A Unified Treatment of Anamorphic Encryption*

Wonseok Choi[1**], Daniel Collins[2,3***], Xiangyu Liu[4†],
Roy Stracovsky[5], and Vassilis Zikas[5]

[1] DGIST
[2] New York University
[3] The Hebrew University
[4] CISPA Helmholtz Center for Information Security
[5] Georgia Institute of Technology

wonseok@dgist.ac.kr
d.collins@nyu.edu
xiangyu.liu@cispa.de
{rstracovsky3, vzikas}@gatech.edu

Abstract. (Receiver) anamorphic encryption, introduced by Persiano et al. [Eurocrypt'22], allows for hiding a covert message $\hat{\mathsf{m}}$ (dubbed the *anamorphic message*) within a public-key ciphertext encrypting another message m so that: (1) a "dictator" adversary with access to the associated secret key cannot detect that such an $\hat{\mathsf{m}}$ has been hidden, let alone learn information about it, and (2) knowledge of an additional key (called the *double key*), which is kept hidden from the dictator, allows to fully recover $\hat{\mathsf{m}}$. Anamorphic encryption schemes are typically constructed by mixing symmetric-key and public-key encryption in a clever way, and existing instantiations achieve a variety of novel security guarantees. This has resulted in a disparate landscape of security notions for anamorphic encryption with no systematization and fragmented knowledge about the relationships between them.

We put forth a systematic study of anamorphic encryption and summarize our findings as follows. We begin by developing a template for defining indistinguishability-based security for anamorphic encryption. From this, we systematically obtain several new but meaningful definitions for anamorphic encryption. We contextualize all existing and new security notions by proving implications and separations between them. Our methodology also covers *asymmetric* anamorphic encryption—where the anamorphic message $\hat{\mathsf{m}}$ is hidden using public-key techniques—and uncovers a security setting not previously considered: when the dictator and the anamorphic sender collude. Finally, we apply our treatment to capture chosen-ciphertext attacks, which were recently introduced to anamorphic cryptography by Jaeger and Stracovsky [Asiacrypt'24], and propose the first construction of anamorphic encryption that achieves all desirable security properties in this setting.

* This is the full version of the paper with the same title appearing in the proceedings of CRYPTO 2026.
** Work partially completed while Wonseok Choi was at Purdue University and the Georgia Institute of Technology.
*** Work partially completed while Daniel Collins was at the Georgia Institute of Technology, Purdue University, and Texas A&M University.
† Work partially completed while Xiangyu Liu was at Purdue University and the Georgia Institute of Technology.

Table of Contents

A Unified Treatment of Anamorphic Encryption . . . 1
Wonseok Choi, Daniel Collins, Xiangyu Liu, Roy Stracovsky, and Vassilis Zikas
1 Introduction . . . 3
1.1 Contributions and Overview . . . 3
2 Preliminaries . . . 7
2.1 Public-Key Encryption . . . 7
2.2 Symmetric Encryption . . . 9
2.3 Authenticated Encryption . . . 10
2.4 Pseudorandom Functions . . . 10
2.5 Anamorphic Encryption . . . 11
3 Security of Anamorphic Encryption in the CPA Setting . . . 12
3.1 A Comprehensive Approach to CPA Definitions . . . 12
3.2 Security Notions against the Dictator . . . 14
3.3 Security Notions against Anamorphic Senders . . . 17
3.4 Relations among Notions in the CPA Setting . . . 21
4 Security of Anamorphic Encryption in the CCA Setting . . . 26
4.1 Robustness . . . 26
4.2 Security against the Dictator in the CCA Setting . . . 29
4.3 Security against Anamorphic Senders in the CCA Setting . . . 32
4.4 Relations among Notions in the CCA Setting . . . 36
5 Randomness-Recovering PKE: Achieving All Security Notions . . . 40
6 Security of Asymmetric AME against Collusion . . . 43
6.1 Security Notions . . . 43
6.2 Relations with Others . . . 44
7 AME Schemes Satisfying All Asymmetric Security Notions . . . 46
7.1 Achieving CIND-xCCA1 Security (Only) . . . 46
7.2 Achieving both CIND-xCCA1 Security and STE-xCCA1 Security . . . 47
A Additional Results in the CPA Setting . . . 52
A.1 Not Well-Defined Definitions . . . 52
A.2 Impossible Definitions . . . 52
B Related Work and Discussions . . . 53
B.1 Related Work . . . 53
B.2 Further Discussion . . . 54
C Motivations for Additional Security Considerations . . . 55

1 Introduction

Encryption has been under threat by governments around the globe for decades, from export restrictions tantamount to rendering the cryptography useless and the Clipper chip of the 1990s [Fro94], to proposals made as recently as 2025 to ban or restrict the usage of end-to-end encryption [Woo25].

Supposing that encryption is indeed outlawed or heavily monitored in a given jurisdiction, a natural question is to study to what extent secure communication can still take place. To deal with this problem, (receiver) anamorphic encryption (AME) was proposed by Persiano et al. [PPY22] as a way for parties to secretly communicate even in the presence of an adversarial "dictator" who can compromise the secret decryption keys of users.[6]

An AME scheme consists of a standard public-key encryption (PKE) scheme $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ and an associated anamorphic triplet $\Sigma = (\mathsf{aKGen}, \mathsf{aEnc}, \mathsf{aDec})$ and can be deployed in two modes: normal or anamorphic. In its normal mode, the scheme is used as a standard PKE scheme. If the anamorphic mode is deployed, then the anamorphic key generation outputs a double key dk in addition to the usual public-secret key-pair $(\mathsf{pk}, \mathsf{sk})$. Given parties can exchange dk out-of-band, dk can then be used to carry an anamorphic message (or double message) $\hat{\mathsf{m}}$ alongside the normal message m via anamorphic encryption $\mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}})$. An anamorphic ciphertext can then be both normally decrypted to m using sk, or be decrypted to $\hat{\mathsf{m}}$ using dk.

Since its introduction [PPY22], there has been a flurry of research on anamorphic encryption. Some works examine what kind of security can be provided against a dictator, including *robustness* [BGH+24], ensuring that regular encryptions must fail to be anamorphically decrypted, and more generally chosen ciphertext attacks [JS24]. Some works consider security against an anamorphic *sender* with knowledge of the double key dk but not the receiver's PKE secret key sk. Related is *asymmetric* anamorphic encryption, where the sender holds a sender double key sdk and the receiver holds a *separate* receiver double key rdk [CGM24a]. Constructive aspects, which we do not explore in this paper, include homomorphic AME [CGM24a,DL26], anamorphic signatures [DL25a,JS24,KPP+23a], anamorphic-resistant encryption [ABG+26,CCGM25,DG25], and anamorphic key exchange [WHL24].

In part due to its quickly emerging prevalence, the current landscape is somewhat fragmented, with existing works investigating special cases of anamorphic encryption and its security. This makes it difficult to navigate and identify gaps (e.g., in security notions constructions should target). As such, we believe that a systematic study of existing security notions and their meaningful extensions is an important step forward for systematizing existing knowledge, extending it, and facilitating future research on anamorphic encryption.

In this work, we put forth a systematic treatment of the anamorphic encryption landscape. This allows us to provide a comprehensive collection of security notions (including new, natural notions our methodology uncovers) and implications between them. Systematic treatments have proven highly successful in previous studies of primitives including public-key encryption [BDPR98], authenticated encryption [BN08], and garbled circuits [BHR12].

1.1 Contributions and Overview

We use a clear-cut, repeatable, and easily evaluable methodology to enumerate all natural definitions (old and new) for anamorphic cryptography. We summarize relevant definitions in the CPA setting that we consider in Table 1, and capture the CCA setting, including robustness, in Table 2. Each definition targets a natural security goal in a particular setting. We further explore implications and separations between notions and summarize our findings in Fig. 1. In Table 3, we identify which of these notions are satisfied by existing constructions. Our systematic study reveals several (new and) meaningful security definitions, which cannot be instantiated by existing constructions or straightforward adaptations thereof. We propose constructions satisfying these notions, in both the symmetric and asymmetric settings, to complete the picture.

[6] Persiano et al. [PPY22] additionally propose the more esoteric notion of *sender anamorphic encryption* which is beyond the scope of this work.

Systematic Methodology for Security Notions. Numerous security definitions for anamorphic cryptography have been proposed [BPRS25,BGH+24,CGM24a,JS24,KPP+23b,PPY22], typically to address (or repair) particular settings as they arose. As such, they were introduced in an ad hoc manner rather than derived from a unified framework. In this work, we instead start with a comprehensive approach for defining security of AME. We begin with a template for defining generic indistinguishability-style security games for AME (Fig. 7) and consider the following aspects.

- How keys are generated (i.e., by normal key generation KGen or by anamorphic key generation aKGen), and which components of the keys are revealed to the adversary $\mathcal{A}$ (i.e., the secret key sk or the double key dk).
- The inputs to the left-or-right challenge oracle (i.e., normal message(s) m_b and/or anamorphic message(s) $\hat{\mathsf{m}}_b$ where $b \in \{0,1\}$).
- How the challenge ciphertexts are generated (i.e., via normal encryption Enc or anamorphic encryption aEnc).

Using the above systematic methodology, we enumerate all resulting security notions for anamorphic encryption, and identify many new and natural notions.

New Security Notions (or Gaps in Existing Definitions). Our systematic characterization uncovers three "fundamental" notions that, together, imply all other anamorphic security notions in the CPA setting as well as standard IND-CPA security for public-key encryption (see Table 1 and Fig. 1).

- STE-xCPA1 security: Given pk and sk, a dictator cannot distinguish whether the cryptosystem is deployed in normal or anamorphic mode. Note that this is the basic requirement of AME introduced in the original work [PPY22].
- SIND-xCPA1 security (Def. 14): Given pk and dk, a sender cannot distinguish normal and anamorphic ciphertexts.
- SIND-nCPA security (Def. 16): Given pk and dk, a sender cannot distinguish what (plain) message a normal ciphertext encrypts.

To our knowledge, these last two definitions above have not been considered in prior work. In addition to their utility as they imply all other notions, these definitions are meaningful in their own right. For instance, an SIND-xCPA1-secure scheme prevents senders from learning who else is concealing anamorphic messages or how many anamorphic messages a recipient receives. As this has not been defined prior, it is unclear which existing schemes achieve this property.

Relations Between Security Notions. Many of our newly identified anamorphic security notions, when considered in isolation, do not imply nor are implied by existing security notions including standard IND-CPA security, STE-xCPA1 security (the core security requirement of AME [PPY22]), nor SIND-aCPA1 security (introduced in [KPP+23b]). Indeed, this highlights that security against the dictator (notions labelled STE-X (Dictator-Stealthiness-X) or DIND-X (Dictator-Indistinguishability)) and security against the anamorphic sender (notions labelled SIND-X (Sender-Indistinguishability)) are incomparable, due to the different roles that the secret key sk and the double key dk play.

As summarized in Fig. 1, we show important implications and separations among the different definitions, many of which are non-trivial. For example, observe that SIND-nCPA and SIND-aCPA3 do not imply each other, since the first is concerned with indistinguishability of Enc and the second with aEnc. By careful construction of counterexamples, we show that this is still the case *even when* STE-xCPA1 (the fundamental security property of AME) also holds, i.e.,

$$\text{SIND-nCPA } (+ \text{ STE-xCPA1}) \not\Longleftrightarrow \text{SIND-aCPA3 } (+ \text{ STE-xCPA1}).$$

Interestingly, we show that if $(\mathsf{pk}, \mathsf{sk})$ and dk are independent, then

$$\begin{aligned} & \text{SIND-nCPA} && \Longleftarrow \text{SIND-aCPA3} + \text{STE-xCPA1}. \\ \textit{but}\quad & \text{SIND-nCPA} + \text{STE-xCPA1} && \not\Longrightarrow \text{SIND-aCPA3}. \end{aligned}$$

Security Notions	**Indistinguishability Goal**	$\mathcal{A}$**'s Additional Info.**
IND-CPA	$\mathsf{Enc}(\mathsf{m}_0) \approx \mathsf{Enc}(\mathsf{m}_1)$	
CPA security against the dictator		
STE-xCPA1 [PPY22]	Normal Mode $\approx$ Anam. Mode $\mathsf{Enc}(\mathsf{m}) \approx \mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}})$	sk // $\mathsf{KGen}(1^\lambda)$ in NM $\mathsf{aKGen}(1^\lambda)$ in AM
DIND-xCPA1 [PPY22]	$\mathsf{Enc}(\mathsf{m}) \approx \mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}})$	sk // $\mathsf{aKGen}(1^\lambda)$
DIND-aCPA2 [PPY22]	$\mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}}_0) \approx \mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}}_1)$	sk // $\mathsf{aKGen}(1^\lambda)$
CPA security against the anamorphic senders		
SIND-xCPA1 (Def. 14)	$\mathsf{Enc}(\mathsf{m}) \approx \mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}})$	(s)dk // $\mathsf{aKGen}(1^\lambda)$
SIND-xCPA2 (Def. 15)	$\mathsf{Enc}(\mathsf{m}_0) \approx \mathsf{aEnc}(\mathsf{m}_1, \hat{\mathsf{m}})$	(s)dk // $\mathsf{aKGen}(1^\lambda)$
SIND-nCPA (Def. 16)	$\mathsf{Enc}(\mathsf{m}_0) \approx \mathsf{Enc}(\mathsf{m}_1)$	(s)dk // $\mathsf{aKGen}(1^\lambda)$
SIND-aCPA1 [KPP+23b]	$\mathsf{aEnc}(\mathsf{m}_0, \hat{\mathsf{m}}) \approx \mathsf{aEnc}(\mathsf{m}_1, \hat{\mathsf{m}})$	(s)dk // $\mathsf{aKGen}(1^\lambda)$
SIND-aCPA2 (Def. 18)	$\mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}}_0) \approx \mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}}_1)$	(s)dk // $\mathsf{aKGen}(1^\lambda)$
SIND-aCPA3 (Def. 19)/[CGM24a]	$\mathsf{aEnc}(\mathsf{m}_0, \hat{\mathsf{m}}_0) \approx \mathsf{aEnc}(\mathsf{m}_1, \hat{\mathsf{m}}_1)$	(s)dk // $\mathsf{aKGen}(1^\lambda)$
CPA security against the collusion in the asymmetric setting		
CIND-xCPA1 (Def. 33)	$\mathsf{Enc}(\mathsf{m}) \approx \mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}})$	$\mathsf{sk}, \mathsf{sdk}$ // $\mathsf{aKGen}(1^\lambda)$
CIND-aCPA2 (Def. 34)/[BPRS25]	$\mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}}_0) \approx \mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}}_1)$	$\mathsf{sk}, \mathsf{sdk}$ // $\mathsf{aKGen}(1^\lambda)$

Table 1. Table of CPA security notions for $\mathsf{AME} = (\Pi, \Sigma)$ with $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ and $\Sigma = (\mathsf{aKGen}, \mathsf{aEnc}, \mathsf{aDec})$. We omit the input pk in Enc and pk, dk (or sdk) in aEnc in the second column.

This highlights the subtleties inherent to comparing security notions.

On the other hand, conditioned on SIND-xCPA1, we show that SIND-nCPA and SIND-aCPA3 are equivalent (Theorem 11). We refer the reader to Section 3 for full details on the relations between notions.

Completing the CCA Picture. The aforementioned CPA landscape drawn by previous works is even more incomplete when considering chosen-ciphertext attacks (CCA). To address this, we start by formalizing STE-xCCA1 security in the CCA setting, a strictly stronger notion than the widely-applied STE-xCPA1 notion [PPY22], but which better matches the CCA security setting that is by now standard (and even expected) for normal encryption.

– STE-xCCA1 security: Given pk and sk, a dictator cannot distinguish whether the cryptosystem is deployed in normal or anamorphic mode, even when given access to a CCA oracle which returns $\perp$ in the normal mode and returns anamorphic decryptions in the anamorphic mode.

We emphasize that no existing AME scheme we are aware of achieves the above STE-xCCA1 security (see Table 3), much less other potential CCA definitions for anamorphic encryption. To address this gap, we extend all the CPA security notions that we consider to the CCA setting, and systematically investigate the relations among them, including their connection with robustness [BGH+24]. To address the insufficiency of the existing notion of robustness, we consider two stronger variants, namely ROB-CT security and ROB-CT+ security, building upon the robustness notion defined in [BGH+24]. We additionally explore their relations with CPA and CCA security of AME (see Fig. 1) and show that a scheme that is STE-xCPA1 secure and ROB-CT+ secure is also STE-xCCA1 secure. This is analogous to the well-known result that privacy and ciphertext integrity implies IND-CCA security for authenticated encryption [BN08].

As in the CPA case, our systematization allows us to identify a small set of notions that together imply all other notions, including the standard IND-CCA definition. Namely,

1. STE-xCCA1 security (Def. 23),
2. SIND-xCCA1 security (Def. 26),
3. SIND-nCCA security (Def. 28), or other equivalent notions conditioned on the second property (see Theorem 26).

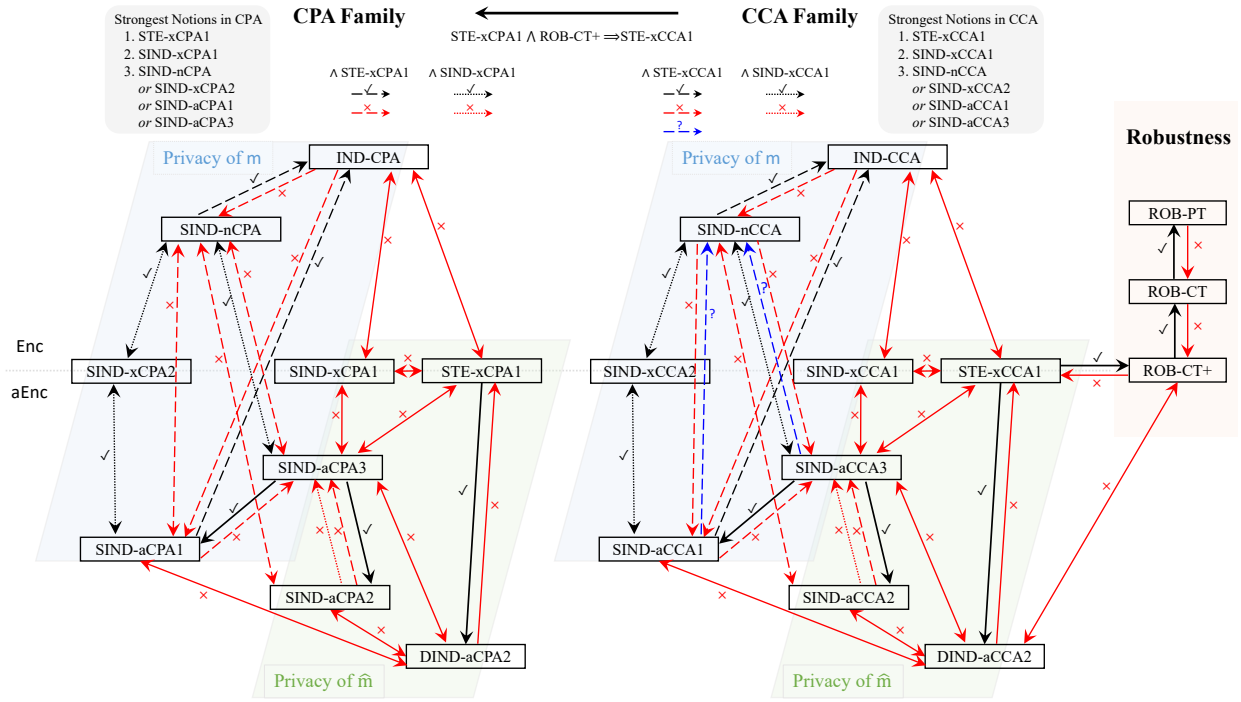


Fig. 1. Relations among different security notions for symmetric AME. "A $\xrightarrow{\checkmark}$ B" indicates that A implies B, while "A $\xrightarrow{\times}$ B" indicates that the implication is impossible. "A $\overset{\checkmark}{\dashrightarrow}$ B", "A $\overset{\times}{\dashrightarrow}$ B", and "A $\overset{?}{\dashrightarrow}$ B" indicate that, conditioned on C, A implies B, A does not imply B, or the relation is unknown, respectively. Here, C is STE-xCPA1 or SIND-xCPA1 in the CPA family and STE-xCCA1 or SIND-xCCA1 in CCA family.

To complete the last part of the CCA picture, we propose a generic paradigm for designing secure AME schemes, which for the first time simultaneously satisfies the above three strongest notions, STE-xCCA1, SIND-xCCA1 and SIND-nCCA, from a randomness-recovering PKE (rr-PKE) scheme. Though rr-PKE schemes have been shown to admit anamorphic triplets in [KPP+23b], the generic compiler of [KPP+23b] is not secure in the CCA setting. Without a unified landscape, previous work was unable to target all relevant notions simultaneously. We enhance the compiler to achieve all three (and, per our implications, all) security notions, leveraging the encode-then-encipher paradigm [BR00] in the ideal cipher model. Our constructions demonstrate the feasibility of achieving our security notions simultaneously, and we leave efficient and tailored approaches to satisfying these notions as important future work.

Collusion Issues in the Asymmetric Setting. In the asymmetric AME setting, the double key is split into the sender double key sdk (held by anamorphic senders) and the receiver double key rdk (known only to the receiver). Under our unified treatment, we identify two notions that are well-defined only in the asymmetric setting. We call these Colluder-Indistinguishability-xCPA1 (CIND-xCPA1) and CIND-aCPA2 in the CPA setting, and CIND-xCCA1 and CIND-aCCA2 in the CCA setting. At a high level, these notions capture security against a powerful adversary who knows both the secret key sk and the sender double key sdk. We further adapt the generic compiler above to the asymmetric setting to achieve CIND-xCCA1 security (and hence CIND-aCCA2 security).

