impossible
aCPA2	Not Well-Defined	DIND-aCPA2 [PPY22]	SIND-aCPA2	CIND-aCPA2 [BPRS26]
aCPA3	Not Well-Defined	Impossible	SIND-aCPA3 [CGM24]	Impossible

- **Not Well Defined:** the corresponding key in oracles do not exist.



Our Formalization: Security

	STE (sk)	DIND (sk)	SIND (dk or sdk)	CIND (sk, sdk)
xCPA1	STE-xCPA1 [PPY22]	DIND-xCPA1 [PPY22]	SIND-xCPA1	CIND-xCPA1
xCPA2	Impossible	Impossible	SIND-xCPA2	Impossible
nCPA	Impossible	Impossible	SIND-nCPA	Impossible
aCPA1	Not Well-Defined	Impossible	SIND-aCPA1	Impossible
aCPA2	Not Well-Defined	DIND-aCPA2 [PPY22]	SIND-aCPA2	CIND-aCPA2 [BPRS26]
aCPA3	Not Well-Defined	Impossible	SIND-aCPA3 [CGM24]	Impossible

- **Impossible:** conflict with (anamorphic) decryption correctness.



Our Formalization: Security

	STE (sk)	DIND (sk)	SIND (dk or sdk)	CIND (sk, sdk)
xCPA1	STE-xCPA1 [PPY22]	DIND-xCPA1 [PPY22]	SIND-xCPA1	CIND-xCPA1
xCPA2	Impossible	Impossible	SIND-xCPA2	Impossible
nCPA	Impossible	Impossible	SIND-nCPA	Impossible
aCPA1	Not Well-Defined	Impossible	SIND-aCPA1	Impossible
aCPA2	Not Well-Defined	DIND-aCPA2 [PPY22]	SIND-aCPA2	CIND-aCPA2 [BPRS26]
aCPA3	Not Well-Defined	Impossible	SIND-aCPA3 [CGM24]	Impossible

- Existing Definitions



Our Formalization: Security

	STE (sk)	DIND (sk)	SIND (dk or sdk)	CIND (sk, sdk)
xCPA1	STE-xCPA1 [PPY22]	DIND-xCPA1 [PPY22]	SIND-xCPA1	CIND-xCPA1
xCPA2	Impossible	Impossible	SIND-xCPA2	Impossible
nCPA	Impossible	Impossible	SIND-nCPA	Impossible
aCPA1	Not Well-Defined	Impossible	SIND-aCPA1	Impossible
aCPA2	Not Well-Defined	DIND-aCPA2 [PPY22]	SIND-aCPA2	CIND-aCPA2 [BPRS26]
aCPA3	Not Well-Defined	Impossible	SIND-aCPA3 [CGM24]	Impossible

- **New Definitions**



Our Formalization: Security

	STE (sk)	DIND (sk)	SIND (dk or sdk)	CIND (sk, sdk)
xCPA1	STE-xCPA1 [PPY22]	DIND-xCPA1 [PPY22]	SIND-xCPA1	CIND-xCPA1
xCPA2	Impossible	Impossible	SIND-xCPA2	Impossible
nCPA	Impossible	Impossible	SIND-nCPA	Impossible
aCPA1	Not Well-Defined	Impossible	SIND-aCPA1	Impossible
aCPA2	Not Well-Defined	DIND-aCPA2 [PPY22]	SIND-aCPA2	CIND-aCPA2 [BPRS26]
aCPA3	Not Well-Defined	Impossible	SIND-aCPA3 [CGM24]	Impossible