Summary. We address the lack of a systematic security treatment for anamorphic encryption. By applying a clear, repeatable methodology to enumerate a comprehensive class of security definitions, we identify new, meaningful notions even within the CPA setting (which has already been considered frequently in prior work), demonstrating the utility of our approach. It also allows us to identify a minimal set of security

Security Notions	Indistinguishability Goal	$\mathcal{A}$'s Additional Info.	$\mathcal{A}$'s Oracles
Robustness			
ROB-PT [BGH+24]	$\mathsf{aDec}(\mathsf{sk}, (\mathsf{r})\mathsf{dk}, \mathsf{Enc}(\mathsf{m})) \approx \perp$	sk // $\mathsf{aKGen}(1^\lambda)$	$\perp$
ROB-CT (Def. 21)	$\mathsf{aDec}(\mathsf{sk}, (\mathsf{r})\mathsf{dk}, \mathsf{ct}) \approx \perp$ // ct from $\mathcal{A}$	sk // $\mathsf{aKGen}(1^\lambda)$	$\perp$
ROB-CT+ (Def. 22)/[JS24,DG25]	$\mathsf{aDec}(\mathsf{sk}, (\mathsf{r})\mathsf{dk}, \mathsf{ct}) \approx \perp$ // ct from $\mathcal{A}$	sk // $\mathsf{aKGen}(1^\lambda)$	$\textsc{AEnc}((\mathsf{s})\mathsf{dk}, \cdot, \cdot)$
Standard CCA security			
IND-CCA	$\mathsf{Enc}(\mathsf{pk}, \mathsf{m}_0) \approx \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_1)$		$\textsc{Dec}(\mathsf{sk}, \cdot)$
CCA Security against the dictator			
STE-xCCA1 (Def. 23)	Normal Mode $\approx$ Anam. Mode $\mathsf{Enc}(\mathsf{m}) \approx \mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}})$	sk // $\mathsf{KGen}(1^\lambda)$ in NM $\mathsf{aKGen}(1^\lambda)$ in AM	$\perp$ in NM $\textsc{ADec}(\mathsf{sk}, (\mathsf{r})\mathsf{dk}, \cdot)$ in AM
DIND-xCCA1 (Def. 24)/[JS24]	$\mathsf{Enc}(\mathsf{m}) \approx \mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}})$	sk // $\mathsf{aKGen}(1^\lambda)$	$\textsc{ADec}(\mathsf{sk}, (\mathsf{r})\mathsf{dk}, \cdot)$
DIND-aCCA2 (Def. 25)/[JS24]	$\mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}}_0) \approx \mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}}_1)$	sk // $\mathsf{aKGen}(1^\lambda)$	$\textsc{ADec}(\mathsf{sk}, (\mathsf{r})\mathsf{dk}, \cdot)$
CCA security against the anamorphic senders			
SIND-xCCA1 (Def. 26)	$\mathsf{Enc}(\mathsf{m}) \approx \mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}})$	$(\mathsf{s})\mathsf{dk}$ // $\mathsf{aKGen}(1^\lambda)$	$\textsc{Dec}(\mathsf{sk}, \cdot)$ $\textsc{ADec}(\mathsf{sk}, (\mathsf{r})\mathsf{dk}, \cdot)$
SIND-xCCA2 (Def. 27)	$\mathsf{Enc}(\mathsf{m}_0) \approx \mathsf{aEnc}(\mathsf{m}_1, \hat{\mathsf{m}})$	$(\mathsf{s})\mathsf{dk}$ // $\mathsf{aKGen}(1^\lambda)$	$\textsc{Dec}(\mathsf{sk}, \cdot)$ $\textsc{ADec}(\mathsf{sk}, (\mathsf{r})\mathsf{dk}, \cdot)$
SIND-nCCA (Def. 28)	$\mathsf{Enc}(\mathsf{m}_0) \approx \mathsf{Enc}(\mathsf{m}_1)$	$(\mathsf{s})\mathsf{dk}$ // $\mathsf{aKGen}(1^\lambda)$	$\textsc{Dec}(\mathsf{sk}, \cdot)$ $\textsc{ADec}(\mathsf{sk}, (\mathsf{r})\mathsf{dk}, \cdot)$
SIND-aCCA1 (Def. 29)	$\mathsf{aEnc}(\mathsf{m}_0, \hat{\mathsf{m}}) \approx \mathsf{aEnc}(\mathsf{m}_1, \hat{\mathsf{m}})$	$(\mathsf{s})\mathsf{dk}$ // $\mathsf{aKGen}(1^\lambda)$	$\textsc{Dec}(\mathsf{sk}, \cdot)$ $\textsc{ADec}(\mathsf{sk}, (\mathsf{r})\mathsf{dk}, \cdot)$
SIND-aCCA2 (Def. 30)	$\mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}}_0) \approx \mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}}_1)$	$(\mathsf{s})\mathsf{dk}$ // $\mathsf{aKGen}(1^\lambda)$	$\textsc{Dec}(\mathsf{sk}, \cdot)$ $\textsc{ADec}(\mathsf{sk}, (\mathsf{r})\mathsf{dk}, \cdot)$
SIND-aCCA3 (Def. 31)	$\mathsf{aEnc}(\mathsf{m}_0, \hat{\mathsf{m}}_0) \approx \mathsf{aEnc}(\mathsf{m}_1, \hat{\mathsf{m}}_1)$	$(\mathsf{s})\mathsf{dk}$ // $\mathsf{aKGen}(1^\lambda)$	$\textsc{Dec}(\mathsf{sk}, \cdot)$ $\textsc{ADec}(\mathsf{sk}, (\mathsf{r})\mathsf{dk}, \cdot)$
CCA security against the collusion in the asymmetric setting			
CIND-xCCA1 (Def. 33)	$\mathsf{Enc}(\mathsf{m}) \approx \mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}})$	$\mathsf{sk}, \mathsf{sdk}$ // $\mathsf{aKGen}(1^\lambda)$	$\textsc{ADec}(\mathsf{sk}, \mathsf{rdk}, \cdot)$
CIND-aCCA2 (Def. 34)	$\mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}}_0) \approx \mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}}_1)$	$\mathsf{sk}, \mathsf{sdk}$ // $\mathsf{aKGen}(1^\lambda)$	$\textsc{ADec}(\mathsf{sk}, \mathsf{rdk}, \cdot)$

Table 2. Table of robustness and CCA security notions for $\mathsf{AME} = (\Pi, \Sigma)$ with $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ and $\Sigma = (\mathsf{aKGen}, \mathsf{aEnc}, \mathsf{aDec})$. We omit the input pk in Enc and pk, dk (or sdk) in aEnc in the second column.

definitions schemes should target to achieve maximal security. Our methodology also naturally extends to and uncovers further gaps within the CCA and asymmetric anamorphic settings.

2 Preliminaries

We use λ to denote the security parameter which is used implicitly throughout the paper. For a distribution $\mathcal{D}$, we let $x \leftarrow \mathcal{D}$ denote the operation of sampling x according to $\mathcal{D}$. For probabilistic algorithm $\mathcal{A}$, we let $y \leftarrow \mathcal{A}(x; r)$ or $y := \mathcal{A}(x; r)$ denote the operation of running $\mathcal{A}$ with input x and randomness r and assigning the output to y. If the randomness is uniformly sampled, we write $y \leftarrow \mathcal{A}(x)$. For deterministic algorithms $\mathcal{A}$, we also write as $y := \mathcal{A}(x)$ or $y := \mathcal{A}(x; r)$. "PPT" is short for probabilistic polynomial-time.

2.1 Public-Key Encryption

Definition 1 (Public-Key Encryption). *A public-key encryption (PKE) scheme is a triplet of algorithms* $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *defined in the following.*

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$. *On input the security parameter* λ*, the key generation algorithm outputs a secret key* sk *and a public key* pk.
- $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$. *On input* pk *and a message* m*, the encryption algorithm outputs a ciphertext* ct*. If the randomness* R *is specified, we denote as* $\mathsf{ct} := \mathsf{Enc}(\mathsf{pk}, \mathsf{m}; R)$.
- $\mathsf{m}/\perp \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$. *On input* sk *and* ct*, the decryption algorithm outputs a decryption* m *or a failure indicator* $\perp$.

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 (Sec. 5)	✓	✓	✓	✓	✓	✓	✓	✓	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 (Sec. 7.1)	✓	✓	✓	✓	✓	×	✓	✓	✓
Ours: rr-PKE + RO (Sec. 7.2)	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 3. Analysis of existing approaches to build AME. rr-PKE denotes randomness-recovering PKE. Certain items are generic compilers/paradigms, where security relies on the security of the underlying building blocks. "△" denotes that the scheme can be enhanced to achieve the specified security notion, but it was not addressed in the original paper. "✓*" refers to the enhanced transforms proposed in the full version of [JS24], which, like us, additionally propose DIND-aCPA2-secure schemes. "IC" and "RO" stand for ideal cipher and random oracle, respectively.

Correctness of Π. *For every message* m, *except with negligible probability in* λ *over the randomness of* KGen *and* Enc, *it holds that* $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \mathsf{m}$, *where* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$, *and* $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$.

Definition 2 (IND-CPA Security of PKE). *A PKE scheme* $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* IND-CPA*-secure, if, for every PPT adversary* $\mathcal{A}$, *the advantage*

$$\mathsf{Adv}^{\text{ind-cpa}}_{\Pi}(\mathcal{A}) := |\Pr[\text{IND-CPA}_{\Pi,1}(\mathcal{A}) = 1] - \Pr[\text{IND-CPA}_{\Pi,0}(\mathcal{A}) = 1]|$$

is negligible in λ, *where* $\text{IND-CPA}_{\Pi,b}(\mathcal{A})$ *for* $b \in \{0,1\}$ *are defined in Fig. 2.*

$\text{IND-CPA}_{\Pi,b}(\mathcal{A})$:	$\textsc{Chal}(\mathsf{m}_0, \mathsf{m}_1)$
1 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$	1 : **return** $\mathsf{Enc}(\mathsf{pk}, \mathsf{m}_b)$
2 : **return** $\mathcal{A}^{\textsc{Chal}}(\mathsf{pk})$	

Fig. 2. IND-CPA security game for PKE scheme Π.

Definition 3 (IND-CCA security of PKE). *A PKE scheme* $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* IND-CCA*-secure, if, for every PPT adversary* $\mathcal{A}$, *the advantage*

$$\mathsf{Adv}^{\text{ind-cca}}_{\Pi}(\mathcal{A}) := |\Pr[\text{IND-CCA}_{\Pi,1}(\mathcal{A}) = 1] - \Pr[\text{IND-CCA}_{\Pi,0}(\mathcal{A}) = 1]|$$

is negligible in λ, *where* $\text{IND-CCA}_{\Pi,b}(\mathcal{A})$ *for* $b \in \{0,1\}$ *are defined in Fig. 3.*

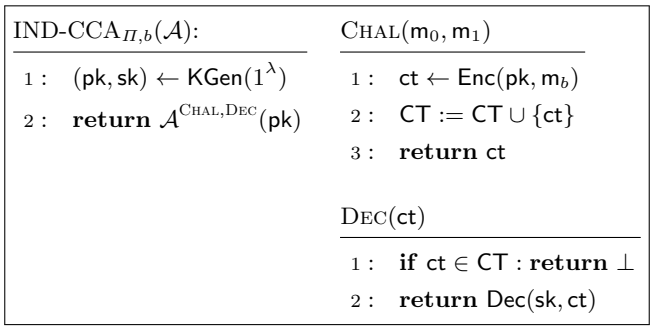


Fig. 3. IND-CCA security game for PKE scheme Π.

Definition 4 (Ciphertext Pseudorandomness of PKE). *A PKE scheme* $\Pi = (\mathsf{KGen},\mathsf{Enc},\mathsf{Dec})$ *is* PR-CCA*-secure, if, for every PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{pr-cca}}_{\Pi}(\mathcal{A}) := |\Pr[\text{PR-CCA}_{\Pi,1}(\mathcal{A}) = 1] - \Pr[\text{PR-CCA}_{\Pi,0}(\mathcal{A}) = 1]|$$

is negligible in λ*, where* $\text{PR-CCA}_{\Pi,b}(\mathcal{A})$ *for* $b \in \{0,1\}$ *are defined in Fig. 4.*

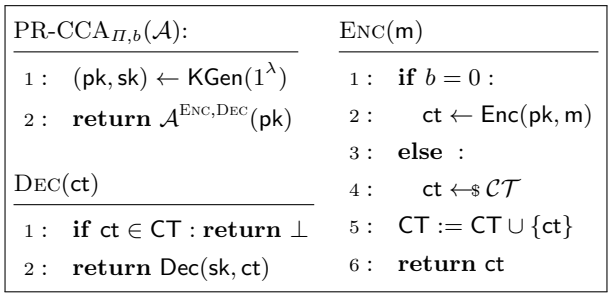


Fig. 4. PR-CCA security game for PKE scheme Π associated with ciphertext space $\mathcal{CT}$.

PR-CCA-secure PKE schemes can be constructed from a variety of assumptions, e.g., the oracle Diffie-Hellman assumption [Möl04], and the Matrix-DDH assumption [BKP14,HLQG18,LLHG23], and the Modular-LWE assumption [MX23].

2.2 Symmetric Encryption

Definition 5 (Symmetric Encryption). *A (randomized) symmetric encryption (SE) scheme is a triplet of algorithms* $\mathsf{SE} = (\mathsf{KGen},\mathsf{Enc},\mathsf{Dec})$ *defined in the following.*

- $\mathsf{k} \leftarrow \mathsf{KGen}(1^\lambda)$*: On input the security parameter* λ*, the key generation algorithm outputs a secret key* k*.*
- $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{k},\mathsf{m})$*: On input* k *and a message* m*, the encryption algorithm outputs a ciphertext* ct*. If the randomness* R *is specified, we denote as* $\mathsf{ct} := \mathsf{Enc}(\mathsf{k},\mathsf{m};R)$*.*
- $\mathsf{m}/\perp \leftarrow \mathsf{Dec}(\mathsf{k},\mathsf{ct})$*: On input* k *and* ct*, the decryption algorithm outputs a message* m *or a failure indicator* $\perp$*.*

Correctness of SE*. For every message* m*, except with negligible probability over the randomness of* KGen *and* Enc*, it holds that* $\mathsf{Dec}(\mathsf{k},\mathsf{ct}) = \mathsf{m}$*, where* $\mathsf{k} \leftarrow \mathsf{KGen}(1^\lambda)$*, and* $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{k},\mathsf{m})$*.*

Definition 6 (Ciphertext Pseudorandomness (PR-CT) of SE [PPY22,KPP+23b]). *A symmetric encryption scheme* $\mathsf{SE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* PR-CT*-secure, if, for every PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{pr-ct}}_{\mathsf{SE}}(\mathcal{A}) := |\Pr[\text{PR-CT}_{\mathsf{SE},1}(\mathcal{A}) = 1] - \Pr[\text{PR-CT}_{\mathsf{SE},0}(\mathcal{A}) = 1]|$$

is negligible in λ*, where* $\text{PR-CT}_{\mathsf{SE},b}(\mathcal{A})$ *for* $b \in \{0,1\}$ *are defined in Fig. 5.*

$\text{PR-CT}_{\mathsf{SE},b}(\mathcal{A})$:	$\textsc{Enc}(\mathsf{m})$
$\mathsf{k} \leftarrow \mathsf{KGen}(1^\lambda)$	**if** $b = 0$:
return $\mathcal{A}^{\textsc{Enc}}$	**return** $\mathsf{Enc}(\mathsf{k}, \mathsf{m})$
	else :
	return $\mathsf{ct} \leftarrow\!\!\$\ \{0,1\}^{\lvert\mathsf{Enc}(\mathsf{k},\mathsf{m})\rvert}$

Fig. 5. PR-CT security game for SE scheme SE.

2.3 Authenticated Encryption

Definition 7 (Authenticated Encryption). *An authenticated encryption (AE) scheme is a particular symmetric encryption scheme* $\mathsf{AE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *which additionally satisfies the following. For every PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{ae}}_{\mathsf{AE}}(\mathcal{A}) := |\Pr[\text{AE}_{\mathsf{AE},1}(\mathcal{A}) = 1] - \Pr[\text{AE}_{\mathsf{AE},0}(\mathcal{A}) = 1]|$$

is negligible in λ*, where* $\text{AE}_{\mathsf{AE},b}(\mathcal{A})$ *for* $b \in \{0,1\}$ *are defined in Fig. 6.*

$\text{AE}_{\mathsf{AE},b}(\mathcal{A})$:	$\textsc{Enc}(\mathsf{m})$	$\textsc{Dec}(\mathsf{ct})$
$\mathsf{k} \leftarrow \mathsf{KGen}(1^\lambda)$	**if** $b = 0$:	**if** $b = 0 \wedge \mathsf{ct} \notin \mathsf{CT}$:
return $\mathcal{A}^{\textsc{Enc},\textsc{Dec}}$	$\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{k}, \mathsf{m})$	**return** $\mathsf{Dec}(\mathsf{k}, \mathsf{ct})$
	else :	**else** :
	$\mathsf{ct} \leftarrow\!\!\$\ \{0,1\}^{\lvert\mathsf{Enc}(\mathsf{k},\mathsf{m})\rvert}$	**return** $\perp$
	$\mathsf{CT} := \mathsf{CT} \cup \{\mathsf{ct}\}$	
	return ct	

Fig. 6. Security game for AE scheme AE.

2.4 Pseudorandom Functions

Definition 8 (Pseudorandom Functions). *A pseudorandom function (PRF) is a family of functions* $\{\mathsf{F}_\mathsf{k} : \mathcal{X} \to \mathcal{Y}\}_{\mathsf{k}\in\mathcal{K}}$ *that satisfy the following properties.*

- *Given* k *and* $x \in \mathcal{X}$*, the computation of* $\mathsf{F}_\mathsf{k}(x)$ *is efficient.*
- *For any PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{prf}}_{\mathsf{F}}(\mathcal{A}) := \left|\Pr[\mathcal{A}^{\mathsf{F}_\mathsf{k}(\cdot)} = 1] - \Pr[\mathcal{A}^{\mathsf{R}(\cdot)} = 1]\right|$$

is negligible in λ*, where* $\mathsf{k} \leftarrow \mathcal{K}$*, and* R *is a random function from* $\mathcal{X}$ *to* $\mathcal{Y}$*.*

2.5 Anamorphic Encryption

Now we recall the syntax for symmetric and asymmetric anamorphic encryption (AME).

Definition 9 (Symmetric Anamorphic Encryption). *A symmetric anamorphic encryption scheme* AME *consists of a public-key encryption scheme* $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *and an associated anamorphic triplet* $\Sigma = (\mathsf{aKGen}, \mathsf{aEnc}, \mathsf{aDec})$. AME *has two modes: normal mode and anamorphic mode. When deployed in the normal mode, the primitive is identical to* Π. *When deployed in the anamorphic mode,* Π *is replaced by* Σ *where the algorithms are defined as follows:*

- $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{aKGen}(1^\lambda)$. *On input the security parameter* λ*, the anamorphic key generation algorithm outputs a key-pair* $(\mathsf{pk}, \mathsf{sk})$ *and a double key* dk.
- $\mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}})$. *On input the public key* pk*, the double key* dk*, a (normal) message* m *and an anamorphic message* $\hat{\mathsf{m}}$*, the anamorphic encryption algorithm outputs an (anamorphic) ciphertext* ct.
- $\hat{\mathsf{m}}/\perp \leftarrow \mathsf{aDec}(\mathsf{sk}, \mathsf{dk}, \mathsf{ct})$. *On input the secret key* sk*, the double key* dk*, and a ciphertext* ct*, the anamorphic decryption algorithm outputs an anamorphic message* $\hat{\mathsf{m}}$ *or a failure indicator* $\perp$.

Correctness of Σ. *For every (all-powerful) adversary* $\mathcal{A}$*, the probability*

$$\Pr\left[\begin{array}{l|ll} (\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{aKGen}(1^\lambda); (\mathsf{m}, \hat{\mathsf{m}}) \leftarrow \mathcal{A}(\mathsf{pk}); & : & \mathsf{aDec}(\mathsf{sk}, \mathsf{dk}, \mathsf{ct}_2) \neq \hat{\mathsf{m}} \\ \mathsf{ct}_1 \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}); \mathsf{ct}_2 \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}}) & & \vee\ \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}_2) \neq \mathsf{m} \\ & & \vee\ \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}_1) \neq \mathsf{m} \end{array}\right]$$

is negligible in λ.

Remark 1 (On Anamorphic Extensions). Banfi et al. [BGH+24] proposed *anamorphic extensions*, which, for deployment reasons, decouple double key generation from public/secret key-pair generation. While some anamorphic constructions use this decoupling [CCGM25,CGM24a], others inherently require the double key to be output alongside the key-pair [PPY22,PPY24]. We intend our unified treatment to apply comprehensively, so we allow anamorphic key generation to directly output the double key. Indeed, anamorphic extensions can be generically cast as anamorphic triplets, but not the other way around.

Remark 2 (On the Input of Anamorphic Decryption). By the definition in [PPY22], the anamorphic decryption uses only the double key dk to decrypt an anamorphic ciphertext. In this work we adopt a more generalized definition by requiring aDec to use not only dk but also the secret key sk, as defined in subsequent works [KPP+23b,PPY24,CGM24a,CGM25,DG25,DL25b]. As shown in the following sections, this generalisation enables us to analyse the security against the anamorphic sender who holds dk but not sk in both the symmetric setting and the asymmetric setting.

Remark 3 (Stateful AME). Banfi et al. [BGH+24] considered a stateful variant of anamorphic encryption where aEnc and aDec may input and output a state. This allows for different constructions and one can consider various security notions in this setting (e.g., where the state may or may not be given to the adversary, where the adversary can make oracle queries with adversarially chosen state, etc.), but we do not consider this further here.

Definition 10 (Asymmetric Anamorphic Encryption). *An asymmetric anamorphic encryption scheme* $\mathsf{AME} = (\Pi, \Sigma)$ *is defined similar to symmetric AME except that the double key* dk *is split into a sender double key* sdk *and a receiver double key* rdk. *Accordingly, the algorithms of* Σ *are as follows:*

- $(\mathsf{pk}, \mathsf{sk}, \mathsf{sdk}, \mathsf{rdk}) \leftarrow \mathsf{aKGen}(1^\lambda)$.
- $\mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{sdk}, \mathsf{m}, \hat{\mathsf{m}})$.
- $\hat{\mathsf{m}}/\perp \leftarrow \mathsf{aDec}(\mathsf{sk}, \mathsf{rdk}, \mathsf{ct})$.

The correctness is defined in a similar way and we omit here.

Remark 4 (On Defining Asymmetric AME). We decompose the double key into a sender double key sdk and a receiver double key rdk, as was done in [PPY24]. This is equivalent to the definitions in [BPRS25,CGM24a,DL26] where anamorphic key generation outputs a double key dk (held by the sender) as well as a trapdoor key tk (held by the receiver). However, we avoid formalizing these keys as a public double key and a secret double key, since the public double key is not public at all! In fact, it must be kept private against the dictator, lest the existence of an anamorphic channel be revealed.

To simplify the presentation and improve readability, we treat AME schemes as symmetric by default, unless stated otherwise. Further results for asymmetric AME are discussed in Section 6.

3 Security of Anamorphic Encryption in the CPA Setting

In this section, we conduct a systematic study of AME security definitions, and investigate relations among the various notions. An overview of our results is shown in Table 1 and Fig. 1. We defer the discussion on not well-defined and impossible definitions to Appendix A.

3.1 A Comprehensive Approach to CPA Definitions

Our methodology consists of three steps. Firstly, we systematically derive a comprehensive collection of security definitions for anamorphic encryption. Secondly, we analyze their relationships through implications and separations. Finally, we use these relationships to identify which definitions constructions should target.

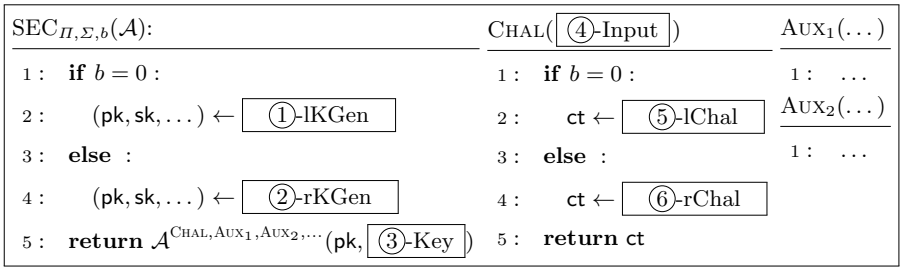


Fig. 7. A Template for defining security for an AME scheme $\mathsf{AME} = (\Pi, \Sigma)$. The auxiliary oracles $\textsc{Aux}_1, \textsc{Aux}_2$ are empty in the CPA setting but will be replaced with (standard) decryption oracle Dec and/or anamorphic decryption oracle aDec in the CCA setting (cf. Section 4).

Fig. 7 presents a generic indistinguishability-style security game SEC for an anamorphic encryption scheme $\mathsf{AME} = (\Pi, \Sigma)$, serving as a template we will procedurally fill out. The game begins by running left-or-right key generation (denoted ①-lKGen and ②-rKGen) dependent on a bit b, then providing the public key and any other relevant keys (denoted ③-Key) to the adversary. Note that we do not make this input dependent on b, as this would allow for trivial distinguishing. The specific algorithms and keys assigned to these slots are defined when filling out the template, and we call a particular collection of slot definitions a *key generation setting*. After key generation, the adversary accesses a challenge oracle Chal with inputs (denoted ④-Input) which performs left-or-right encryption (denoted ⑤-lChal and ⑥-rChal) also dependent on the bit b. The specific oracle inputs and left-or-right encryption algorithms assigned to these slots are similarly defined when filling out the template, and we call a particular collection of definitions a *challenge oracle setting*. Finally, the game defines auxiliary oracles $\textsc{Aux}_1$, $\textsc{Aux}_2$, ... that are empty in the CPA setting considered in this section, but will be instantiated on a definition-by-definition basis in the CCA setting (see Section 4).

With the generic template security definition established, we can now iterate through the possible key generation settings, summarized in Table 4. We begin with the key generation algorithms for $b = 0$ and

$b = 1$, which instantiate the template slots ①-lKGen and ②-rKGen. As slots ①-lKGen and ②-rKGen can only be either normal key generation KGen or anamorphic key generation aKGen, there are three possible configurations: KGen or aKGen is run depending on b, KGen is always run, or aKGen is always run. We can now consider which additional keys are given to the adversary in ③-Key. In any setting where KGen is run, we can only provide the adversary with the secret key sk as only $(\mathsf{pk}, \mathsf{sk})$ is output by the normal key generation process. In any setting where only aKGen is run, we can provide sk, the double key dk (or sending double key sdk for asymmetric schemes), or both sk and dk (or sdk) to the adversary. Overall, we get five key generation settings for symmetric and asymmetric anamorphic encryption each. Note that while some settings are well-defined, they are inherently unachievable as the adversary possesses all keys. We highlight these in red. All key generation settings are summarized in Table 4, where we also assign name to each (achievable) setting that reflect the adversary's implicit role and the underlying security context.

Table 4. Possible key generation settings. The cases highlighted in red are not well-defined (or unachievable).

	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	$(\mathsf{sk}, \mathsf{dk})$	$(\mathsf{sk}, \mathsf{sdk})$

We now move to the challenge oracle settings, summarized in Table 5. We first iterate through all settings independently of the key generation settings and will later analyze which combinations are possible. Analogous to key generation, we can consider which encryption algorithms are run for $b = 0$ and $b = 1$, which instantiate the template slots ⑤-lChal and ⑥-rChal. There are three possible configurations: normal encryption Enc or anamorphic encryption aEnc is run depending on b, Enc is always run, or aEnc is always run. To consider the adversarially-chosen challenge oracle input ④-Input, note that left-or-right inputs are only possible for message types that both algorithms accept (e.g., left-or-right anamorphic messages $(\hat{\mathsf{m}}_0, \hat{\mathsf{m}}_1)$ are only defined if aEnc is run in both worlds; otherwise, the anamorphic message must remain fixed across $b = 0$ and $b = 1$). For the first configuration, either normal or anamorphic encryption are run; here, slot ④-Input can be a plain message and a fixed anamorphic message $(\mathsf{m}, \hat{\mathsf{m}})$, or left-or-right plain messages with a fixed anamorphic message $(\mathsf{m}_0, \mathsf{m}_1, \hat{\mathsf{m}})$. For the second configuration, normal encryption is always run, so ④-Input can only be left-or-right plain messages $(\mathsf{m}_0, \mathsf{m}_1)$. Finally, in the third configuration, anamorphic encryption is always run and ④-Input can be left-or-right plain messages with a fixed anamorphic message $(\mathsf{m}_0, \mathsf{m}_1, \hat{\mathsf{m}})$, fixed plain message with left-or-right anamorphic messages $(\mathsf{m}, \hat{\mathsf{m}}_0, \hat{\mathsf{m}}_1)$, or left-or-right plain and anamorphic messages $(\mathsf{m}_0, \mathsf{m}_1, \hat{\mathsf{m}}_0, \hat{\mathsf{m}}_1)$. The possible challenge oracle settings are shown in Table 5, where we label each using prefixes 'x', 'n', or 'a' to denote normal or anamorphic encryption, normal encryption always, or anamorphic encryption always, respectively. The numbers distinguish which left-or-right inputs the adversary may query to the challenge oracle.

Having established both the key generation and challenge oracle settings, we now consider every possible combination in Table 6. To keep notions compact and easy to identify the information an adversary obtains, we denote a security notion with 'A-B', where 'A' $\in$ {'STE', 'DIND', 'SIND', 'CIND'} denotes the key generation setting listed in Table 4, and 'B' presents the challenge oracle setting listed in Table 5. If a setting is not well-defined (e.g., the challenge oracle setting utilizes a key that was not generated in the key generation setting), we mark it as such. For settings that are well-defined but unachievable are marked as impossible, and we later detail these definitions. For ones that exist in literature, we provide the corresponding citation; otherwise, we highlight new definitions in blue.

Table 5. Possible challenge oracle settings.

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

Table 6. Possible CPA definitions for anamorphic encryption. New definitions are highlighted in blue. The SIND-aCPA3 notion was introduced in the asymmetric setting in [CGM24a] and is first considered here in the symmetric setting. The CIND-aCPA2 notion was also considered in an concurrent work [BPRS25].

	STE (sk)	DIND (sk)	SIND (dk or sdk)	CIND ($\mathsf{sk}, \mathsf{sdk}$)
xCPA1	STE-xCPA1 [PPY22]	DIND-xCPA1 [PPY22]	SIND-xCPA1 (Def. 14)	CIND-xCPA1 (Def. 33)
xCPA2	Impossible	Impossible	SIND-xCPA2 (Def. 15)	Impossible
nCPA	Impossible	Impossible	SIND-nCPA (Def. 16)	Impossible
aCPA1	Not Well-Defined	Impossible	SIND-aCPA1 [KPP+23b]	Impossible
aCPA2	Not Well-Defined	DIND-aCPA2 [PPY22]	SIND-aCPA2 (Def. 18)	CIND-aCPA2 (Def. 34)/[BPRS25]
aCPA3	Not Well-Defined	Impossible	SIND-aCPA3 (Def. 19)/[CGM24a]	Impossible

We then focus on those well-defined and achievable definitions. Interestingly, although these definitions were derived procedurally, we find that each is meaningful in that they capture specific security notions that correspond to scenarios users may reasonably want to target. For each definition, we explicitly write out the game (properly defining auxiliary oracles if necessary), discuss these scenarios, and note whether or not the definition exists in prior literature. Definitions that are not well-defined or impossible to achieve are discussed in Appendix A.

3.2 Security Notions against the Dictator

STE-xCPA1. Fig. 8 shows the STE-xCPA1 security game, which serves as the fundamental security requirement for AME in the original work [PPY22] and has been followed by all subsequent studies. At a high level, it captures a scenario where a dictator compels the disclosure of the secret key sk and must distinguish between a normal user who generates keys and receives normal encryptions, and an anamorphic user who has run anamorphic key generation and receives ciphertexts containing hidden anamorphic messages. Note that this game does not have any auxiliary oracles, as the adversary can simulate normal public key encryption using pk (indeed, this applies to all games we discuss) and anamorphic encryption is not always defined (as dk may not exist). This definition represents the minimal requirement for a meaningful notion of anamorphism and was proposed in [PPY22], which first proposed anamorphic encryption. There, it is formalized by two games labeled $\mathsf{NormalGame}$ and $\mathsf{FullyAGame}$ and is captured in Definition 2 as one of several properties anamorphic encryption schemes should satisfy.[7] The adaptation to asymmetric anamorphic encryption was first written in [CGM24a] which proposed the latter concept. There, it is formalized by two games labeled RealG and $\mathsf{AnamorphicG}$ and is captured in Definition 7.

[7] In more detail, [PPY22] includes STE-xCPA1 as part of the definition of a *secure* AME scheme (plus regular IND-CPA security and that $\mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \cdot)$ and $\mathsf{aDec}(\mathsf{sk}, \mathsf{dk}, \cdot)$ form a symmetric encryption scheme for any m). To better study the relations among different security properties, we formalize STE-xCPA1 security as above.

Definition 11 (STE-xCPA1 Security of AME [PPY22]). *An AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$ *is* STE-xCPA1*-secure, if for every PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{ste-xcpa1}}_{\Pi,\Sigma}(\mathcal{A}) := |\Pr[\text{STE-xCPA1}_{\Pi,\Sigma,1}(\mathcal{A}) = 1] - \Pr[\text{STE-xCPA1}_{\Pi,\Sigma,0}(\mathcal{A}) = 1]|$$

is negligible in λ*, where* $\text{STE-xCPA1}_{\Pi,\Sigma,b}(\mathcal{A})$ *are defined in Fig. 8.*

$\text{STE-xCPA1}_{\Pi,\Sigma,b}(\mathcal{A})$:

1 : **if** $b = 0$:
2 : $\quad(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$
3 : **else** :
4 : $\quad(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{aKGen}(1^\lambda)$
5 : **return** $\mathcal{A}^{\text{CHAL}}(\mathsf{pk}, \mathsf{sk})$

$\text{CHAL}(\mathsf{m}, \hat{\mathsf{m}})$

1 : **if** $b = 0$:
2 : $\quad\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$
3 : **else** :
4 : $\quad\mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}})$
5 : **return** ct

$\text{STE-xCPA1}_{\Pi,\Sigma,b}(\mathcal{A})$:

1 : **if** $b = 0$:
2 : $\quad(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$
3 : **else** :
4 : $\quad(\mathsf{pk}, \mathsf{sk}, \mathsf{sdk}, \mathsf{rdk}) \leftarrow \mathsf{aKGen}(1^\lambda)$
5 : **return** $\mathcal{A}^{\text{CHAL}}(\mathsf{pk}, \mathsf{sk})$

$\text{CHAL}(\mathsf{m}, \hat{\mathsf{m}})$

1 : **if** $b = 0$:
2 : $\quad\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$
3 : **else** :
4 : $\quad\mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{sdk}, \mathsf{m}, \hat{\mathsf{m}})$
5 : **return** ct

Fig. 8. STE-xCPA1 security games [PPY22] for a symmetric (left) or an asymmetric (right) AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

DIND-xCPA1. Fig. 9 shows the DIND-xCPA1 security game for symmetric (left) and asymmetric (right) anamorphic encryption. At a high level, it captures a scenario where a dictator compels the disclosure of the secret key sk and must distinguish between normal encryptions and anamorphic encryptions sent to an anamorphic user, i.e., a user who has run anamorphic key generation. Note that this game provides an auxiliary non-challenge anamorphic encryption oracle, which captures the realistic setting that this user will likely be receiving anamorphic ciphertexts in the background (that the dictator can influence), but the dictator may be targeting specific communication it wants to distinguish (for instance from a specific sender that the dictator wants to track). This definition was proposed in [PPY22], though they do not include the non-challenge anamorphic encryption oracle. Their version is formalized by two games labeled aneGame and FullyAGame and is discussed in Lemma 2. It is easy to prove that the two versions are (polynomially) equivalent.

Definition 12 (DIND-xCPA1 Security of AME). *An AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$ *is DIND-xCPA1-secure, if for every PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{dind-xcpa1}}_{\Pi,\Sigma}(\mathcal{A}) := |\Pr[\text{DIND-xCPA1}_{\Pi,\Sigma,1}(\mathcal{A}) = 1] - \Pr[\text{DIND-xCPA1}_{\Pi,\Sigma,0}(\mathcal{A}) = 1]|$$

is negligible in λ*, where* $\text{DIND-xCPA1}_{\Pi,\Sigma,b}(\mathcal{A})$ *are defined in Fig. 9.*

DIND-aCPA2. Fig. 10 shows the DIND-aCPA2 security game for symmetric (left) and asymmetric (right) anamorphic encryption. At a high level, it captures a scenario where a dictator compels the disclosure of the secret key sk and attempts to learn information about an anamorphic message contained within an anamorphic ciphertext sent to an anamorphic user, i.e., a user who has run anamorphic key generation. This

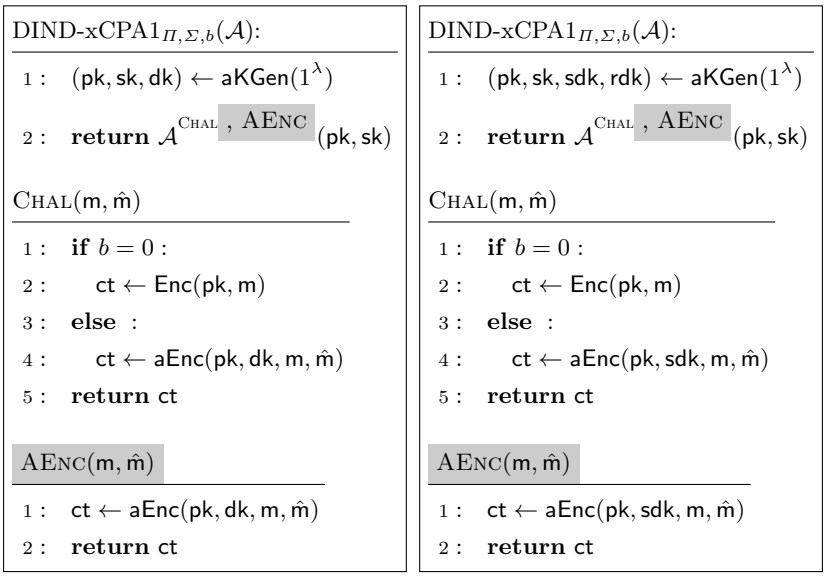


Fig. 9. DIND-xCPA1 security games for a symmetric (left) or an asymmetric (right) AME scheme $\mathsf{AME} = (\Pi, \Sigma)$, where the two different versions, with or without oracle AEnc are polynomially equivalent.

definition captures the above via a standard IND-CPA-style definition for the anamorphic message, where the adversary can influence both the plain and anamorphic messages. No auxiliary oracle for anamorphic encryption is needed, as the adversary can simulate this by querying $(\mathsf{m}, \hat{\mathsf{m}}, \hat{\mathsf{m}})$ to the challenge oracle. A single-challenge version of the definition was proposed in [PPY22]. There, it is formalized by the game labeled IndCPA^η and is discussed in Theorem 1 and Lemma 3.

Definition 13 (DIND-aCPA2 Security of AME). *An AME scheme $\mathsf{AME} = (\Pi, \Sigma)$ is DIND-aCPA2-secure, if for every PPT adversary $\mathcal{A}$, the advantage*

$$\mathsf{Adv}^{\text{dind-acpa2}}_{\Pi,\Sigma}(\mathcal{A}) := |\Pr[\text{DIND-aCPA2}_{\Pi,\Sigma,1}(\mathcal{A}) = 1] - \Pr[\text{DIND-aCPA2}_{\Pi,\Sigma,0}(\mathcal{A}) = 1]|$$

is negligible in λ, where DIND-aCPA2$_{\Pi,\Sigma,b}(\mathcal{A})$ *are defined in Fig. 10.*

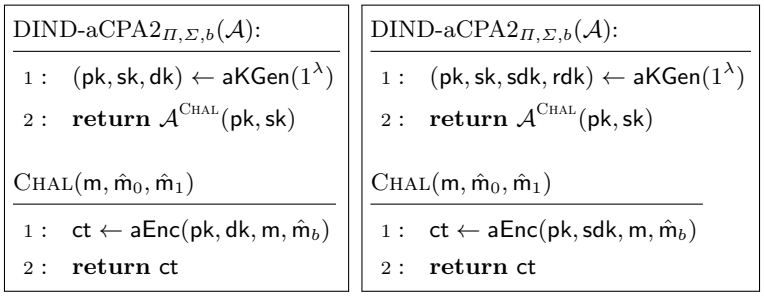


Fig. 10. DIND-aCPA2 security games for a symmetric (left) or an asymmetric (right) AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

We have the following implications.[8]

[8] This was proven as Lemma 1 in the full version of [PPY22].

$$\text{STE-xCPA1} \implies \text{DIND-xCPA1} \implies \text{DIND-aCPA2}.$$

Theorem 1. *For an AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$*, STE-xCPA1 implies DIND-xCPA1, and DIND-xCPA1 implies DIND-aCPA2.*

Proof. We first prove the implication from STE-xCPA1 to DIND-xCPA1. Note that the security games $\text{STE-xCPA1}_{\Pi,\Sigma,1}$ and $\text{DIND-xCPA1}_{\Pi,\Sigma,1}$ are identical when $b = 1$. Therefore, given the indistinguishability between $\text{STE-xCPA1}_{\Pi,\Sigma,0}$ and $\text{STE-xCPA1}_{\Pi,\Sigma,1}$, to prove DIND-xCPA1, it suffices to prove the indistinguishability between $\text{STE-xCPA1}_{\Pi,\Sigma,0}$ and $\text{DIND-xCPA1}_{\Pi,\Sigma,0}$. To this end, we construct a reduction algorithm $\mathcal{A}'$ to show that, if a PPT adversary $\mathcal{A}$ can distinguish these two games, then $\mathcal{A}'$ is able to distinguish $\text{STE-xCPA1}_{\Pi,\Sigma,0}(\mathcal{A}')$ from $\text{STE-xCPA1}_{\Pi,\Sigma,1}(\mathcal{A}')$.

$\mathcal{A}'$ simulates the experiment for $\mathcal{A}$ as follows. It first obtains $(\mathsf{pk}, \mathsf{sk})$ from the challenger of the experiment $\text{STE-xCPA1}_{\Pi,\Sigma,b}(\mathcal{A}')$ ($b \in \{0, 1\}$), and passes $(\mathsf{pk}, \mathsf{sk})$ to $\mathcal{A}$. Whenever $\mathcal{A}'$ receives a challenge query $\textsc{Chal}(\mathsf{m}, \hat{\mathsf{m}})$ from $\mathcal{A}$, it returns $\mathsf{Enc}(\mathsf{pk}, \mathsf{m})$. Finally, $\mathcal{A}'$ outputs what $\mathcal{A}$ outputs.

It is easy to see that if $b = 0$, then $\mathcal{A}'$ perfectly simulates $\text{STE-xCPA1}_{\Pi,\Sigma,0}(\mathcal{A})$, and if $b = 1$, then $\mathcal{A}'$ perfectly simulates $\text{DIND-xCPA1}_{\Pi,\Sigma,0}(\mathcal{A})$. Therefore, $\mathcal{A}'$ can distinguish $\text{STE-xCPA1}_{\Pi,\Sigma,0}(\mathcal{A}')$ from $\text{STE-xCPA1}_{\Pi,\Sigma,1}(\mathcal{A}')$ as long as $\mathcal{A}$ can distinguish $\text{STE-xCPA1}_{\Pi,\Sigma,0}(\mathcal{A})$ from $\text{DIND-xCPA1}_{\Pi,\Sigma,0}(\mathcal{A})$, which finishes the proof of STE-xCPA1 $\implies$ DIND-xCPA1.

According to DIND-xCPA1 security, the game $\text{DIND-xCPA1}_{\Pi,\Sigma,0}$ (where the challenge oracle returns $\mathsf{Enc}(\mathsf{pk}, \mathsf{m})$) is indistinguishable to both game $\text{DIND-aCPA2}_{\Pi,\Sigma,0}$ (where the challenge oracle returns $\mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}}_0)$) and game $\text{DIND-aCPA2}_{\Pi,\Sigma,1}$ (where the challenge oracle returns $\mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}}_1)$). As a result, $\text{DIND-aCPA2}_{\Pi,\Sigma,0}$ and $\text{DIND-aCPA2}_{\Pi,\Sigma,1}$ are indistinguishable, which concludes the proof of DIND-xCPA1 $\implies$ DIND-aCPA2. □

3.3 Security Notions against Anamorphic Senders

In this subsection we explore notions related to the security against anamorphic senders.