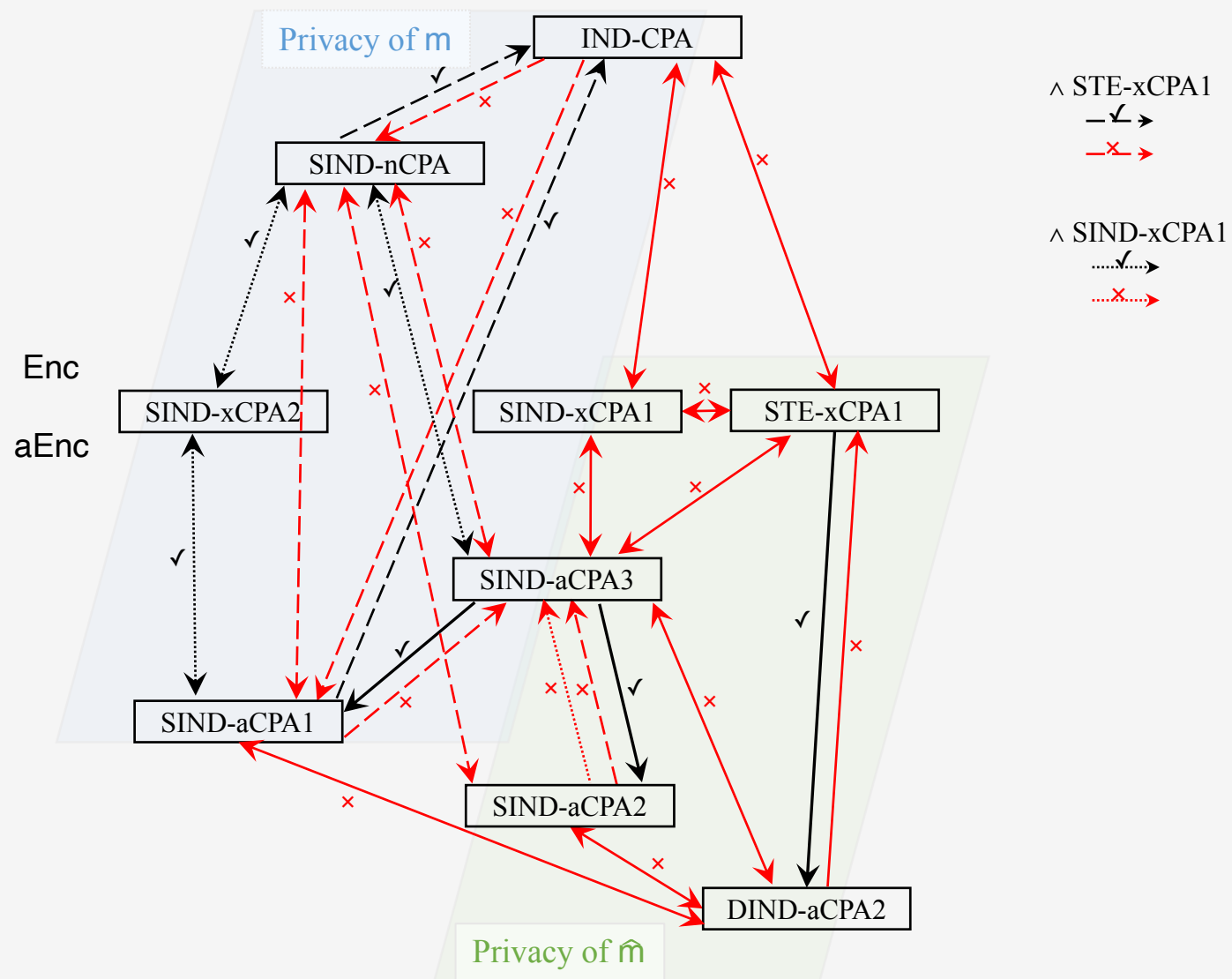
• New Definitions

E.g., SIND-xCPA1:

Given dk (or sdk), $\text{Enc}(m) \approx \text{aEnc}(m, \hat{m})$

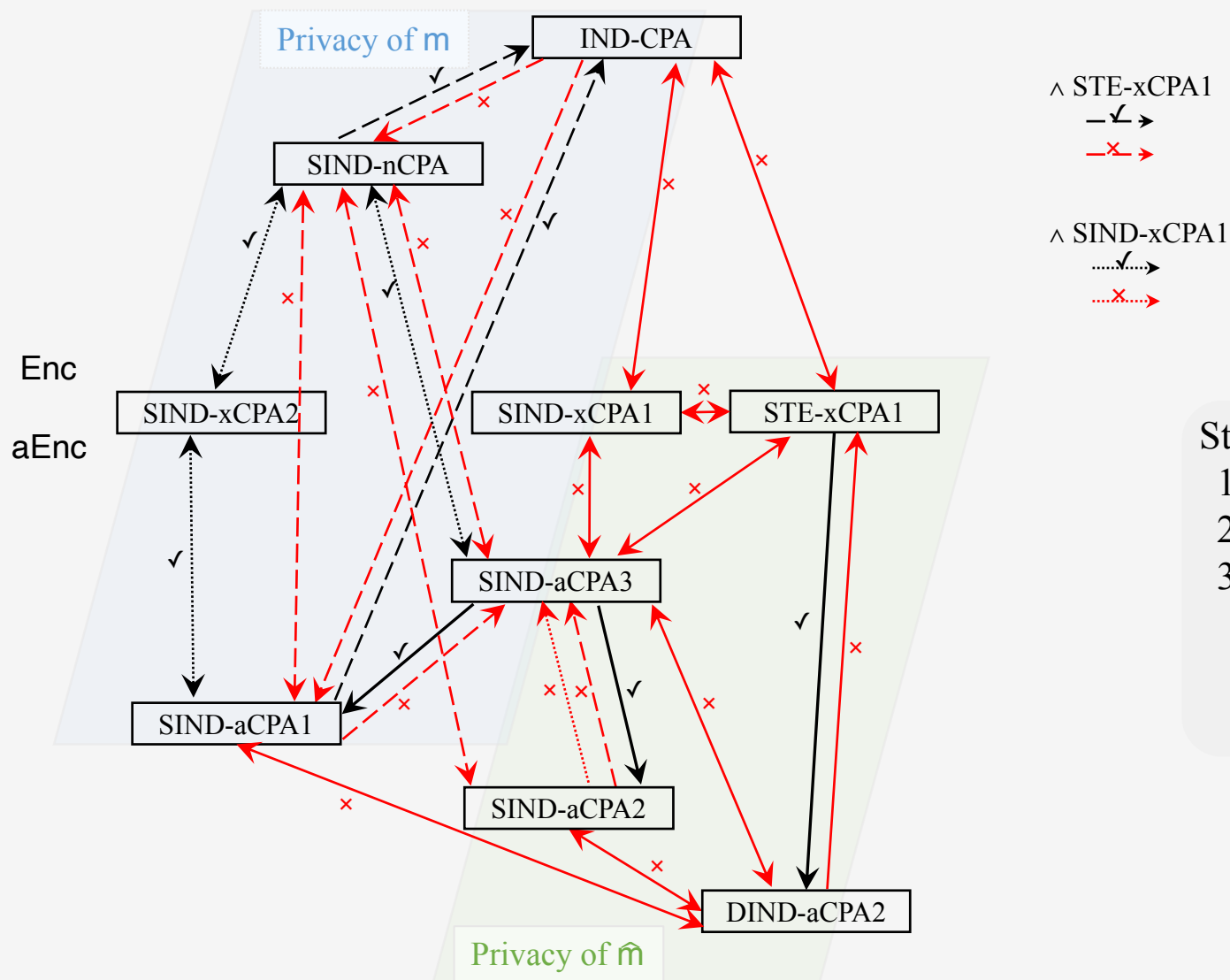


Relations (Symmetric Setting)





Relations (Symmetric Setting)



Strongest Notions in CPA

1. STE-xCPA1
2. SIND-xCPA1
3. SIND-nCPA
or SIND-xCPA2
or SIND-aCPA1
or SIND-aCPA3



Security in the CCA Setting



Security in the CCA Setting

- Decryption oracle $\text{DEC}(\text{sk}, \cdot)$
- Anamorphic decryption oracle $\text{ADEC}(\text{sk}, \text{dk}, \cdot)$ or $\text{ADEC}(\text{sk}, \text{rdk}, \cdot)$



Security in the CCA Setting

- Decryption oracle $\text{DEC}(\text{sk}, \cdot)$
- Anamorphic decryption oracle $\text{ADEC}(\text{sk}, \text{dk}, \cdot)$ or $\text{ADEC}(\text{sk}, \text{rdk}, \cdot)$

	STE (sk)	DIND (sk)	SIND (dk or sdk)	CIND (sk, sdk)
xCCA1	STE-xCCA1	DIND-xCCA1	SIND-xCCA1	CIND-xCCA1
xCCA2	Impossible	Impossible	SIND-xCCA2	Impossible
nCCA	Impossible	Impossible	SIND-nCCA	Impossible
aCCA1	Not Well-Defined	Impossible	SIND-aCCA1	Impossible
aCCA2	Not Well-Defined	DIND-aCCA2	SIND-aCCA2	CIND-aCCA2
aCCA3	Not Well-Defined	Impossible	SIND-aCCA3	Impossible



Security in the CCA Setting

- Decryption oracle $\text{DEC}(\text{sk}, \cdot)$
- Anamorphic decryption oracle $\text{ADEC}(\text{sk}, \text{dk}, \cdot)$ or $\text{ADEC}(\text{sk}, \text{rdk}, \cdot)$