SIND-xCPA1. Fig. 11 shows the SIND-xCPA1 security games for AME. At a high level, it captures a scenario where an anamorphic sender trusted with the double key dk attempts to distinguish between normal encryptions and anamorphic encryptions sent (by another user trusted with the double key) to the anamorphic user, i.e., a user who has run anamorphic key generation. Their failure to do so ensures that such a sender is unable to determine whether or not other users are sending anamorphic messages to the recipient. No auxiliary oracle for anamorphic encryption is needed, as the adversary can simulate this using the $(\mathsf{pk}, \mathsf{dk})$ it obtained. To our knowledge, this definition has not been proposed before.

Definition 14 (SIND-xCPA1 Security of AME). *An AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$ *is* SIND-xCPA1*-secure, if for every PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{sind-xcpa1}}_{\Pi,\Sigma}(\mathcal{A}) := |\Pr[\text{SIND-xCPA1}_{\Pi,\Sigma,1}(\mathcal{A}) = 1] - \Pr[\text{SIND-xCPA1}_{\Pi,\Sigma,0}(\mathcal{A}) = 1]|$$

is negligible in λ*, where* $\text{SIND-xCPA1}_{\Pi,\Sigma,b}(\mathcal{A})$ *are defined in Fig. 11.*

SIND-xCPA2. Fig. 12 shows the SIND-xCPA2 security game for AME. At a high level, it captures a scenario where an anamorphic sender trusted with the double key dk attempts to distinguish between normal encryptions and anamorphic encryptions sent (by another user trusted with the double key) to an anamorphic user who has run anamorphic key generation, using influence they have over the plain message in an attempt to uncover which ciphertexts contain anamorphic messages. No auxiliary oracle for anamorphic encryption is needed, as the adversary can simulate this using the $(\mathsf{pk}, \mathsf{dk})$ it obtained. To our knowledge, this definition has not been proposed before.

Definition 15 (SIND-xCPA2 Security of AME). *An AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$ *is SIND-xCPA2-secure, if for every PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{sind-xcpa2}}_{\Pi,\Sigma}(\mathcal{A}) := |\Pr[\text{SIND-xCPA2}_{\Pi,\Sigma,1}(\mathcal{A}) = 1] - \Pr[\text{SIND-xCPA2}_{\Pi,\Sigma,0}(\mathcal{A}) = 1]|$$

is negligible in λ*, where* $\text{SIND-xCPA2}_{\Pi,\Sigma,b}(\mathcal{A})$ *are defined in Fig. 12.*

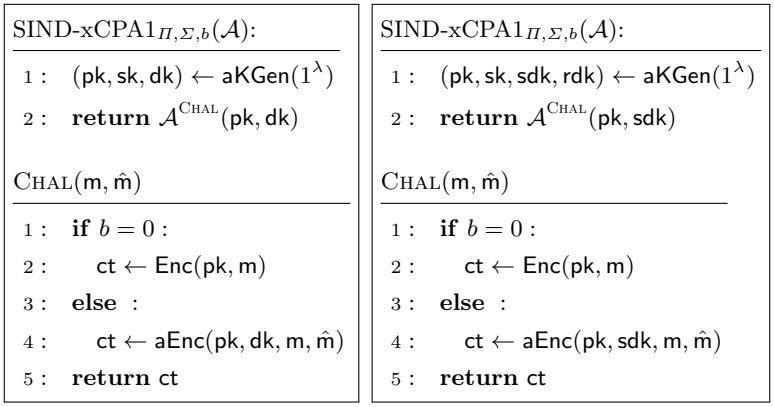


Fig. 11. SIND-xCPA1 security games for a symmetric (left) or an asymmetric (right) AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

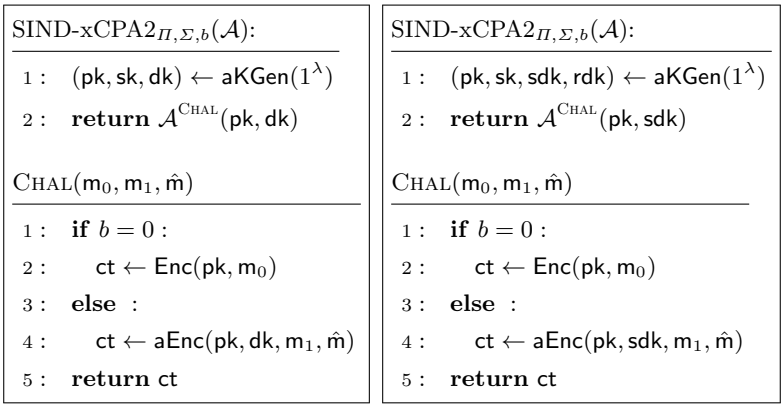


Fig. 12. SIND-xCPA2 security game for a symmetric (left) or an asymmetric (right) AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

SIND-nCPA. Fig. 13 shows the SIND-nCPA security game for symmetric (left) and asymmetric (right) anamorphic encryption. At a high level, it captures a scenario where an anamorphic sender trusted with the double key dk attempts to learn information about a plain message contained within an normal ciphertext sent to an anamorphic user, i.e., a user who has run anamorphic key generation. Such a ciphertext may come from *any* sender (not just an anamorphic one). This definition captures the above via a standard IND-CPA-style definition for a normally encrypted ciphertext message, where the adversarial sender can influence the plain message. No auxiliary oracle for anamorphic encryption is needed, as the adversary can simulate this using the $(\mathsf{pk}, \mathsf{dk})$ it obtained. To our knowledge, this definition has not been considered in prior literature.

Definition 16 (SIND-nCPA Security of AME). *An AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$ *is* SIND-nCPA*-secure, if for every PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{sind-ncpa}}_{\Pi,\Sigma}(\mathcal{A}) := |\Pr[\text{SIND-nCPA}_{\Pi,\Sigma,1}(\mathcal{A}) = 1] - \Pr[\text{SIND-nCPA}_{\Pi,\Sigma,0}(\mathcal{A}) = 1]|$$

is negligible in λ*, where* $\text{SIND-nCPA}_{\Pi,\Sigma,b}(\mathcal{A})$ *are defined in Fig. 13.*

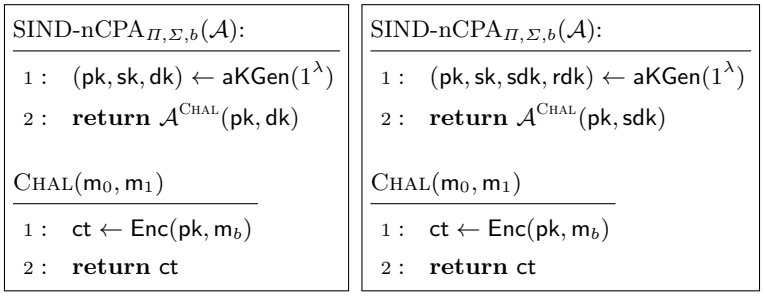


Fig. 13. SIND-nCPA security games for a symmetric (left) or an asymmetric (right) AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

In Theorem 8 we will see that SIND-nCPA security does not follow from existing IND-CPA security and STE-xCPA1 security (cf. Def. 11).

SIND-aCPA1. Fig. 14 shows the SIND-aCPA1 security game for symmetric (left) and asymmetric (right) anamorphic encryption. At a high level, it captures a scenario where an anamorphic sender trusted with the double key dk attempts to learn information about a normal message contained within an anamorphic ciphertext sent to an anamorphic user, i.e., a user who has run anamorphic key generation. This definition captures the above via a standard IND-CPA-style definition for the normal message encrypted in an anamorphic ciphertext, where the adversarial sender can influence the anamorphic message. No auxiliary oracle for anamorphic encryption is needed, as the adversary can simulate this using the $(\mathsf{pk}, \mathsf{dk})$ it obtained. A single-challenge version of the definition was proposed in [KPP+23b]. There, it is formalized by the game labeled $\mathsf{SingleAnG}^\eta$ and Definitions 8 and 9.

Definition 17 (SIND-aCPA1 Security of AME). *An AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$ *is SIND-aCPA1-secure, if for every PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{sind-acpa1}}_{\Pi,\Sigma}(\mathcal{A}) := |\Pr[\text{SIND-aCPA1}_{\Pi,\Sigma,1}(\mathcal{A}) = 1] - \Pr[\text{SIND-aCPA1}_{\Pi,\Sigma,0}(\mathcal{A}) = 1]|$$

is negligible in λ*, where* $\text{SIND-aCPA1}_{\Pi,\Sigma,b}(\mathcal{A})$ *are defined in Fig. 14.*

SIND-aCPA2. Fig. 15 shows the SIND-aCPA2 security game AME. At a high level, it captures a scenario where an anamorphic sender trusted with the double key dk attempts to learn information about an anamorphic message contained within an anamorphic ciphertext sent to an anamorphic user, i.e., a user who has

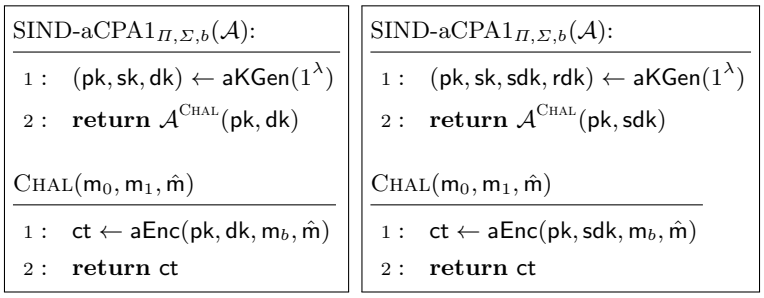


Fig. 14. SIND-aCPA1 security games for a symmetric (left) or an asymmetric (right) AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

run anamorphic key generation. This definition captures the above via a standard IND-CPA-style definition for the anamorphic message encrypted in an anamorphic ciphertext, where the adversarial sender can influence the anamorphic message. No auxiliary oracle for anamorphic encryption is needed, as the adversary can simulate this by querying $(\mathsf{m}, \hat{\mathsf{m}}, \hat{\mathsf{m}})$ to the challenge oracle. A single-challenge version with the message m in oracle $\text{AENC}(\mathsf{m}, \cdot)$ being fixed was originally defined in [PPY22] (the IndCPA security), and a single-challenge version of this definition in the asymmetric setting was considered in an early version of [BPRS25] (the GsIND-CPA security), which was uploaded to ePrint after this work was first uploaded.

Definition 18 (SIND-aCPA2 Security of AME). *An AME scheme $\mathsf{AME} = (\Pi, \Sigma)$ is SIND-aCPA2-secure, if for every PPT adversary $\mathcal{A}$, the advantage*

$$\mathsf{Adv}^{\text{sind-acpa2}}_{\Pi,\Sigma}(\mathcal{A}) := |\Pr[\text{SIND-aCPA2}_{\Pi,\Sigma,1}(\mathcal{A}) = 1] - \Pr[\text{SIND-aCPA2}_{\Pi,\Sigma,0}(\mathcal{A}) = 1]|$$

is negligible in λ, where SIND-aCPA2$_{\Pi,\Sigma,b}(\mathcal{A})$ *are defined in Fig. 15.*

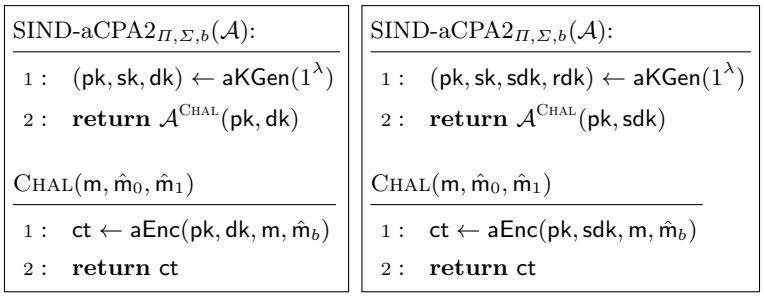


Fig. 15. SIND-aCPA2 security game for a symmetric (left) or an asymmetric (right) AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

SIND-aCPA3. Fig. 16 shows the SIND-aCPA3 security game for symmetric (left) and asymmetric (right) anamorphic encryption. At a high level, it captures a scenario where an anamorphic sender trusted with the double key dk attempts to uncover any information it can about normal or anamorphic messages contained within anamorphic ciphertexts sent to an anamorphic user, i.e., a user who has run anamorphic key generation. This definition captures the above via a standard IND-CPA-style definition for both the normal and anamorphic message encrypted in an anamorphic ciphertext, where the adversarial sender can influence the anamorphic message. No auxiliary oracle for anamorphic encryption is needed, as the adversary can simulate this using the $(\mathsf{pk},\mathsf{dk})$ it obtained. This notion was first proposed in [CGM24a], which wrote it for asymmetric anamorphic encryption (which thus also applies to symmetric anamorphic encryption). There, it is formalized in Definition 10 by a game labeled FAsyAnam-IND-CPA.

Definition 19 (SIND-aCPA3 Security of AME). *An AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$ *is* SIND-aCPA3*-secure, if for every PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{sind-acpa3}}_{\Pi,\Sigma}(\mathcal{A}) := |\Pr[\text{SIND-aCPA3}_{\Pi,\Sigma,1}(\mathcal{A}) = 1] - \Pr[\text{SIND-aCPA3}_{\Pi,\Sigma,0}(\mathcal{A}) = 1]|$$

is negligible in λ*, where* $\text{SIND-aCPA3}_{\Pi,\Sigma,b}(\mathcal{A})$ *are defined in Fig. 16.*

$\text{SIND-aCPA3}_{\Pi,\Sigma,b}(\mathcal{A})$:

1 : $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{aKGen}(1^\lambda)$
2 : **return** $\mathcal{A}^{\text{CHAL}}(\mathsf{pk}, \mathsf{dk})$

$\text{CHAL}(\mathsf{m}_0, \mathsf{m}_1, \hat{\mathsf{m}}_0, \hat{\mathsf{m}}_1)$

1 : $\mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}_b, \hat{\mathsf{m}}_b)$
2 : **return** ct

$\text{SIND-aCPA3}_{\Pi,\Sigma,b}(\mathcal{A})$:

1 : $(\mathsf{pk}, \mathsf{sk}, \mathsf{sdk}, \mathsf{rdk}) \leftarrow \mathsf{aKGen}(1^\lambda)$
2 : **return** $\mathcal{A}^{\text{CHAL}}(\mathsf{pk}, \mathsf{sdk})$

$\text{CHAL}(\mathsf{m}_0, \mathsf{m}_1, \hat{\mathsf{m}}_0, \hat{\mathsf{m}}_1)$

1 : $\mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{sdk}, \mathsf{m}_b, \hat{\mathsf{m}}_b)$
2 : **return** ct

Fig. 16. SIND-aCPA3 security games for a symmetric (left) or an asymmetric (right) AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

It is straightforward to see that SIND-aCPA3 implies SIND-aCPA1 (when $\hat{\mathsf{m}}_0 = \hat{\mathsf{m}}_1$) and SIND-aCPA2 (when $\mathsf{m}_0 = \mathsf{m}_1$) in the challenge queries.

3.4 Relations among Notions in the CPA Setting

In this section, we investigate the relations among various security notions. Perhaps surprisingly, we find that most notions, when considered in isolation, do not imply one another, due to the incomparability between sk and dk. We therefore study implication relationships under additional assumptions, specifically STE-xCPA1 or SIND-xCPA1, and show that in these settings, certain notions become strictly stronger than others. As a result, we identify three fundamental notions, STE-xCPA1, SIND-xCPA1, and SIND-nCPA, which together imply all other notions, including the basic IND-CPA notion. In particular one may target these three notions when deploying an AME scheme.

For simplicity, we establish our results primarily in the symmetric setting. The corresponding asymmetric results follow straightforwardly. We discuss the two asymmetric-only notions, CIND-xCPA1 and CIND-aCPA2, in Section 6.

Theorem 2 (SIND-aCPA(1/2/3) $\not\Rightarrow$ DIND-aCPA2). *For an AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$*, SIND-aCPA(1/2/3) does not imply DIND-aCPA2, and vice versa.*

Proof. One direction, SIND-aCPA1 $\not\Rightarrow$ DIND-aCPA2, can be proved via a counterexample such that $\mathsf{dk} = \perp$, and the anamorphic encryption algorithm $\mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}})$ returns $\mathsf{PKE.Enc}(\mathsf{pk}, (\mathsf{m}||\hat{\mathsf{m}}))$ directly, where PKE is an IND-CPA-secure PKE scheme. It is easy to see that given sk, both m and $\hat{\mathsf{m}}$ can be trivially derived and hence DIND-aCPA2 does not hold. However, it still satisfies SIND-aCPA1 security ($\mathsf{aEnc}(\mathsf{pk}, \mathsf{dk} = \perp, \mathsf{m}_0, \hat{\mathsf{m}}) \approx \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk} = \perp, \mathsf{m}_1, \hat{\mathsf{m}})$). This also proves that SIND-aCPA(2/3) $\not\Rightarrow$ DIND-aCPA2.

The other direction, SIND-aCPA(1/2/3) $\not\Leftarrow$ DIND-aCPA2, can be proved via a construction where sk is contained in dk. Therefore, with the knowledge of dk one is able to perform both normal and anamorphic decryption. □

Remark 5. Theorem 2 highlights how the secret key sk and the double key dk are incomparable. Such inherent incomparability highlights two distinct aspects of security in an AME scheme: security against the dictator and security against the anamorphic sender. As a result, this reminds us to be cautious when designing (or more precisely, deploying) a "secure" AME scheme.

Relations among SIND-aCPA(1/2/3). According to the definitions we already know that SIND-aCPA3 implies SIND-aCPA(1/2). Below we show that conditioned on STE-xCPA1, SIND-aCPA3 is (still) strictly stronger than SIND-aCPA(1/2) (Theorems 3 and 4). Later we further show that conditioned on SIND-xCPA1, the two notions SIND-aCPA1 and SIND-aCPA3 are equivalent (Theorem 11), and both are strictly stronger than SIND-aCPA2 (Theorem 12).

Theorem 3 (SIND-aCPA1 $\wedge$ STE-xCPA1 $\not\Rightarrow$ SIND-aCPA3). *For an AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$*, SIND-aCPA1 security and STE-xCPA1 security together do not imply SIND-aCPA3 security.*

Proof. Let $\widetilde{\mathsf{PKE}} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure PKE scheme, and $\widetilde{\mathsf{SE}} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a ciphertext-pseudorandom (PR-CT) secure symmetric encryption scheme. We construct an AME scheme $\mathsf{AME} = (\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}), \Sigma = (\mathsf{aKGen}, \mathsf{aEnc}, \mathsf{aDec}))$ as follows.

- The key generation algorithm returns $(\mathsf{pk}, \mathsf{sk}) \leftarrow \widetilde{\mathsf{PKE}}.\mathsf{KGen}(1^\lambda)$.
- The anamorphic key generation algorithm returns $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk})$, where $(\mathsf{pk}, \mathsf{sk}) \leftarrow \widetilde{\mathsf{PKE}}.\mathsf{KGen}(1^\lambda)$, and $\mathsf{dk} \leftarrow \widetilde{\mathsf{SE}}.\mathsf{KGen}(1^\lambda)$.
- The encryption algorithm returns $\mathsf{ct} = (\widetilde{\mathsf{PKE}}.\mathsf{Enc}(\mathsf{pk}, \mathsf{m})||str)$, where str is a string randomly sampled from the ciphertext space of $\widetilde{\mathsf{SE}}$.[9]
- The anamorphic encryption algorithm returns $\mathsf{ct} = (\widetilde{\mathsf{PKE}}.\mathsf{Enc}(\mathsf{pk}, \mathsf{m})||\widetilde{\mathsf{SE}}.\mathsf{Enc}(\mathsf{dk}, \hat{\mathsf{m}}))$.

It is easy to see that $\mathsf{AME} = (\Pi, \Sigma)$ satisfy both SIND-aCPA1 security (due to IND-CPA security of $\widetilde{\mathsf{PKE}}$) and STE-xCPA1 security (due to ciphertext pseudorandomness of $\widetilde{\mathsf{SE}}$). However, it does not satisfy SIND-aCPA3 security, since given dk, one can easily decrypt the partial ciphertext to obtain $\hat{\mathsf{m}}_b$, and hence distinguish the experiments $\text{SIND-aCPA3}_{\Pi,\Sigma,b}(\mathcal{A})$ for $b \in \{0, 1\}$ with advantage 1. □

Theorem 4 (SIND-aCPA2 $\wedge$ STE-xCPA1 $\not\Rightarrow$ SIND-aCPA3). *For an AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$*, SIND-aCPA2 security and STE-xCPA1 security together do not imply SIND-aCPA3 security.*

Proof. Let $\widetilde{\mathsf{AME}}$ be an AME scheme with STE-xCPA1 security and SIND-aCPA2 security, and $\widetilde{\mathsf{SE}} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a PR-CT-secure symmetric encryption scheme. We construct an AME scheme $\mathsf{AME} = (\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}), \Sigma = (\mathsf{aKGen}, \mathsf{aEnc}, \mathsf{aDec}))$ as follows.

- The key generation algorithm returns $(\mathsf{pk}, \mathsf{sk}) \leftarrow \widetilde{\mathsf{AME}}.\mathsf{KGen}(1^\lambda)$.
- The anamorphic key generation algorithm returns $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk} = (\mathsf{dk}_1, \mathsf{dk}_2))$, where $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}_1) \leftarrow \widetilde{\mathsf{AME}}.\mathsf{aKGen}(1^\lambda)$, and $\mathsf{dk}_2 \leftarrow \widetilde{\mathsf{SE}}.\mathsf{KGen}(1^\lambda)$.
- The encryption algorithm returns $\mathsf{ct} = (\widetilde{\mathsf{AME}}.\mathsf{Enc}(\mathsf{pk}, \mathsf{m})||str)$, where str is a string randomly sampled from the ciphertext space of $\widetilde{\mathsf{SE}}$.
- The anamorphic encryption algorithm returns $\mathsf{ct} = (\widetilde{\mathsf{AME}}.\mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}_1, \mathsf{m}, \hat{\mathsf{m}})||\widetilde{\mathsf{SE}}.\mathsf{Enc}(\mathsf{dk}_2, \mathsf{m}))$.

It is easy to see that AME satisfies both SIND-aCPA2 security (due to SIND-aCPA2 security of $\widetilde{\mathsf{AME}}$ and independence of $\widetilde{\mathsf{SE}}.\mathsf{Enc}(\mathsf{dk}_2, \mathsf{m})$) and STE-xCPA1 security (due to ciphertext pseudorandomness of $\widetilde{\mathsf{SE}}$ and STE-xCPA1 security of $\widetilde{\mathsf{AME}}$). However, it does not satisfy SIND-aCPA3 security, since given dk, one can easily decrypt the symmetric ciphertext part to obtain the normal message m, and hence distinguish the experiments $\text{SIND-aCPA3}_{\Pi,\Sigma,b}(\mathcal{A})$ for $b \in \{0, 1\}$ with advantage 1. □

Relations between SIND-nCPA and SIND-aCPA3. The following results highlight the incomparability between notions when assuming STE-xCPA1.

Theorem 5 (SIND-nCPA $\wedge$ STE-xCPA1 $\not\Rightarrow$ SIND-aCPA3). *For an STE-xCPA1-secure AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$*, SIND-nCPA security does not imply SIND-aCPA3 security.*

[9] Note that the ciphertext space of SE could be infinite depending on the length of the anamorphic message to be encrypted. Here we basically assume a fixed anamorphic message space, and sample the same length each time.

Proof. Let $\widetilde{\mathsf{PKE}}$ be an IND-CPA-secure PKE scheme, and $\widetilde{\mathsf{SE}}$ be a PR-CT-secure symmetric encryption scheme (cf. Def. 6). We give an AME construction $\mathsf{AME} = (\Pi, \Sigma)$ which satisfies both STE-xCPA1 and SIND-nCPA but not SIND-aCPA3.

- The key generation algorithm returns $(\mathsf{pk}, \mathsf{sk}) \leftarrow \widetilde{\mathsf{PKE}}.\mathsf{KGen}(1^\lambda)$.
- The anamorphic key generation algorithm returns $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk})$, where $(\mathsf{pk}, \mathsf{sk}) \leftarrow \widetilde{\mathsf{PKE}}.\mathsf{KGen}(1^\lambda)$, and $\mathsf{dk} \leftarrow \widetilde{\mathsf{SE}}.\mathsf{KGen}(1^\lambda)$.
- The encryption algorithm returns

$$\mathsf{ct} = (\widetilde{\mathsf{PKE}}.\mathsf{Enc}(\mathsf{pk}, \mathsf{m})||str_1||str_2),$$

  where str_1 and str_2 are two strings randomly sampled from the ciphertext space of $\widetilde{\mathsf{SE}}$.
- The anamorphic encryption algorithm returns

$$\mathsf{ct} = (\widetilde{\mathsf{PKE}}.\mathsf{Enc}(\mathsf{pk}, \mathsf{m})||\widetilde{\mathsf{SE}}.\mathsf{Enc}(\mathsf{dk}, \mathsf{m})||\widetilde{\mathsf{SE}}.\mathsf{Enc}(\mathsf{dk}, \hat{\mathsf{m}})).$$

Now, STE-xCPA1 of AME is ensured by the PR-CT security of $\widetilde{\mathsf{SE}}$, and SIND-nCPA is ensured by the IND-CPA security of $\widetilde{\mathsf{PKE}}$ and the independence of $(\mathsf{pk}, \mathsf{sk})$ and dk. However, it is easy to see that AME does not satisfy SIND-aCPA3 at all, since dk gives full access to read the normal message m in an anamorphic ciphertext. □

Theorem 6 (SIND-aCPA3 ∧ STE-xCPA1 ⇏ SIND-nCPA). *For an STE-xCPA1-secure AME scheme $\mathsf{AME} = (\Pi, \Sigma)$, SIND-aCPA3 security does not imply SIND-nCPA security.*

Proof. Let $\widetilde{\mathsf{PKE}}$ be an IND-CPA-secure PKE scheme, and $\widetilde{\mathsf{SE}}$ be a PR-CT-secure symmetric encryption scheme (cf. Def. 6). We give an AME construction $\mathsf{AME} = (\Pi, \Sigma)$ which satisfies both STE-xCPA1 and SIND-aCPA3 but not SIND-nCPA as follows.

- The key generation algorithm returns $\mathsf{pk} = (\mathsf{pk}_0, \mathsf{pk}_1)$ and $\mathsf{sk} = \mathsf{sk}_0$, where $(\mathsf{pk}_b, \mathsf{sk}_b) \leftarrow \widetilde{\mathsf{PKE}}.\mathsf{KGen}(1^\lambda)$ for $b \in \{0, 1\}$.
- The anamorphic key generation algorithm returns $\mathsf{pk} = (\mathsf{pk}_0, \mathsf{pk}_1)$, $\mathsf{sk} = \mathsf{sk}_0$, and $\mathsf{dk} = (\mathsf{sk}_1, \mathsf{k})$, where $(\mathsf{pk}_b, \mathsf{sk}_b) \leftarrow \widetilde{\mathsf{PKE}}.\mathsf{KGen}(1^\lambda)$ for $b \in \{0, 1\}$, and $\mathsf{k} \leftarrow \widetilde{\mathsf{SE}}.\mathsf{KGen}(1^\lambda)$.
- The encryption algorithm returns

$$\mathsf{ct} = (\mathsf{c}_0||\mathsf{c}_1||\mathsf{c}_2) = (\widetilde{\mathsf{PKE}}.\mathsf{Enc}(\mathsf{pk}_0, \mathsf{m})||\widetilde{\mathsf{PKE}}.\mathsf{Enc}(\mathsf{pk}_1, \mathsf{m})||\widetilde{\mathsf{PKE}}.\mathsf{Enc}(\mathsf{pk}_0, r)),$$

  where r is a random string sampled from the message space of $\widetilde{\mathsf{PKE}}$.
- The anamorphic encryption algorithm returns

$$\mathsf{ct} = (\mathsf{c}_0||\mathsf{c}_1||\mathsf{c}_2) = (\widetilde{\mathsf{PKE}}.\mathsf{Enc}(\mathsf{pk}_0, \mathsf{m})||\widetilde{\mathsf{PKE}}.\mathsf{Enc}(\mathsf{pk}_1, \mathbf{0})||\widetilde{\mathsf{PKE}}.\mathsf{Enc}(\mathsf{pk}_0, r)),$$

  where $\mathbf{0}$ denotes an all-zero string, and $r \leftarrow \widetilde{\mathsf{SE}}.\mathsf{Enc}(\mathsf{k}, \hat{\mathsf{m}})$.

Now, STE-xCPA1 of AME is ensured by the ciphertext pseudorandomness (PR-CT) of $\widetilde{\mathsf{SE}}$ and the IND-CPA security of $\widetilde{\mathsf{PKE}}$ under sk_1, and SIND-aCPA3 is ensured by the IND-CPA security of $\widetilde{\mathsf{PKE}}$ under sk_0. However, it is easy to see that AME does not satisfy SIND-nCPA at all, since $\mathsf{dk} = (\mathsf{sk}_1, \mathsf{k})$ gives full access to read the normal message m from c_1. □

An observation is that, if $(\mathsf{pk}, \mathsf{sk})$ and dk are independent,[10] then we can prove SIND-aCPA3 and STE-xCPA1 together imply SIND-nCPA as follows.

[10] Note that this is not the case of anamorphic extension defined in [BGH+24], where the generation of dk may still depend on $(\mathsf{pk}, \mathsf{sk})$.

$$
\begin{aligned}
&(\mathsf{pk}, \mathsf{dk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_0)) \\
\approx &(\mathsf{pk}, \mathsf{dk}', \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_0)) && \text{// independence of } (\mathsf{pk}, \mathsf{sk}) \text{ and } \mathsf{dk} \\
\approx &(\mathsf{pk}, \mathsf{dk}', \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}_0, \hat{\mathsf{m}})) && \text{// STE-xCPA1} \\
\approx &(\mathsf{pk}, \mathsf{dk}', \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}_1, \hat{\mathsf{m}})) && \text{// SIND-aCPA3} \\
\approx &(\mathsf{pk}, \mathsf{dk}', \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_1)) && \text{// STE-xCPA1} \\
\approx &(\mathsf{pk}, \mathsf{dk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_1)) && \text{// independence of } (\mathsf{pk}, \mathsf{sk}) \text{ and } \mathsf{dk}
\end{aligned}
$$

Here $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{aKGen}(1^\lambda)$, and dk' is an independent double key sampled according to aKGen. Note that the other direction does not hold, i.e., even if $(\mathsf{pk}, \mathsf{sk})$ and dk are independent, we still have SIND-nCPA + STE-xCPA1 $\not\Rightarrow$ SIND-aCPA3, due to the above example in the proof of Theorem 5.

Remark 6. Given the result holds in the CPA setting and the counterexample in the proof of Theorem 6, it seems highly likely that in the CCA setting, SIND-aCCA3 (Def. 31) and STE-xCCA1 (Def. 23) together do not imply SIND-nCCA (Def. 28). However, we were unable to construct a counterexample to show this. Intuitively, an adversary (whether holding sk or dk) might reuse (part of) a challenge ciphertext in decryption oracles calls, which makes showing both SIND-aCCA3 and STE-xCCA1 hold more difficult without inadvertently also achieving SIND-nCCA security. We leave a formal proof or refutation of our conjecture as an interesting open problem.

Theorem 7 (SIND-nCPA ($\wedge$ STE-xCPA1) $\not\Leftrightarrow$ SIND-aCPA2 ($\wedge$ STE-xCPA1)). *For an STE-xCPA1-secure AME scheme $\mathsf{AME} = (\Pi, \Sigma)$, SIND-nCPA security does not imply SIND-aCPA2 security, and vice versa.*

Proof. One direction, SIND-nCPA $\wedge$ STE-xCPA1 $\not\Rightarrow$ SIND-aCPA2, is established through a counterexample in the proof of Theorem 3, where an anamorphic ciphertext is in the form of

$$(\mathsf{PKE.Enc}(\mathsf{pk}, \mathsf{m}) || \mathsf{SE.Enc}(\mathsf{dk}, \hat{\mathsf{m}})),$$

and dk allows full access to the anamorphic message but does not compromise the security of m.

The other direction, SIND-nCPA $\not\Leftarrow$ SIND-aCPA2 $\wedge$ STE-xCPA1, is proved through a counterexample where a normal ciphertext is in the form of $(\mathsf{m}||str)$, and an anamorphic ciphertext is in the form of

$$(\mathsf{m} || \mathsf{SE.Enc}(\mathsf{dk}, \mathsf{PKE.Enc}(\mathsf{pk}, \hat{\mathsf{m}}))).$$

In this case, STE-xCPA1 is inherited from the PR-CT security of SE, and SIND-aCPA2 is guaranteed by the IND-CPA security of PKE. However, SIND-nCPA is not achieved since the normal message m is exposed in plain within the ciphertext. □

Relations with (standard) IND-CPA. We show that standard IND-CPA security is implied by many of our notions.

Theorem 8 (SIND-nCPA $\wedge$ STE-xCPA1 $\Rightarrow$ IND-CPA, and SIND-nCPA $\not\Leftarrow$ IND-CPA $\wedge$ STE-xCPA1). *For an AME scheme $\mathsf{AME} = (\Pi, \Sigma)$, SIND-nCPA security and STE-xCPA1 security together imply IND-CPA security. However, IND-CPA security, even combined with STE-xCPA1 security, does not imply SIND-nCPA security.*

Proof. We first prove SIND-nCPA $\wedge$ STE-xCPA1 $\Rightarrow$ IND-CPA. By STE-xCPA1 security, $\text{SIND-nCPA}_{\Pi,\Sigma,b}(\mathcal{A})$ and $\text{IND-CPA}_{\Pi,b}(\mathcal{A})$ are indistinguishable to the adversary $\mathcal{A}$ who holds pk only (see the proof of Theorem 1). Therefore, the indistinguishability between $\text{SIND-nCPA}_{\Pi,\Sigma,0}(\mathcal{A})$ and $\text{SIND-nCPA}_{\Pi,\Sigma,1}(\mathcal{A})$ directly implies the indistinguishability between $\text{IND-CPA}_{\Pi,0}(\mathcal{A})$ and $\text{IND-CPA}_{\Pi,1}(\mathcal{A})$, which is exactly the definition of IND-CPA security.

The proof for the other direction SIND-nCPA $\not\Leftarrow$ IND-CPA $\wedge$ STE-xCPA1 is straightforward, since both IND-CPA security and STE-xCPA1 security assume that the adversary is not given the double key dk. If

an AME scheme (Π, Σ) is designed such that sk is contained in dk then the normal message m is revealed immediately,[11] and hence SIND-nCPA security does not hold. □

Theorem 9 (SIND-aCPA1 ∧ STE-xCPA1 ⇒ IND-CPA, and SIND-aCPA1 ⇏ IND-CPA ∧ STE-xCPA1). *For an AME scheme $\mathsf{AME} = (\Pi, \Sigma)$, SIND-aCPA1 security and STE-xCPA1 security together imply IND-CPA security. However, IND-CPA security, even combined with STE-xCPA1 security, does not imply SIND-aCPA1 security.*

Proof. The proof for SIND-aCPA1 ∧ STE-xCPA1 ⇒ IND-CPA is similar to the proof of Theorem 8. Namely, by STE-xCPA1 security, $\text{SIND-aCPA1}_{\Pi,\Sigma,b}(\mathcal{A})$ and $\text{IND-CPA}_{\Pi,b}(\mathcal{A})$ are indistinguishable to the adversary $\mathcal{A}$ holding pk only. Therefore, IND-CPA security directly follows from SIND-aCPA1 security.

The proof of the opposite direction SIND-aCPA1 ⇏ IND-CPA ∧ STE-xCPA1 is illustrated by a counterexample of an AME scheme where sk is a part of dk. Therefore, holding dk enables both decryption and anamorphic decryption, and hence SIND-aCPA1 does not hold at all. However, IND-CPA security and STE-xCPA1 security may still hold. □

Corollary 1. *For an AME scheme $\mathsf{AME} = (\Pi, \Sigma)$, SIND-aCPA3 security and STE-xCPA1 security together imply IND-CPA security. However, IND-CPA security, even combined with STE-xCPA1 security, does not imply SIND-aCPA3 security.*

Theorem 10 (IND-CPA (∧ STE-xCPA1) ⇎ SIND-aCPA2 (∧ STE-xCPA1)). *For an STE-xCPA1-secure AME scheme $\mathsf{AME} = (\Pi, \Sigma)$, IND-CPA security does not imply SIND-aCPA2 security, and vice versa.*

The proof is similar to the proof of Theorem 7 and we omit it here.

Relations Conditioned on SIND-xCPA1. We now explore the relations conditioned on the SIND-xCPA1 security.

Theorem 11. *For an SIND-xCPA1-secure AME scheme $\mathsf{AME} = (\Pi, \Sigma)$, the following four security notions are equivalent:*

1. *SIND-nCPA (Def. 16, given dk, to distinguish $\mathsf{Enc}(\mathsf{m}_0)$ from $\mathsf{Enc}(\mathsf{m}_1)$).*
2. *SIND-xCPA2 (Def. 15, given dk, to distinguish $\mathsf{Enc}(\mathsf{m}_0)$ from $\mathsf{aEnc}(\mathsf{m}_1, \hat{\mathsf{m}})$).*
3. *SIND-aCPA1 (Def. 17, given dk, to distinguish $\mathsf{aEnc}(\mathsf{m}_0, \hat{\mathsf{m}})$ from $\mathsf{aEnc}(\mathsf{m}_1, \hat{\mathsf{m}})$).*
4. *SIND-aCPA3 (Def. 19, given dk, to distinguish $\mathsf{aEnc}(\mathsf{m}_0, \hat{\mathsf{m}}_0)$ from $\mathsf{aEnc}(\mathsf{m}_1, \hat{\mathsf{m}}_1)$).*

Proof. For simplicity, in the proof we omit the input pk (and dk) in $\mathsf{Enc}(\mathsf{pk}, \cdot)$ and $\mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \cdot, \cdot)$, but focus only on the input of (anamorphic) messages. Recall that in all the above four security definitions, anamorphic key generation is invoked and dk is given to the adversary, and the four notions differ on the challenge oracles only. Conditioned on SIND-xCPA1 (i.e., $\mathsf{Enc}(\mathsf{m}) \approx \mathsf{aEnc}(\mathsf{m}, \hat{\mathsf{m}})$), the equivalence among them are obtained due to the following.

$$
\begin{array}{ccc}
\mathsf{aEnc}(\mathsf{m}_0, \hat{\mathsf{m}}) & \approx & \mathsf{aEnc}(\mathsf{m}_1, \hat{\mathsf{m}}) \quad \text{//SIND-aCPA1} \\
\wr\!\wr & & \| \\
\mathsf{Enc}(\mathsf{m}_0) & \approx & \mathsf{aEnc}(\mathsf{m}_1, \hat{\mathsf{m}}) \quad \text{//SIND-xCPA2} \\
\| & & \wr\!\wr \\
\mathsf{Enc}(\mathsf{m}_0) & \approx & \mathsf{Enc}(\mathsf{m}_1) \quad \text{//SIND-nCPA} \\
\wr\!\wr & & \wr\!\wr \\
\mathsf{aEnc}(\mathsf{m}_0, \hat{\mathsf{m}}_0) & \approx & \mathsf{aEnc}(\mathsf{m}_1, \hat{\mathsf{m}}_1), \quad \text{//SIND-aCPA3}
\end{array}
$$

where the indistinguishability in each row defines a specific security notion, and all "$\wr\!\wr$" are due to the SIND-xCPA1 security. This concludes the proof. □

[11] A concrete example is the ElGamal anamorphic triplet shown in [KPP+23b], where the double key consists of a symmetric key K and the secret key x of the public key $X = g^x$.

However, SIND-aCPA2 (Def. 18) is strictly weaker than SIND-aCPA3 (equivalent to SIND-nCPA, SIND-xCPA2, and SIND-aCPA1) conditioned on SIND-xCPA1.

Theorem 12 (SIND-aCPA2 ($\wedge$ SIND-xCPA1) $\not\Rightarrow$ SIND-aCPA3). *For an SIND-xCPA1-secure AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$*, SIND-aCPA2 does not imply SIND-aCPA3.*

Proof. Let $\widetilde{\mathsf{PKE}}$ be an IND-CPA-secure PKE scheme and $\widetilde{\mathsf{AME}}$ be a SIND-xCPA1-secure and SIND-aCPA3-secure AME scheme. We now construct another AME scheme AME as follows.

- The key generation algorithm returns $(\mathsf{pk}, \mathsf{sk}) := ((\mathsf{pk}_1, \mathsf{pk}_2), \mathsf{sk}_1)$, where $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \widetilde{\mathsf{AME}}.\mathsf{KGen}(1^\lambda)$ and $(\mathsf{pk}_2, \mathsf{sk}_2) \leftarrow \widetilde{\mathsf{PKE}}.\mathsf{KGen}(1^\lambda)$.
- The anamorphic key generation algorithm returns $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) := ((\mathsf{pk}_1, \mathsf{pk}_2), \mathsf{sk}_1, (\mathsf{dk}_1, \mathsf{sk}_2))$, where $(\mathsf{pk}_1, \mathsf{sk}_1, \mathsf{dk}_1) \leftarrow \widetilde{\mathsf{AME}}.\mathsf{aKGen}(1^\lambda)$ and $(\mathsf{pk}_2, \mathsf{sk}_2) \leftarrow \widetilde{\mathsf{PKE}}.\mathsf{KGen}(1^\lambda)$.
- The encryption algorithm returns

$$\mathsf{ct} = (\widetilde{\mathsf{AME}}.\mathsf{Enc}(\mathsf{pk}_1, \mathsf{m}) || \widetilde{\mathsf{PKE}}.\mathsf{Enc}(\mathsf{pk}_2, \mathsf{m})).$$

- The anamorphic encryption algorithm returns

$$\mathsf{ct} = (\widetilde{\mathsf{AME}}.\mathsf{aEnc}(\mathsf{pk}_1, \mathsf{dk}_1, \mathsf{m}, \hat{\mathsf{m}}) || \widetilde{\mathsf{PKE}}.\mathsf{Enc}(\mathsf{pk}_2, \mathsf{m})).$$

It is easy to see that SIND-xCPA1 security and SIND-aCPA2 security inherit from those of $\widetilde{\mathsf{AME}}$ and the independence of $(\mathsf{pk}_1, \mathsf{sk}_1, \mathsf{dk}_1)$ and $(\mathsf{pk}_2, \mathsf{sk}_2)$. However, given $\mathsf{dk} = (\mathsf{dk}_1, \mathsf{sk}_2)$, the adversary can do normal decryption on the second part of ct and hence SIND-aCPA3 does not hold. □

Remark 7 (On Obtaining The Strongest CPA Security). In the CPA setting, an AME scheme must satisfy STE-xCPA1, SIND-xCPA1, and SIND-nCPA to achieve the strongest security. As shown in Table 3, several existing AME constructions already achieve these guarantees. Note that, in AME schemes based on hybrid encryption [CGM24a] and IBE-PKE [CGM24a], the double key dk is a symmetric encryption key used to encrypt $\hat{\mathsf{m}}$ and derive a key k, which is subsequently used in another symmetric encryption or MAC scheme. To prove SIND-xCPA security, one must argue that k remains pseudorandom even to an adversary holding dk, which necessitates idealized primitives; see Section 5 for additional details, including the more complicated CCA setting.

4 Security of Anamorphic Encryption in the CCA Setting

In this section, we investigate the security of AME in the chosen-ciphertext attack (CCA) setting, including its relationship with robustness [BGH+24]. Our analysis follows the comprehensive approach developed for the CPA setting (cf. Fig. 7 in Section 3.1), and extends it by granting the adversary the access to additional decryption oracles—namely the standard decryption oracle Dec, or the anamorphic decryption oracle aDec, or both—depending on the scenarios.

4.1 Robustness

Since CCA security is highly related with robustness (cf. Def. 20), we start by discussing more on it.