	STE (sk)	DIND (sk)	SIND (dk or sdk)	CIND (sk, sdk)
xCCA1	STE-xCCA1	DIND-xCCA1	SIND-xCCA1	CIND-xCCA1
xCCA2	Impossible	Impossible	SIND-xCCA2	Impossible
nCCA	Impossible	Impossible	SIND-nCCA	Impossible
aCCA1	Not Well-Defined	Impossible	SIND-aCCA1	Impossible
aCCA2	Not Well-Defined	DIND-aCCA2	SIND-aCCA2	CIND-aCCA2
aCCA3	Not Well-Defined	Impossible	SIND-aCCA3	Impossible

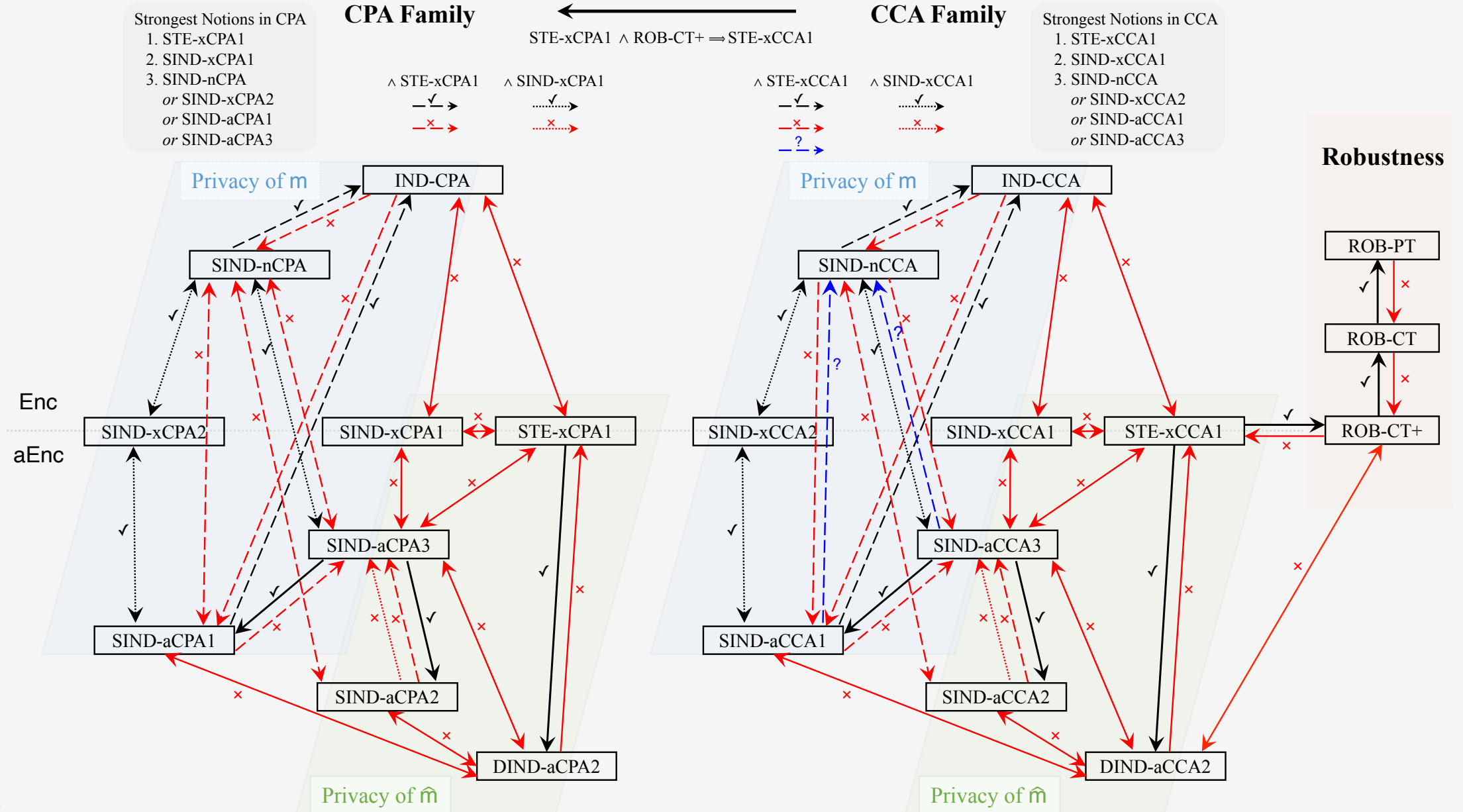
- CCA security vs. robustness
 - ROB-PT [BGH+24]
 - (new) ROB-CT, ROB-CT+