Definition 20 (ROB-PT Security of AME [BGH+24]). *An AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$ *is* ROB-PT*-secure, if, for every PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{rob-pt}}_{\Pi,\Sigma}(\mathcal{A}) := \Pr[\text{ROB-PT}_{\Pi,\Sigma}(\mathcal{A}) = 1]$$

is negligible in λ*, where* $\text{ROB-PT}_{\Pi,\Sigma}(\mathcal{A})$ *is defined in Fig. 17.*

$\text{ROB-PT}_{\Pi,\Sigma}(\mathcal{A})$:	
1 :	$(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{aKGen}(1^\lambda)$
2 :	$\mathsf{m} \leftarrow \mathcal{A}(\mathsf{pk}, \mathsf{sk})$
3 :	**if** $\mathsf{aDec}(\mathsf{sk}, \mathsf{dk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m})) \neq \perp$:
4 :	**return** 1
5 :	**return** 0

$\text{ROB-PT}_{\Pi,\Sigma}(\mathcal{A})$:	
1 :	$(\mathsf{pk}, \mathsf{sk}, \mathsf{sdk}, \mathsf{rdk}) \leftarrow \mathsf{aKGen}(1^\lambda)$
2 :	$\mathsf{m} \leftarrow \mathcal{A}(\mathsf{pk}, \mathsf{sk})$
3 :	**if** $\mathsf{aDec}(\mathsf{sk}, \mathsf{rdk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m})) \neq \perp$:
4 :	**return** 1
5 :	**return** 0

Fig. 17. ROB-PT security game for a symmetric (left) or an asymmetric (right) AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

In this work we define robustness in the single-challenge setting, differing from the multiple challenge setting by Banfi et al. [BGH+24], where the adversary $\mathcal{A}$ has up to a polynomial bound of times to sample message m, and $\mathcal{A}$ wins the security game as long as in one challenge query $\mathsf{aDec}(\mathsf{sk}, \mathsf{dk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m})) \neq \perp$. Via standard hybrid argument, robustness in the single-challenge setting implies the robustness in the multiple challenge setting with a loss factor Q, the number of queries made by $\mathcal{A}$.

Remark 8 (Two (Stronger) Variants of ROB-PT). We could further define two stronger variants of ROB-PT reasonably as follows.

- In the first variant, which we refer to as ROB-PT-R, the adversary is additionally allowed to choose the randomness for Enc. This notion applies in the scenarios where the adversary can influence the randomness used in the encryption process. However, it remains weaker than the ROB-CT notion we define below.
- In the second variant, referred to as SIND-ROB-PT, the adversary additionally has access to the double key dk. This notion applies when the adversary (acting as the anamorphic sender) aims to make the dictator aware of the use of anamorphic channel by prompting the dictator to send a specific message to the receiver, thereby framing the receiver. However, the anamorphic sender avoids submitting the double key to the dictator directly, as doing so would expose them to potential punishment. Note that this notion appears to be incomparable with the ROB-CT notion defined below.

The original definition of robustness by [BGH+24] considered only standard ciphertexts generated via normal encryption process. However, beyond these, there exist numerous "non-standard" ciphertexts, such as those produced through improper randomness or entirely adversarial ciphertexts crafted by attackers. To address this, we extend the robustness notion to allow the adversary to select ciphertexts itself and formalize the enhanced notion as ROB-CT, defined below.

Definition 21 (ROB-CT Security of AME). *An AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$ *is* ROB-CT*-secure, if for every PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{rob-ct}}_{\Pi,\Sigma}(\mathcal{A}) := \Pr[\text{ROB-CT}_{\Pi,\Sigma}(\mathcal{A}) = 1]$$

is negligible in λ*, where* $\text{ROB-CT}_{\Pi,\Sigma}(\mathcal{A})$ *is defined in Fig. 18.*

Now we consider a stronger notion ROB-CT+, where the adversary additionally has access to an anamorphic encryption oracle and can output any adversarial ciphertexts.[12]

Definition 22 (ROB-CT+ Security of AME). *An AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$ *is* ROB-CT+*-secure, if, for every PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{rob-ct+}}_{\Pi,\Sigma}(\mathcal{A}) := \Pr[\text{ROB-CT+}_{\Pi,\Sigma}(\mathcal{A}) = 1]$$

is negligible in λ*, where* $\text{ROB-CT+}_{\Pi,\Sigma}(\mathcal{A})$ *is defined in Fig. 19.*

[12] ROB-CT+ is defined analogously to the notion of strong anamorphic unforgeability in a concurrent work [DG25].

$\text{ROB-CT}_{\Pi,\Sigma}(\mathcal{A})$:

1 : $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{aKGen}(1^\lambda)$
2 : $\mathsf{ct}^* \leftarrow \mathcal{A}(\mathsf{pk}, \mathsf{sk})$
3 : **if** $\mathsf{aDec}(\mathsf{sk}, \mathsf{dk}, \mathsf{ct}^*) \neq \perp$:
4 : **return** 1
5 : **return** 0

$\text{ROB-CT}_{\Pi,\Sigma}(\mathcal{A})$:

1 : $(\mathsf{pk}, \mathsf{sk}, \mathsf{sdk}, \mathsf{rdk}) \leftarrow \mathsf{aKGen}(1^\lambda)$
2 : $\mathsf{ct}^* \leftarrow \mathcal{A}(\mathsf{pk}, \mathsf{sk})$
3 : **if** $\mathsf{aDec}(\mathsf{sk}, \mathsf{rdk}, \mathsf{ct}^*) \neq \perp$:
4 : **return** 1
5 : **return** 0

Fig. 18. ROB-CT security game for a symmetric (left) or an asymmetric (right) AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

$\text{ROB-CT+}_{\Pi,\Sigma}(\mathcal{A})$:

1 : $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{aKGen}(1^\lambda)$
2 : $\mathsf{ct}^* \leftarrow \mathcal{A}^{\text{AEnc}}(\mathsf{pk}, \mathsf{sk})$
3 : **if** $(\mathsf{aDec}(\mathsf{sk}, \mathsf{dk}, \mathsf{ct}^*) \neq \perp) \wedge (\mathsf{ct}^* \notin \mathsf{CT})$:
4 : **return** 1
5 : **return** 0

AEnc$(\mathsf{m}, \hat{\mathsf{m}})$

1 : $\mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}})$
2 : $\mathsf{CT} := \mathsf{CT} \cup \{\mathsf{ct}\}$
3 : **return** ct

$\text{ROB-CT+}_{\Pi,\Sigma}(\mathcal{A})$:

1 : $(\mathsf{pk}, \mathsf{sk}, \mathsf{sdk}, \mathsf{rdk}) \leftarrow \mathsf{aKGen}(1^\lambda)$
2 : $\mathsf{ct}^* \leftarrow \mathcal{A}^{\text{AEnc}}(\mathsf{pk}, \mathsf{sk})$
3 : **if** $(\mathsf{aDec}(\mathsf{sk}, \mathsf{rdk}, \mathsf{ct}^*) \neq \perp) \wedge (\mathsf{ct}^* \notin \mathsf{CT})$:
4 : **return** 1
5 : **return** 0

AEnc$(\mathsf{m}, \hat{\mathsf{m}})$

1 : $\mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{sdk}, \mathsf{m}, \hat{\mathsf{m}})$
2 : $\mathsf{CT} := \mathsf{CT} \cup \{\mathsf{ct}\}$
3 : **return** ct

Fig. 19. ROB-CT+ security games for a symmetric (left) or an asymmetric (right) AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

Theorem 13. *For an AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$*, ROB-CT+ implies ROB-CT, which in turn implies ROB-PT. Conversely, the implications in the opposite direction do not hold.*

Proof. As the forward implications are straightforward to see, we focus on the proof of the opposite direction.

The claim ROB-CT+ $\not\Leftarrow$ ROB-CT is proved via the following counterexample. Let AE be an authenticated encryption scheme, and PKE be a standard (IND-CPA-secure) PKE scheme. Now we construct an AME scheme whose anamorphic encryption on $(\mathsf{m}, \hat{\mathsf{m}})$ returns

$$\mathsf{ct} = (\mathsf{PKE.Enc}(\mathsf{pk}, \mathsf{m}) || \mathsf{AE.Enc}(\mathsf{dk}, \hat{\mathsf{m}})).$$

ROB-CT of the scheme is guaranteed by the security of AE. However, it does not satisfy ROB-CT+, since given a ciphertext $\mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}})$, one can easily replace the first part of ct to another encryption of m and hence obtain a valid anamorphic ciphertext.

Another concrete example occurs in the stateful AME scheme in [BGH+24] where in the anamorphic encryption, the randomness r used in encryption is replaced with $r := \mathsf{F}_{\mathsf{dk}}(ctr) \oplus \hat{\mathsf{m}}$. Here $\mathsf{F}_{\mathsf{dk}}(\cdot)$ is a secure pseudorandom function (PRF) with key dk, and ctr is a counter kept in state. The scheme satisfies ROB-CT, but not ROB-CT+ due to the malleability of r.

The claim ROB-CT $\not\Leftarrow$ ROB-PT is proven by the counterexample of the Koppula-Waters (KW) encryption scheme [KW19] shown in [PPY24]. In the anamorphic KW encryption, the double key dk (the receiver double key in [PPY24]) is not involved in the anamorphic encryption algorithm. That is, the sender, holding only the public key, can select whether encrypt a message via normal encryption, or encrypt a message as well as an anamorphic message via anamorphic encryption. ROB-PT holds as long as the anamorphic message $\hat{\mathsf{m}} \neq 0^\lambda$ (λ is some global default parameter), which is easily achieved if we just exclude 0^λ from the space of anamorphic messages. However, ROB-CT does not hold at all, since it is easy to generate an anamorphic ciphertext via anamorphic encryption. □

4.2 Security against the Dictator in the CCA Setting

Now we extend the standard STE-xCPA1 security of AME (cf. Def. 11) to the CCA setting, which we call STE-xCCA1 security. Informally, it states that given additionally an oracle which returns $\perp$ in the normal mode and returns the anamorphic decryption result in the anamorphic mode, the adversary (acting as the dictator) still cannot distinguish these two modes.

Definition 23 (STE-xCCA1 Security of AME). *An AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$ *is* STE-xCCA1*-secure, if, for every PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{ste-xcca1}}_{\Pi,\Sigma}(\mathcal{A}) := |\Pr[\text{STE-xCCA1}_{\Pi,\Sigma,1}(\mathcal{A}) = 1] - \Pr[\text{STE-xCCA1}_{\Pi,\Sigma,0}(\mathcal{A}) = 1]|$$

is negligible in λ*, where* $\text{STE-xCCA1}_{\Pi,\Sigma,b}(\mathcal{A})$ *are defined in Fig. 20.*

$\text{STE-xCCA1}_{\Pi,\Sigma,b}(\mathcal{A})$:

1 : **if** $b = 0$:
2 : $\quad(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$
3 : **else** :
4 : $\quad(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{aKGen}(1^\lambda)$
5 : **return** $\mathcal{A}^{\text{CHAL},\text{ADEC}}(\mathsf{pk}, \mathsf{sk})$

CHAL$(\mathsf{m}, \hat{\mathsf{m}})$

1 : **if** $b = 0$:
2 : $\quad\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$
3 : **else** :
4 : $\quad\mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}})$
5 : $\mathsf{CT} := \mathsf{CT} \cup \{\mathsf{ct}\}$
6 : **return** ct

ADEC(ct)

1 : **if** $b = 0 \vee \mathsf{ct} \in \mathsf{CT}$:
2 : $\quad$**return** $\perp$
3 : **else** :
4 : $\quad$**return** $\mathsf{aDec}(\mathsf{sk}, \mathsf{dk}, \mathsf{ct})$

$\text{STE-xCCA1}_{\Pi,\Sigma,b}(\mathcal{A})$:

1 : **if** $b = 0$:
2 : $\quad(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$
3 : **else** :
4 : $\quad(\mathsf{pk}, \mathsf{sk}, \mathsf{sdk}, \mathsf{rdk}) \leftarrow \mathsf{aKGen}(1^\lambda)$
5 : **return** $\mathcal{A}^{\text{CHAL},\text{ADEC}}(\mathsf{pk}, \mathsf{sk})$

CHAL$(\mathsf{m}, \hat{\mathsf{m}})$

1 : **if** $b = 0$:
2 : $\quad\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$
3 : **else** :
4 : $\quad\mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{sdk}, \mathsf{m}, \hat{\mathsf{m}})$
5 : $\mathsf{CT} := \mathsf{CT} \cup \{\mathsf{ct}\}$
6 : **return** ct

ADEC(ct)

1 : **if** $b = 0 \vee \mathsf{ct} \in \mathsf{CT}$:
2 : $\quad$**return** $\perp$
3 : **else** :
4 : $\quad$**return** $\mathsf{aDec}(\mathsf{sk}, \mathsf{rdk}, \mathsf{ct})$

Fig. 20. STE-xCCA1 security games for a symmetric (left) or an asymmetric (right) AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

We argue that STE-xCCA1 is strictly stronger than (all) robustness notions, including ROB-CT+ (Def. 22), ROB-CT (Def. 21), and ROB-PT (Def. 20).

Theorem 14. *For an AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$*, STE-xCCA1 implies ROB-CT+, which in turn implies ROB-CT, and finally ROB-PT. Conversely, the implications in the opposite direction do not hold.*

Proof. The implication STE-xCCA1 $\Rightarrow$ ROB-CT+ is straightforward: if the adversary can generate a fresh ciphertext ct^* that decrypts successfully via aDec, then it is able to distinguish between the normal mode and the anamorphic mode by querying ct^* to the decryption oracle, thereby breaking STE-xCCA1 security.

Given that ROB-CT+ is strictly stronger than ROB-CT and then ROB-PT (formally stated in Theorem 13), it suffices to prove STE-xCCA1 $\not\Leftarrow$ ROB-CT+.

Let $\widetilde{\mathsf{AME}}$ be an STE-xCCA1-secure AME scheme, and therefore ROB-CT+-secure. Now we design a new AME scheme AME whose (anamorphic) key generation algorithms are identical to those in $\widetilde{\mathsf{AME}}$, and

$$\mathsf{AME.Enc}(\mathsf{pk}, \mathsf{m}) := \widetilde{\mathsf{AME}}.\mathsf{Enc}(\mathsf{pk}, (0||\mathsf{m})),$$
$$\mathsf{AME.aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}}) := \widetilde{\mathsf{AME}}.\mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, (1||\mathsf{m}), \hat{\mathsf{m}}).$$

ROB-CT+ of AME is derived from ROB-CT+ of $\widetilde{\mathsf{AME}}$. However, AME does not satisfy STE-xCCA1 security (or even STE-xCPA1 security), since one can easily distinguish the normal mode from the anamorphic mode with the knowledge of sk. □

We then extend DIND-xCPA1 (Def. 12) and DIND-aCPA2 (Def. 13) to the CCA setting.

Definition 24 (DIND-xCCA1 Security of AME). *An AME scheme $\mathsf{AME} = (\Pi, \Sigma)$ is DIND-xCCA1-secure, if for every PPT adversary $\mathcal{A}$, the advantage*

$$\mathsf{Adv}^{\text{dind-xcca1}}_{\Pi,\Sigma}(\mathcal{A}) := |\Pr[\text{DIND-xCCA1}_{\Pi,\Sigma,1}(\mathcal{A}) = 1] - \Pr[\text{DIND-xCCA1}_{\Pi,\Sigma,0}(\mathcal{A}) = 1]|$$

is negligible in λ, where $\text{DIND-xCCA1}_{\Pi,\Sigma,b}(\mathcal{A})$ are defined in Fig. 21.

$\text{DIND-xCCA1}_{\Pi,\Sigma,b}(\mathcal{A})$:
1 : $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{aKGen}(1^\lambda)$
2 : **return** $\mathcal{A}^{\text{CHAL},\text{ADEC}}(\mathsf{pk}, \mathsf{sk})$

CHAL$(\mathsf{m}, \hat{\mathsf{m}})$
1 : **if** $b = 0$:
2 : $\quad \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$
3 : **else** :
4 : $\quad \mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}})$
5 : $\mathsf{CT} := \mathsf{CT} \cup \{\mathsf{ct}\}$
6 : **return** ct

ADEC(ct)
1 : **if** $\mathsf{ct} \in \mathsf{CT}$: **return** $\perp$
2 : **return** $\mathsf{aDec}(\mathsf{sk}, \mathsf{dk}, \mathsf{ct})$

$\text{DIND-xCCA1}_{\Pi,\Sigma,b}(\mathcal{A})$:
1 : $(\mathsf{pk}, \mathsf{sk}, \mathsf{sdk}, \mathsf{rdk}) \leftarrow \mathsf{aKGen}(1^\lambda)$
2 : **return** $\mathcal{A}^{\text{CHAL},\text{ADEC}}(\mathsf{pk}, \mathsf{sk})$

CHAL$(\mathsf{m}, \hat{\mathsf{m}})$
1 : **if** $b = 0$:
2 : $\quad \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$
3 : **else** :
4 : $\quad \mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{sdk}, \mathsf{m}, \hat{\mathsf{m}})$
5 : $\mathsf{CT} := \mathsf{CT} \cup \{\mathsf{ct}\}$
6 : **return** ct

ADEC(ct)
1 : **if** $\mathsf{ct} \in \mathsf{CT}$: **return** $\perp$
2 : **return** $\mathsf{aDec}(\mathsf{sk}, \mathsf{rdk}, \mathsf{ct})$

Fig. 21. DIND-xCCA1 security games for a symmetric (left) or an asymmetric (right) AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

Remark 9. In the above definition, the adversary has no access to the anamorphic encryption oracle, since this can be 'replaced' by the challenge oracle that either returns an anamorphic ciphertext (perfectly simulation) or a normal ciphertext (indistinguishable from the adversary).

Definition 25 (DIND-aCCA2 Security of AME). *An AME scheme $\mathsf{AME} = (\Pi, \Sigma)$ is DIND-aCCA2-secure, if for every PPT adversary $\mathcal{A}$, the advantage*

$$\mathsf{Adv}^{\text{dind-acca2}}_{\Pi,\Sigma}(\mathcal{A}) := |\Pr[\text{DIND-aCCA2}_{\Pi,\Sigma,1}(\mathcal{A}) = 1] - \Pr[\text{DIND-aCCA2}_{\Pi,\Sigma,0}(\mathcal{A}) = 1]|$$

is negligible in λ, where $\text{DIND-aCCA2}_{\Pi,\Sigma,b}(\mathcal{A})$ are defined in Fig. 22.

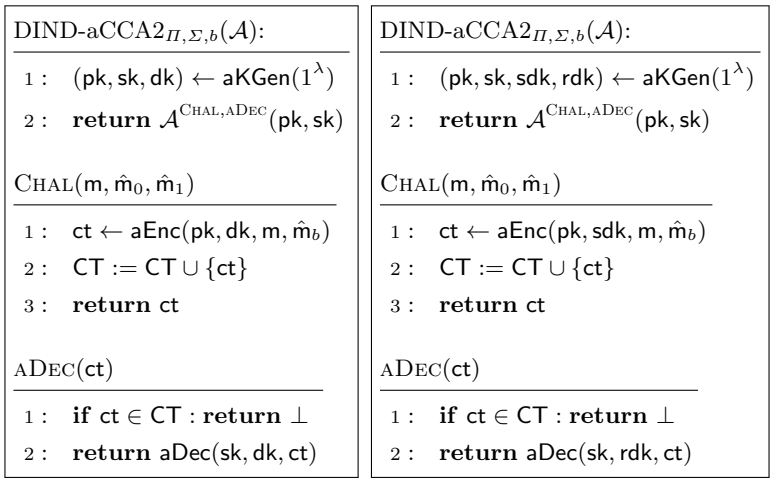


Fig. 22. DIND-aCCA2 security games for a symmetric (left) or an asymmetric (right) AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

Similar to that in the CPA setting, it is easy to get the following implications.

$$\text{STE-xCCA1} \implies \text{DIND-xCCA1} \implies \text{DIND-aCCA2}.$$

Theorem 15. *For an AME scheme $\mathsf{AME} = (\Pi, \Sigma)$, STE-xCCA1 implies DIND-xCCA1, and DIND-xCCA1 implies DIND-aCCA2.*

The proof is similar to the proof of Theorem 1 except that an anamorphic decryption oracle is additionally provided.

Obviously, STE-xCCA1 is strictly stronger than STE-xCPA1. On the other hand, we prove that equipped with ROB-CT+, STE-xCPA1 security can be enhanced to STE-xCCA1 security.

Theorem 16 (STE-xCPA1 $\wedge$ ROB-CT+ $\Rightarrow$ STE-xCCA1). *If an AME scheme (Π, Σ) has both STE-xCPA1 security and ROB-CT+ security, then it is STE-xCCA1-secure.*

Proof. It is straightforward to see that if $b = 0$, then the experiment STE-xCCA1$_{\Pi,\Sigma,0}(\mathcal{A})$ and the experiment STE-xCPA1$_{\Pi,\Sigma,0}(\mathcal{A})$ are identical, since the (anamorphic) decryption oracle in STE-xCCA1$_{\Pi,\Sigma,0}(\mathcal{A})$ always returns a failure indicator, downgrading the security from CCA to CPA. Therefore, to prove STE-xCCA1 security, it is sufficient to bound the distinguishing advantage between STE-xCCA1$_{\Pi,\Sigma,1}(\mathcal{A})$ and STE-xCPA1$_{\Pi,\Sigma,1}(\mathcal{A})$.

Suppose that the adversary $\mathcal{A}$ in STE-xCPA1$_{\Pi,\Sigma,1}(\mathcal{A})$ is also equipped with a decryption oracle which always returns $\perp$. Therefore, $\mathcal{A}$ is able to distinguish the experiment STE-xCPA1$_{\Pi,\Sigma,1}(\mathcal{A})$ from the experiment STE-xCCA1$_{\Pi,\Sigma,1}(\mathcal{A})$, as long as it can generate a ciphertext ct^*, which differs from all challenge ciphertexts it obtain from the challenge oracle and that $\mathsf{aDec}(\mathsf{sk},\mathsf{dk},\mathsf{ct}^*) \neq \perp$. According to the ROB-CT+ security of AME, this happens only with negligible probability, and hence the STE-xCCA1 security of AME holds. □

We then show by the following theorem that even though STE-xCCA1 implies ROB-CT+, the two weaker notions, DIND-xCCA1 and DIND-aCCA2 are incomparable with ROB-CT+.