Relations (Symmetric Setting)



Relations (Symmetric Setting)





Analyzing AME Schemes



Analyzing AME Schemes

Approaches \ Security	STE-xCPA1	SIND-xCPA1	SIND-nCPA	ROB-PT	ROB-CT+	STE-xCCA1	SIND-xCCA1	SIND-nCCA	CIND-xCCA1
Symmetric AME									
Rejection Sampling [PPY22,CGM25]	✓	×	✓	×	×	×	×	×	n/a
rr-PKE [KPP ⁺ 23b,JS24]	✓	✓	✓	✓	✓ [*]	×	△	△	n/a
ElGamal [KPP ⁺ 23b]	✓	×	×	△	×	×	×	×	n/a
SPH [KPP ⁺ 23b]	✓	×	×	✓	×	×	×	×	n/a
(CCA2-secure) CS PKE [KPP ⁺ 23b]	✓	×	✓	✓	×	×	×	×	n/a
(CCA2-secure) CS PKE [BGH ⁺ 24]	✓	×	✓	✓	×	×	×	✓	n/a
PRF-based Scheme [BGH ⁺ 24,JS24]	✓	×	✓	✓	✓ [*]	×	×	✓	n/a
Hybrid Encryption [CGM24a]	✓	△	✓	✓	×	×	△	△	n/a
IBE-PKE [CGM24a]	✓	△	✓	✓	×	×	△	✓	n/a
Ours: rr-PKE + IC	✓	✓	✓	✓	✓	✓	✓	✓	n/a
Asymmetric AME									
NY Paradigm [PPY22]	✓	×	×	×	×	×	×	×	×
(CCA1-secure) CS PKE [CGM24a]	✓	✓	✓	✓	×	×	×	×	×
KW PKE [PPY24]	✓	✓	✓	✓	×	×	✓	✓	×
Dual Regev [DL26]	✓	✓	✓	✓	×	×	×	×	×
(Primal) Regev [DL26]	✓	✓	✓	✓	×	×	×	×	×
Dual GSW [DL26]	✓	✓	✓	✓	×	×	×	×	×
Ours: rr-PKE	✓	✓	✓	✓	✓	×	✓	✓	✓
Ours: rr-PKE + RO	✓	✓	✓	✓	✓	✓	✓	✓	✓



Analyzing AME Schemes

Approaches \ Security	STE-xCPA1	SIND-xCPA1	SIND-nCPA	ROB-PT	ROB-CT+	STE-xCCA1	SIND-xCCA1	SIND-nCCA	CIND-xCCA1
Symmetric AME									
Rejection Sampling [PPY22,CGM25]	✓	×	✓	×	×	×	×	×	n/a
rr-PKE [KPP ⁺ 23b,JS24]	✓	✓	✓	✓	✓ [*]	×	△	△	n/a
ElGamal [KPP ⁺ 23b]	✓	×	×	△	×	×	×	×	n/a
SPH [KPP ⁺ 23b]	✓	×	×	✓	×	×	×	×	n/a
(CCA2-secure) CS PKE [KPP ⁺ 23b]	✓	×	✓	✓	×	×	×	×	n/a
(CCA2-secure) CS PKE [BGH ⁺ 24]	✓	×	✓	✓	×	×	×	✓	n/a
PRF-based Scheme [BGH ⁺ 24,JS24]	✓	×	✓	✓	✓ [*]	×	×	✓	n/a
Hybrid Encryption [CGM24a]	✓	△	✓	✓	×	×	△	△	n/a
IBE-PKE [CGM24a]	✓	△	✓	✓	×	×	△	✓	n/a
Ours: rr-PKE + IC	✓	✓	✓	✓	✓	✓	✓	✓	n/a
Asymmetric AME									
NY Paradigm [PPY22]	✓	×	×	×	×	×	×	×	×
(CCA1-secure) CS PKE [CGM24a]	✓	✓	✓	✓	×	×	×	×	×
KW PKE [PPY24]	✓	✓	✓	✓	×	×	✓	✓	×
Dual Regev [DL26]	✓	✓	✓	✓	×	×	×	×	×
(Primal) Regev [DL26]	✓	✓	✓	✓	×	×	×	×	×
Dual GSW [DL26]	✓	✓	✓	✓	×	×	×	×	×
Ours: rr-PKE	✓	✓	✓	✓	✓	×	✓	✓	✓
Ours: rr-PKE + RO	✓	✓	✓	✓	✓	✓	✓	✓	✓