Theorem 17. *For an AME scheme $\mathsf{AME} = (\Pi, \Sigma)$, DIND-xCCA1 security does not imply ROB-CT+, and ROB-CT+ does not imply DIND-aCCA2 (and hence DIND-xCCA1).*

Proof. We first prove DIND-xCCA1 $\not\Rightarrow$ ROB-CT+. The high-level idea is to let the public key contains a valid message-ciphertext pair for anamorphic encryption, hence failing the ROB-CT+ property. Let SE be an IND-CCA secure symmetric encryption scheme with dk the symmetric key (recall that DIND-xCCA1 requires only the indistinguishability of anamorphic ciphertexts and normal ciphertexts, but not the confidentiality of ciphertexts). We construct an AME scheme with anamorphic triplet Σ as follows.

- The anamorphic key generation returns $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk})$, where $\mathsf{dk} \leftarrow \mathsf{SE.KGen}(1^\lambda)$, $\mathsf{sk} := \bot$, and $\mathsf{pk} := ((\mathsf{m}_\pi, \hat{\mathsf{m}}_\pi), \mathsf{ct}_\pi)$ is a message-ciphertext pair with some random messages $(\mathsf{m}_\pi, \hat{\mathsf{m}}_\pi)$ and $\mathsf{ct}_\pi := (\tilde{\mathsf{ct}}, \mathsf{m}_\pi)$ with $\tilde{\mathsf{ct}} \leftarrow \mathsf{SE.Enc}(\mathsf{dk}, (\mathsf{m}_\pi, \hat{\mathsf{m}}_\pi))$.
- The encryption algorithm returns $\mathsf{ct} := (str||\mathsf{m})$, where str is a random string sampled from the ciphertext space of SE.
- The anamorphic encryption algorithm returns $\mathsf{ct} := (\mathsf{SE.Enc}(\mathsf{dk}, (\mathsf{m}, \hat{\mathsf{m}})), \mathsf{m})$. Note that the normal message m is exposed in plain, but it does not affect the DIND-xCCA1 security, which focuses only on the indistinguishability between $\mathsf{Enc}()$ and $\mathsf{aEnc}()$ (and the basic correctness).
- The decryption algorithm $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct} = (\tilde{\mathsf{ct}}, \mathsf{m}))$ returns m directly.
- The anamorphic decryption algorithm $\mathsf{aDec}(\mathsf{sk}, \mathsf{dk}, \mathsf{ct} = (\tilde{\mathsf{ct}}, \mathsf{m}))$ first takes the symmetric decryption to obtain $(\mathsf{m}', \hat{\mathsf{m}}) \leftarrow \mathsf{SE.Dec}(\mathsf{dk}, \tilde{\mathsf{ct}})$ (this step might return $\bot$ due to the CCA property of SE), and returns $\hat{\mathsf{m}}$ if $\mathsf{m}' = \mathsf{m}$ and $\bot$ otherwise.

It is straightforward to see that if SE has IND-CCA security, then Σ satisfies both correctness and DIND-xCCA1 security. However, given pk which contains a message-ciphertext pair from anamorphic encryption, one can easily break the ROB-CT+ security by just outputting ct_π.

Now we prove ROB-CT+ $\not\Rightarrow$ DIND-aCCA2. This direction is obvious since ROB-CT+ requires nothing about the confidentiality of either m or $\hat{\mathsf{m}}$. Therefore, any signature scheme with strong unforgeability can be transferred into an anamorphic triplet with ROB-CT+ security, where dk is the signing secret key, $\mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}})$ returns $\mathsf{ct} := (\mathsf{m}, \hat{\mathsf{m}}, \sigma)$ with σ a signature of $(\mathsf{m}, \hat{\mathsf{m}})$, and $\mathsf{aDec}(\mathsf{sk}, \mathsf{dk}, \mathsf{ct})$ returns $\hat{\mathsf{m}}$ if σ is a valid signature. ROB-CT+ is inherited from the strong unforgeability of the signature scheme. However, it is obvious that this scheme does not have DIND-aCCA2 security at all. (Even more, it is not a CPA secure encryption scheme!) □

4.3 Security against Anamorphic Senders in the CCA Setting

In this subsection, we investigate the security against the anamorphic sender in the CCA setting. Informally, the anamorphic sender—who holds the double key but not the secret key—is given access to two decryption oracles: a normal decryption oracle and an anamorphic decryption oracle. Security against the anamorphic sender requires that the confidentiality of the normal message, the anamorphic message, or potentially both, be preserved against such an adversary.

Definition 26 (SIND-xCCA1 Security of AME). *An AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$ *is* SIND-xCCA1*-secure, if for every PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{sind-xcca1}}_{\Pi,\Sigma}(\mathcal{A}) := |\Pr[\text{SIND-xCCA1}_{\Pi,\Sigma,1}(\mathcal{A}) = 1] - \Pr[\text{SIND-xCCA1}_{\Pi,\Sigma,0}(\mathcal{A}) = 1]|$$

is negligible in λ*, where* $\text{SIND-xCCA1}_{\Pi,\Sigma,b}(\mathcal{A})$ *are defined in Fig. 23.*

Definition 27 (SIND-xCCA2 Security of AME). *An AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$ *is* SIND-xCCA2*-secure, if for every PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{sind-xcca2}}_{\Pi,\Sigma}(\mathcal{A}) := |\Pr[\text{SIND-xCCA2}_{\Pi,\Sigma,1}(\mathcal{A}) = 1] - \Pr[\text{SIND-xCCA2}_{\Pi,\Sigma,0}(\mathcal{A}) = 1]|$$

is negligible in λ*, where* $\text{SIND-xCCA2}_{\Pi,\Sigma,b}(\mathcal{A})$ *are defined in Fig. 24.*

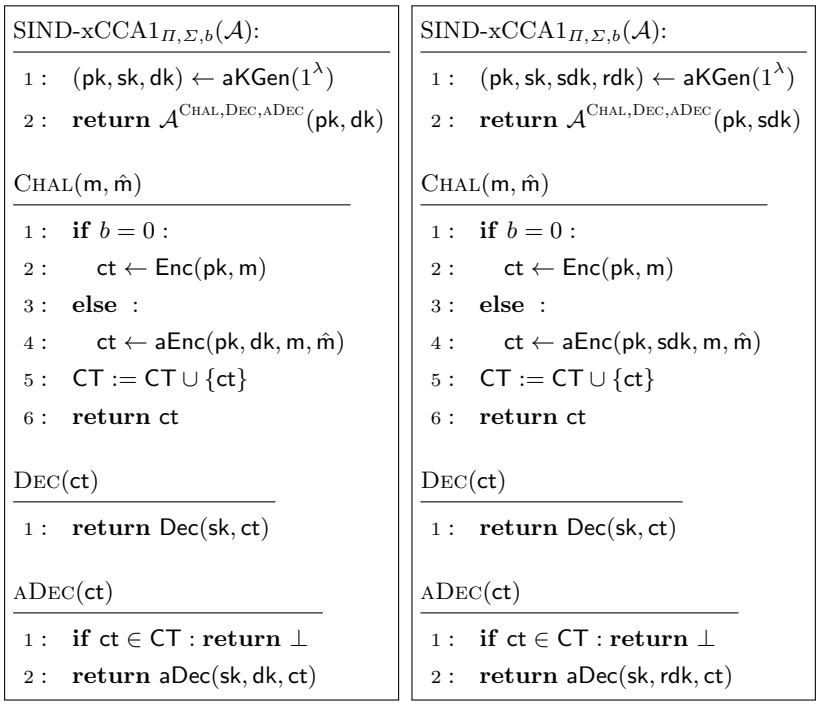


Fig. 23. SIND-xCCA1 security games for a symmetric (left) or an asymmetric (right) AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

Definition 28 (SIND-nCCA Security of AME). *An AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$ *is* SIND-nCCA*-secure, if for every PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{sind-ncca}}_{\Pi,\Sigma}(\mathcal{A}) := |\Pr[\text{SIND-nCCA}_{\Pi,\Sigma,1}(\mathcal{A}) = 1] - \Pr[\text{SIND-nCCA}_{\Pi,\Sigma,0}(\mathcal{A}) = 1]|$$

is negligible in λ*, where* $\text{SIND-nCCA}_{\Pi,\Sigma,b}(\mathcal{A})$ *are defined in Fig. 25.*

Definition 29 (SIND-aCCA1 Security of AME). *An AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$ *is* SIND-aCCA1*-secure, if for every PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{sind-acca1}}_{\Pi,\Sigma}(\mathcal{A}) := |\Pr[\text{SIND-aCCA1}_{\Pi,\Sigma,1}(\mathcal{A}) = 1] - \Pr[\text{SIND-aCCA1}_{\Pi,\Sigma,0}(\mathcal{A}) = 1]|$$

is negligible in λ*, where* $\text{SIND-aCCA1}_{\Pi,\Sigma,b}(\mathcal{A})$ *are defined in Fig. 26.*

Definition 30 (SIND-aCCA2 Security of AME). *An AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$ *is* SIND-aCCA2*-secure, if for every PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{sind-acca2}}_{\Pi,\Sigma}(\mathcal{A}) := |\Pr[\text{SIND-aCCA2}_{\Pi,\Sigma,1}(\mathcal{A}) = 1] - \Pr[\text{SIND-aCCA2}_{\Pi,\Sigma,0}(\mathcal{A}) = 1]|$$

is negligible in λ*, where* $\text{SIND-aCCA2}_{\Pi,\Sigma,b}(\mathcal{A})$ *are defined in Fig. 27.*

Definition 31 (SIND-aCCA3 Security of AME). *An AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$ *is* SIND-aCCA3*-secure, if for every PPT adversary* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}^{\text{sind-acca3}}_{\Pi,\Sigma}(\mathcal{A}) := |\Pr[\text{SIND-aCCA3}_{\Pi,\Sigma,1}(\mathcal{A}) = 1] - \Pr[\text{SIND-aCCA3}_{\Pi,\Sigma,0}(\mathcal{A}) = 1]|$$

is negligible in λ*, where* $\text{SIND-aCCA3}_{\Pi,\Sigma,b}(\mathcal{A})$ *are defined in Fig. 28.*

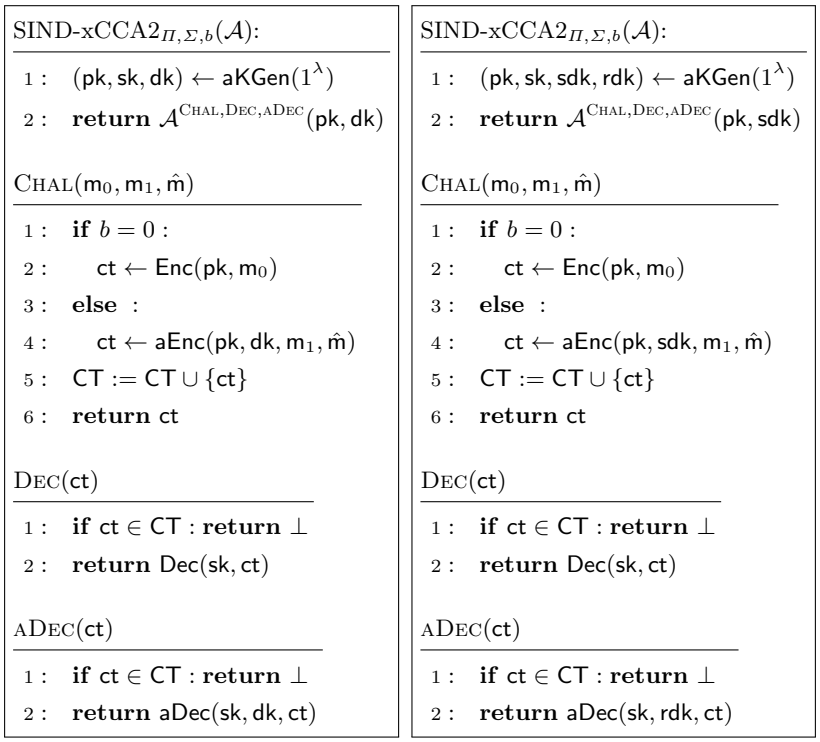


Fig. 24. SIND-xCCA2 security games for a symmetric (left) or an asymmetric (right) AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

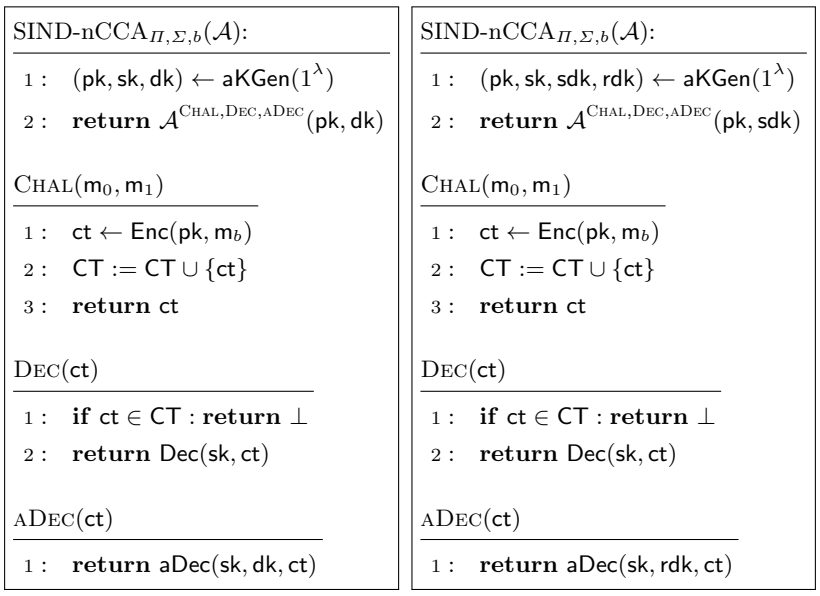


Fig. 25. SIND-nCCA security games for a symmetric (left) or an asymmetric (right) AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

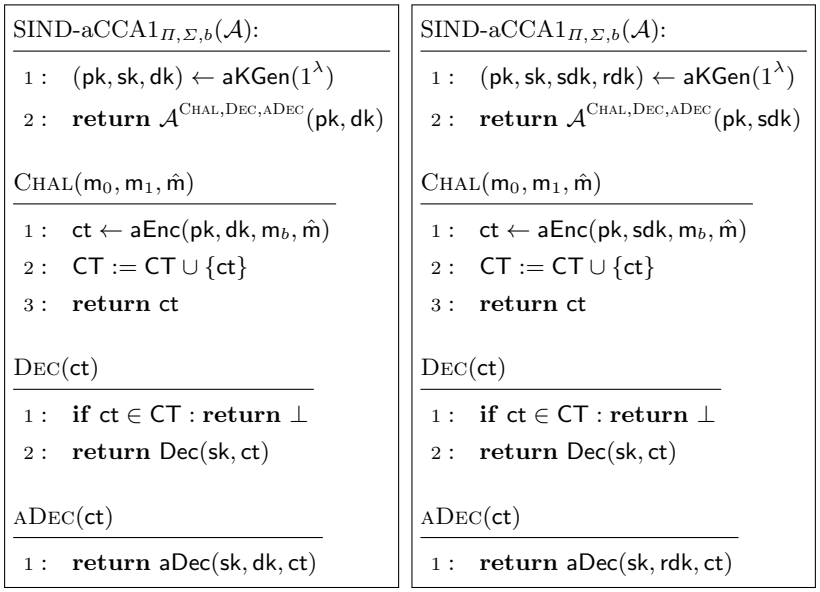


Fig. 26. SIND-aCCA1 security games for a symmetric (left) or an asymmetric (right) AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

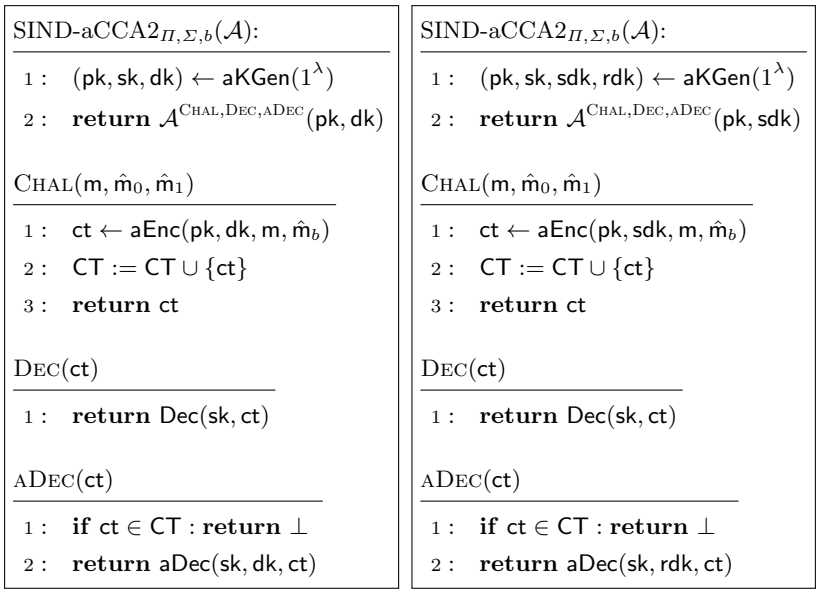


Fig. 27. SIND-aCCA2 security games for a symmetric (left) or an asymmetric (right) AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

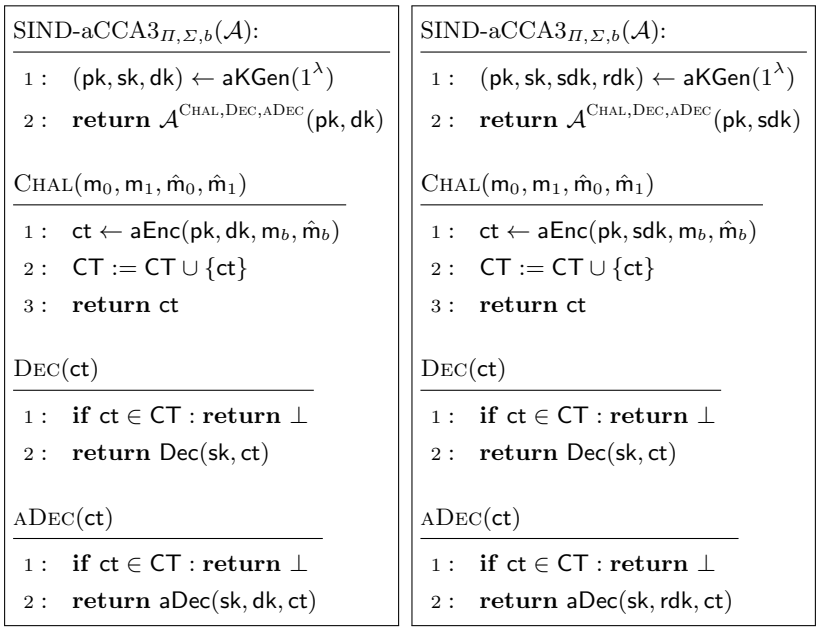


Fig. 28. SIND-aCCA3 security games for a symmetric (left) or an asymmetric (right) AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

As in the CPA setting, SIND-aCCA3 implies SIND-aCCA1 when $\hat{\mathsf{m}}_0 = \hat{\mathsf{m}}_1$, and SIND-aCCA2 when $\mathsf{m}_0 = \mathsf{m}_1$. The proof is straightforward, but we still emphasize that the anamorphic decryption oracle in SIND-aCCA3 (which is not allowed to query the challenge ciphertext) is stricter than that in SIND-aCCA1, and the decryption oracle (which is not allowed to query the challenge ciphertext) is stricter than that in SIND-aCCA2.

However, this is not an issue in the proof. Take the reduction from SIND-aCCA1 to SIND-aCCA3 as an example. In the reduction from switching a SIND-aCCA1 adversary $\mathcal{A}$ (where $\mathcal{A}$ samples $\mathsf{m}_0, \mathsf{m}_1, \hat{\mathsf{m}}$ to the challenge oracle) to a SIND-aCCA3 adversary $\mathcal{A}'$ (where $\mathcal{A}'$ samples $\mathsf{m}_0, \mathsf{m}_1, \hat{\mathsf{m}}_0, \hat{\mathsf{m}}_1$ to the challenge oracle), $\mathcal{A}'$ will sets $\hat{\mathsf{m}}_0 = \hat{\mathsf{m}}_1 = \hat{\mathsf{m}}$ and passes the challenge ciphertext obtained from its challenger to $\mathcal{A}$. Later whenever $\mathcal{A}$ asks an anamorphic decryption query of the challenge ciphertext ct^* (since this query is allowed in the SIND-aCCA1 experiment), $\mathcal{A}'$ directly returns $\hat{\mathsf{m}}$. Due to the correctness of AME, this is statistically close to the experiment where the result $\mathsf{aDec}(\mathsf{sk},\mathsf{dk},\mathsf{ct}^*)$ is returned. The argument for the reduction from SIND-aCCA2 to SIND-aCCA3 is analogous.

4.4 Relations among Notions in the CCA Setting

We now study the relationships among all CCA notions. Similar to the CPA setting, we first argue that SIND-aCCA(1/2/3) and DIND-aCCA2 are incomparable via the following theorem.

Theorem 18 (SIND-aCCA(1/2/3) $\nLeftrightarrow$ DIND-aCCA2). *For an AME scheme $\mathsf{AME} = (\Pi, \Sigma)$, SIND-aCCA(1/2/3) does not imply DIND-aCCA2, and vice versa.*

The proof is similar to the proof of Theorem 2 and we omit it here.

Relations among SIND-aCCA(1/2/3). The relations among SIND-aCCA(1/2/3) are analogous to those in the CPA setting.

Theorem 19 (SIND-aCCA1 $\wedge$ STE-xCCA1 $\nRightarrow$ SIND-aCCA3). *For an AME scheme $\mathsf{AME} = (\Pi, \Sigma)$, SIND-aCCA1 security and STE-xCCA1 security together do not imply SIND-aCCA3 security.*

Proof. Let $\widetilde{\mathsf{AME}}$ be an AME scheme satisfying SIND-aCCA1 and STE-xCCA1 (e.g., the randomness-recoverable CCA PKE with randomness $= \mathsf{AE.Enc}(\mathsf{dk}_1, \hat{\mathsf{m}})$), and $1_{\hat{\mathcal{M}}}$ be an anamorphic message belongs to the anamorphic message space of $\widetilde{\mathsf{AME}}$. We construct another AME scheme with anamorphic message space $\{0,1\}$ satisfying the first two properties but not SIND-aCCA3 as follows.