rr-PKE

- Randomness Recovering PKE (rrPKE):
 - $\text{rrDec}(\text{sk}, \text{ct})$ returns r such that $\text{ct} := \text{Enc}(\text{pk}, m; r)$
 - E.g., Fujisaki-Okamoto transform



rr-PKE

- Randomness Recovering PKE (rrPKE):
 - $\text{rrDec}(\text{sk}, \text{ct})$ returns r such that $\text{ct} := \text{Enc}(\text{pk}, \text{m}; r)$
 - E.g., Fujisaki-Okamoto transform
- [KPP+23]: rrPKE has anamorphic extension



rr-PKE

- Randomness Recovering PKE (rrPKE):
 - $\text{rrDec}(\text{sk}, \text{ct})$ returns r such that $\text{ct} := \text{Enc}(\text{pk}, m; r)$
 - E.g., Fujisaki-Okamoto transform
- [KPP+23]: rrPKE has anamorphic extension
- **Ours**: rrPKE also provides **the strongest** CCA security
 - Symmetric setting: STE-xCCA1, SIND-xCCA1, SIND-nCCA
 - **Asymmetric** setting: STE-xCCA1, SIND-xCCA1, SIND-nCCA, **CIND-xCCA1**

Approaches \ Security	STE-xCPA1	SIND-xCPA1	SIND-nCPA	ROB-PT	ROB-CT+	STE-xCCA1	SIND-xCCA1	SIND-nCCA	CIND-xCCA1
Symmetric AME									
Ours: rr-PKE + IC	✓	✓	✓	✓	✓	✓	✓	✓	n/a
Asymmetric AME									
Ours: rr-PKE + RO	✓	✓	✓	✓	✓	✓	✓	✓	✓



rr-PKE

- Randomness Recovering PKE (rrPKE):
 - $\text{rrDec}(\text{sk}, \text{ct})$ returns r such that $\text{ct} := \text{Enc}(\text{pk}, m; r)$
 - E.g., Fujisaki-Okamoto transform
- [KPP+23]: rrPKE has anamorphic extension
- **Ours**: rrPKE also provides **the strongest** CCA security
 - Symmetric setting: STE-xCCA1, SIND-xCCA1, SIND-nCCA
 - **Asymmetric** setting: STE-xCCA1, SIND-xCCA1, SIND-nCCA, **CIND-xCCA1**

Approaches \ Security	STE-xCPA1	SIND-xCPA1	SIND-nCPA	ROB-PT	ROB-CT+	STE-xCCA1	SIND-xCCA1	SIND-nCCA	CIND-xCCA1
Symmetric AME									
Ours: rr-PKE + IC	✓	✓	✓	✓	✓	✓	✓	✓	n/a
Asymmetric AME									
Ours: rr-PKE + RO	✓	✓	✓	✓	✓	✓	✓	✓	✓

- Proofs are not straightforward, rely on ideal functionalities (RO or IC)



Conclusion



Conclusion

- A single template generates every AME definition



Conclusion

- A single template generates every AME definition
- CCA security and collusion



Conclusion

- A single template generates every AME definition
- CCA security and collusion
- Three (four in asym.) notions imply all!



Conclusion

- A single template generates every AME definition
- CCA security and collusion
- Three (four in asym.) notions imply all!
- rrPKE: achieve the strongest security



Conclusion

- A single template generates every AME definition
- CCA security and collusion
- Three (four in asym.) notions imply all!
- rrPKE: achieve the strongest security
- There is no perfect definition for AME, but there are "correct" ones.



Conclusion

- A single template generates every AME definition
- CCA security and collusion
- Three (four in asym.) notions imply all!
- rrPKE: achieve the strongest security
- There is no perfect definition for AME, but there are "correct" ones.

I am on 2026-2027 job market!

<https://xiangyuliu-yahoo.github.io/>





Conclusion

- A single template generates every AME definition
 - CCA security and collusion
 - Three (four in asym.) notions imply all!
 - rrPKE: achieve the strongest security
-
- There is no perfect definition for AME, but there are "correct" ones.

I am on 2026-2027 job market!

<https://xiangyuliu-yahoo.github.io/>

Thank you! Q&A