- The anamorphic key generation algorithm returns $\mathsf{pk}, \mathsf{sk}, \mathsf{dk} = (\mathsf{dk}_1, \mathsf{dk}_2)$, where $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}_1) \leftarrow \widetilde{\mathsf{AME}}.\mathsf{aKGen}(1^\lambda)$, and dk_2 is a randomly sampled key of a pseudorandom function (PRF) F.
- The encryption algorithm $\mathsf{Enc}(\mathsf{pk}, \mathsf{m})$ returns $\mathsf{ct} = \widetilde{\mathsf{AME}}.\mathsf{Enc}(\mathsf{pk}, \mathsf{m})$.
- The anamorphic encryption algorithm $\mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}})$ returns $\mathsf{ct} = \widetilde{\mathsf{AME}}.\mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}_1, \mathsf{m}, 1_{\hat{\mathcal{M}}})$ such that $\mathsf{F}_{\mathsf{dk}_2}(\mathsf{ct}) = \hat{\mathsf{m}}$ (note that $\hat{\mathsf{m}} \in \{0,1\}$).
- The anamorphic decryption algorithm aDec returns $\mathsf{F}_{\mathsf{dk}_2}(\mathsf{ct})$ if $\widetilde{\mathsf{AME}}.\mathsf{aDec}(\mathsf{sk}, \mathsf{dk}_1, \mathsf{ct}) = 1_{\hat{\mathcal{M}}}$, and $\perp$ otherwise.

The construction above inherits SIND-aCCA1 and STE-xCCA1 from $\widetilde{\mathsf{AME}}$. However, it does not satisfy SIND-aCCA3, since given $\mathsf{dk} = (\mathsf{dk}_1, \mathsf{dk}_2)$ and an anamorphic ciphertext $\mathsf{ct} = \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}})$, one can easily output the anamorphic message $\mathsf{F}_{\mathsf{dk}_2}(\mathsf{ct})$ (by correctness it must hold that $\widetilde{\mathsf{AME}}.\mathsf{aDec}(\mathsf{sk}, \mathsf{dk}_1, \mathsf{ct}) = 1_{\hat{\mathcal{M}}}$). □

Theorem 20 (SIND-aCCA2 $\wedge$ STE-xCCA1 $\not\Rightarrow$ SIND-aCCA3). *For an AME scheme $\mathsf{AME} = (\Pi, \Sigma)$, SIND-aCCA2 security and STE-xCCA1 security together do not imply SIND-aCCA3 security.*

The proof is similar to the proof above, except that the anamorphic encryption algorithm returns $\mathsf{ct} = \widetilde{\mathsf{AME}}.\mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}_1, \mathsf{m}, \hat{\mathsf{m}})$ such that $\mathsf{F}_{\mathsf{dk}_2}(\mathsf{ct}) = \mathsf{m}$ (similarly we assume the normal message space is $\{0,1\}$). We omit the details here.

Relations between SIND-nCCA and SIND-aCCA(1/3). Similar to the CPA setting, we prove the following theorem.

Theorem 21 (SIND-nCCA ($\wedge$ STE-xCCA1) $\not\Rightarrow$ SIND-aCCA(1/3)). *For an AME scheme $\mathsf{AME} = (\Pi, \Sigma)$, SIND-nCCA security and STE-xCCA1 security together do not imply either SIND-aCCA1 security, or SIND-aCCA3 security.*

Proof. Since SIND-aCCA3 is strictly stronger than SIND-aCCA1, it is sufficient to prove that SIND-nCCA ($\wedge$ STE-xCCA1) $\not\Rightarrow$ SIND-aCCA1. Let $\widetilde{\mathsf{AME}}$ be a SIND-nCCA-secure and STE-xCCA1 secure AME scheme. We construct an AME scheme AME with normal message space $\{0,1\}$ as follows.

- The key generation algorithm returns $(\mathsf{pk}, \mathsf{sk}) \leftarrow \widetilde{\mathsf{AME}}.\mathsf{KGen}(1^\lambda)$.
- The anamorphic key generation algorithm returns $\mathsf{pk}, \mathsf{sk}, \mathsf{dk} = (\mathsf{dk}_1, \mathsf{dk}_2)$, where $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}_1) \leftarrow \widetilde{\mathsf{AME}}.\mathsf{aKGen}(1^\lambda)$, and dk_2 is a randomly sampled key of a pseudorandom function F.
- The encryption algorithm $\mathsf{Enc}(\mathsf{pk}, \mathsf{m})$ returns $\mathsf{ct} = \widetilde{\mathsf{AME}}.\mathsf{Enc}(\mathsf{pk}, \mathsf{m})$.
- The anamorphic encryption algorithm $\mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}})$ returns $\mathsf{ct} = \widetilde{\mathsf{AME}}.\mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}_1, \mathsf{m}, \hat{\mathsf{m}})$ such that $\mathsf{F}_{\mathsf{dk}_2}(\mathsf{ct}) = \mathsf{m}$ (note that $\mathsf{m} \in \{0,1\}$).
- The decryption algorithm $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ returns $\widetilde{\mathsf{AME}}.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$.
- The anamorphic decryption algorithm aDec returns $\widetilde{\mathsf{AME}}.\mathsf{aDec}(\mathsf{sk}, \mathsf{dk}_1, \mathsf{ct})$.

SIND-nCCA security and STE-xCCA1 security are inherited from $\widetilde{\mathsf{AME}}$. However, it is easy to see that the compromise of dk_2 will totally break the confidentiality of m in the anamorphic ciphertext generated via aEnc. Namely, AME does not satisfy SIND-aCCA1 at all, concluding the proof. □

However, different from the CPA setting, it is unknown whether the implication from SIND-aCCA(1/3) and STE-xCCA1 to SIND-nCCA holds or not (see Remark 6).

Theorem 22 (SIND-nCCA ($\wedge$ STE-xCCA1) $\not\Rightarrow$ SIND-aCCA2 ($\wedge$ STE-xCCA1)). *For an STE-xCCA1-secure AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$*, SIND-nCCA security does not imply SIND-aCCA2 security, and vice versa.*

Proof. We first prove that SIND-nCCA $\wedge$ STE-xCCA1 $\not\Rightarrow$ SIND-aCCA2. Let $\widetilde{\mathsf{AME}}$ be an AME scheme satisfying SIND-nCCA and STE-xCCA1 (cf. the examples in the next section), and $1_{\hat{\mathcal{M}}}$ be an anamorphic message belongs to the anamorphic message space of $\widetilde{\mathsf{AME}}$. Similar to the proof of Theorem 19, we construct another AME scheme with anamorphic message space $\{0,1\}$ satisfying the first two properties but not SIND-aCCA2 as follows.

- The key generation algorithm returns $(\mathsf{pk}, \mathsf{sk}) \leftarrow \widetilde{\mathsf{AME}}.\mathsf{KGen}(1^\lambda)$.
- The anamorphic key generation algorithm returns $\mathsf{pk}, \mathsf{sk}, \mathsf{dk} = (\mathsf{dk}_1, \mathsf{dk}_2)$, where $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}_1) \leftarrow \widetilde{\mathsf{AME}}.\mathsf{aKGen}(1^\lambda)$, and dk_2 is a randomly sampled key of a pseudorandom function F.
- The encryption algorithm $\mathsf{Enc}(\mathsf{pk}, \mathsf{m})$ returns $\mathsf{ct} = \widetilde{\mathsf{AME}}.\mathsf{Enc}(\mathsf{pk}, \mathsf{m})$.
- The anamorphic encryption algorithm $\mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}})$ returns $\mathsf{ct} = \widetilde{\mathsf{AME}}.\mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}_1, \mathsf{m}, 1_{\hat{\mathcal{M}}})$ such that $\mathsf{F}_{\mathsf{dk}_2}(\mathsf{ct}) = \hat{\mathsf{m}}$ (note that $\hat{\mathsf{m}} \in \{0,1\}$).
- The decryption algorithm $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ returns $\widetilde{\mathsf{AME}}.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$.
- The anamorphic decryption algorithm aDec returns $\mathsf{F}_{\mathsf{dk}_2}(\mathsf{ct})$ if $\widetilde{\mathsf{AME}}.\mathsf{aDec}(\mathsf{sk}, \mathsf{dk}_1, \mathsf{ct}) = 1_{\hat{\mathcal{M}}}$, and $\perp$ otherwise.

The construction above inherits SIND-nCCA and STE-xCCA1 from $\widetilde{\mathsf{AME}}$. However, it does not satisfy SIND-aCCA2, since given $\mathsf{dk} = (\mathsf{dk}_1, \mathsf{dk}_2)$ and an anamorphic ciphertext $\mathsf{ct} = \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}})$, one can easily output the anamorphic message $\mathsf{F}_{\mathsf{dk}_2}(\mathsf{ct})$ (by correctness it must hold that $\widetilde{\mathsf{AME}}.\mathsf{aDec}(\mathsf{sk}, \mathsf{dk}_1, \mathsf{ct}) = 1_{\hat{\mathcal{M}}}$).

In the other direction, to prove SIND-nCCA $\not\Leftarrow$ SIND-aCCA2 $\wedge$ STE-xCCA1, we construct an AME scheme where the normal message is exposed in plain in the ciphertexts (note, however, that it should still ensure the CCA security of the anamorphic message). More precisely, assume $\widehat{\mathsf{AME}}$ is an AME scheme satisfying SIND-aCCA2 and STE-xCCA1, we construct AME as follows.

- The key generation algorithm returns $(\mathsf{pk}, \mathsf{sk}) \leftarrow \widehat{\mathsf{AME}}.\mathsf{KGen}(1^\lambda)$.
- The anamorphic key generation algorithm returns $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \widehat{\mathsf{AME}}.\mathsf{aKGen}(1^\lambda)$.
- The encryption algorithm $\mathsf{Enc}(\mathsf{pk}, \mathsf{m})$ returns $\mathsf{ct} = (\mathsf{m}, \hat{\mathsf{ct}})$, where $\hat{\mathsf{ct}} \leftarrow \widehat{\mathsf{AME}}.\mathsf{Enc}(\mathsf{pk}, \mathsf{m})$.
- The anamorphic encryption algorithm $\mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}})$ returns $\mathsf{ct} := (\mathsf{m}||\hat{\mathsf{ct}})$, where $\hat{\mathsf{ct}} \leftarrow \widehat{\mathsf{AME}}.\mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}})$.
- The decryption algorithm $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct} = (\mathsf{m}||\hat{\mathsf{ct}}))$ returns m if $\widehat{\mathsf{AME}}.\mathsf{Dec}(\mathsf{sk}, \hat{\mathsf{ct}}) = \mathsf{m}$, and $\perp$ otherwise.
- The anamorphic decryption algorithm $\mathsf{aDec}(\mathsf{sk}, \mathsf{dk}, \mathsf{ct} = (\mathsf{m}||\hat{\mathsf{ct}}))$ returns $\widehat{\mathsf{AME}}.\mathsf{aDec}(\mathsf{sk}, \mathsf{dk}, \hat{\mathsf{ct}})$ if $\widehat{\mathsf{AME}}.\mathsf{Dec}(\mathsf{sk}, \hat{\mathsf{ct}}) = \mathsf{m}$, and $\perp$ otherwise.[13]

SIND-aCCA2 and STE-xCCA1 of AME are inherited from $\widehat{\mathsf{AME}}$. However, AME does not satisfy SIND-nCCA at all, since the message is exposed in plain in the ciphertexts. $\square$

Relations with IND-CCA. We show that IND-CCA security is implied by many of our CCA notions.

Theorem 23 (SIND-nCCA $\wedge$ STE-xCCA1 $\Rightarrow$ IND-CCA, and SIND-nCCA $\not\Leftarrow$ IND-CCA $\wedge$ STE-xCCA1). *For an AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$*, SIND-nCCA security and STE-xCCA1 security together imply IND-CCA security. However, IND-CCA security, even combined with STE-xCCA1 security, does not imply SIND-nCCA security.*

[13] We emphasize that it is vital to ensure that normal decryption does not return $\perp$ before returning the anamorphic decryption result, as otherwise CCA security would not hold.

Proof. We first prove SIND-nCCA $\wedge$ STE-xCCA1 $\Rightarrow$ IND-CCA. By STE-xCCA1 security, SIND-nCCA$_{\Pi,\Sigma,b}(\mathcal{A})$ and IND-CCA$_{\Pi,b}(\mathcal{A})$ are indistinguishable to the adversary $\mathcal{A}$ who holds pk only (similar to the proof of Theorem 1 in the CPA setting). Therefore, the indistinguishability between SIND-nCCA$_{\Pi,\Sigma,0}(\mathcal{A})$ and SIND-nCCA$_{\Pi,\Sigma,1}(\mathcal{A})$ directly implies the indistinguishability between IND-CCA$_{\Pi,0}(\mathcal{A})$ and IND-CCA$_{\Pi,1}(\mathcal{A})$, which is exactly the definition of IND-CCA security.

The proof of the opposite direction is straightforward since both IND-CCA security and STE-xCCA1 security assume that the adversary gets no information with the double key dk. If sk is contained in dk then the normal message m is revealed immediately, and hence SIND-nCCA does not hold. □

Theorem 24 (SIND-aCCA1 $\wedge$ STE-xCCA1 $\Rightarrow$ IND-CCA, and SIND-aCCA1 $\not\Leftarrow$ IND-CCA $\wedge$ STE-xCCA1). *For an AME scheme $\mathsf{AME} = (\Pi, \Sigma)$, SIND-aCCA1 security and STE-xCCA1 security together imply the IND-CCA security. However, IND-CCA security, even combined with STE-xCCA1 security, does not imply SIND-aCCA1 security.*

Proof. We first prove SIND-aCCA1 $\wedge$ STE-xCCA1 $\Rightarrow$ IND-CCA. For $b \in \{0,1\}$, we introduce the hybrid game Hybrid$_{\Pi,\Sigma,b}(\mathcal{A})$ which is the same as SIND-aCCA1$_{\Pi,b}(\mathcal{A})$ except that only the public key pk but not the double key dk is passed to $\mathcal{A}$, and there is no anamorphic decryption oracle. By STE-xCCA1 security, Hybrid$_{\Pi,\Sigma,b}(\mathcal{A})$ and IND-CCA$_{\Pi,b}(\mathcal{A})$ are indistinguishable to the adversary $\mathcal{A}$ holding pk. More precisely, we can construct a reduction algorithm $\mathcal{A}'$ that breaks STE-xCCA1 as long as $\mathcal{A}$ can distinguish Hybrid$_{\Pi,\Sigma,b}(\mathcal{A})$ from IND-CCA$_{\Pi,b}(\mathcal{A})$. $\mathcal{A}'$ simulates either Hybrid$_{\Pi,\Sigma,b}(\mathcal{A})$ or IND-CCA$_{\Pi,b}(\mathcal{A})$ for $\mathcal{A}$ as follows:

- $\mathcal{A}'$ receives $(\mathsf{pk}, \mathsf{sk})$ from its STE-xCCA1 security experiment challenger, and passes pk to $\mathcal{A}$. Here $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$ if $\mathcal{A}'$ is accessing the experiment STE-xCCA1$_{\Pi,\Sigma,0}(\mathcal{A}')$ (the normal mode), and $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{aKGen}(1^\lambda)$ if $\mathcal{A}'$ is accessing the experiment STE-xCCA1$_{\Pi,\Sigma,1}(\mathcal{A}')$ (the anamorphic mode).
- Whenever $\mathcal{A}$ asks the challenge oracle with $(\mathsf{m}_0, \mathsf{m}_1)$, $\mathcal{A}'$ randomly samples $\hat{\mathsf{m}}$, sends $(\mathsf{m}_b, \hat{\mathsf{m}})$ to its challenger, and passes the received ct^* to $\mathcal{A}$ directly.
- Whenever $\mathcal{A}$ asks a decryption query with $\mathsf{ct} \neq \mathsf{ct}^*$, the reduction algorithm $\mathcal{A}'$ returns $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ (recall that $\mathcal{A}'$ holds sk).
- Finally $\mathcal{A}'$ outputs what $\mathcal{A}$ outputs.

For $b \in \{0,1\}$, if $\mathcal{A}'$ is accessing the experiment STE-xCCA1$_{\Pi,\Sigma,0}(\mathcal{A}')$ (the normal mode), then $\mathcal{A}'$ simulates IND-CCA$_{\Pi,b}(\mathcal{A})$ for $\mathcal{A}$, and if $\mathcal{A}'$ is accessing the experiment STE-xCCA1$_{\Pi,\Sigma,1}(\mathcal{A}')$ (the anamorphic mode), then $\mathcal{A}'$ simulates Hybrid$_{\Pi,\Sigma,b}(\mathcal{A})$ for $\mathcal{A}$. Therefore, Hybrid$_{\Pi,\Sigma,b}(\mathcal{A})$ and IND-CCA$_{\Pi,b}(\mathcal{A})$ are indistinguishable to the adversary $\mathcal{A}$ due to STE-xCCA1 security.

Finally, any adversary capable of distinguishing Hybrid$_{\Pi,\Sigma,1}(\mathcal{A})$ from Hybrid$_{\Pi,\Sigma,0}(\mathcal{A})$ (where only pk is provided) can be directly used to distinguish SIND-aCCA1$_{\Pi,\Sigma,1}(\mathcal{A})$ and SIND-aCCA1$_{\Pi,\Sigma,0}(\mathcal{A})$ (where both pk and dk are provided), concluding the proof of SIND-aCCA1 $\wedge$ STE-xCCA1 $\Rightarrow$ IND-CCA.

The proof of SIND-aCCA1 $\not\Leftarrow$ IND-CCA $\wedge$ STE-xCCA1, is implied by an AME scheme where sk is a part of dk. Therefore, holding dk is sufficient to do both the normal decryption and the anamorphic decryption, and hence SIND-aCCA1 does not hold at all. However, IND-CCA security and STE-xCCA1 security may still hold. □

Corollary 2. *For an AME scheme $\mathsf{AME} = (\Pi, \Sigma)$, SIND-aCCA3 security and STE-xCCA1 security together imply IND-CCA security. However, IND-CCA security, combined with STE-xCCA1 security, does not imply SIND-aCCA3 security.*

Now we prove that IND-CCA and SIND-aCCA2 are incomparable, even conditioned on STE-xCCA1 security. This is straightforward since IND-CCA ensures the confidentiality of the normal message m, while SIND-aCCA2 ensures the confidentiality of the anamorphic message $\hat{\mathsf{m}}$.

Theorem 25 (IND-CCA ($\wedge$ STE-xCCA1) $\not\Leftrightarrow$ SIND-aCCA2 ($\wedge$ STE-xCCA1)). *For an STE-xCCA1-secure AME scheme $\mathsf{AME} = (\Pi, \Sigma)$, IND-CCA security does not imply SIND-aCCA2 security, and vice versa.*

Proof. On one direction that IND-CCA $\wedge$ STE-xCCA1 $\not\Rightarrow$ SIND-aCCA2, we construct a counterexample of AME where dk provides the full access to $\hat{\mathsf{m}}$ in an anamorphic ciphertext, which is the same as the first example in the proof of Theorem 22.

On the other direction that IND-CCA $\not\Leftarrow$ SIND-aCCA2 $\wedge$ STE-xCCA1, we construct a counterexample of AME where the normal message m is exposed in plain in ciphertexts, which is the same as the second example in the proof of Theorem 22. □

Similar to that in the CPA setting, conditioned on SIND-xCCA1, we have the following equivalence.

Theorem 26. *For an SIND-xCCA1-secure AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$*, the following four security notions are equivalent:*

1. *SIND-nCCA (Def. 28, given* dk*,* Dec*, and* aDec*, to distinguish* $\mathsf{Enc}(\mathsf{m}_0)$ *from* $\mathsf{Enc}(\mathsf{m}_1)$*).*
2. *SIND-xCCA2 (Def. 27, given* dk*,* Dec*, and* aDec*, to distinguish* $\mathsf{Enc}(\mathsf{m}_0)$ *from* $\mathsf{aEnc}(\mathsf{m}_1, \hat{\mathsf{m}})$*).*
3. *SIND-aCCA1 (Def. 29, given* dk*,* Dec*, and* aDec*, to distinguish* $\mathsf{aEnc}(\mathsf{m}_0, \hat{\mathsf{m}})$ *from* $\mathsf{aEnc}(\mathsf{m}_1, \hat{\mathsf{m}})$*).*
4. *SIND-aCCA3 (Def. 31, given* dk*,* Dec*, and* aDec*, to distinguish* $\mathsf{aEnc}(\mathsf{m}_0, \hat{\mathsf{m}}_0)$ *from* $\mathsf{aEnc}(\mathsf{m}_1, \hat{\mathsf{m}}_1)$*).*

The proof is exactly the same as that for Theorem 11 except that there are two additional decryption oracles Dec and aDec, which, are equivalently provided in all security games.

Theorem 27 (SIND-aCCA2 ($\wedge$ SIND-xCCA1) $\not\Rightarrow$ SIND-aCCA3). *For an SIND-xCCA1-secure AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$*, SIND-aCCA2 does not imply SIND-aCCA3.*

Proof. Neither SIND-aCCA2 nor SIND-xCCA1 guarantees the confidentiality of m in ciphertexts. Therefore, the theorem can be proved via a counterexample of an AME scheme where m is exposed in plain in the (normal or anamorphic) ciphertext. □

5 Randomness-Recovering PKE: Achieving All Security Notions

Given the relationships we have established among all security notions (cf. Fig. 1), an AME scheme is most secure if it achieves STE-xCCA1 (Def. 23), SIND-xCCA1 (Def. 26), and SIND-nCCA (Def. 28) simultaneously. In this subsection we review the generic compiler of AME from randomness-recovering PKE (rr-PKE) [KPP+23b], and show that under a stronger assumption on the underlying PKE scheme and symmetric encryption scheme, the rr-PKE compiler can achieve all of these security notions. This construction validates the utility of the security notions our methodology derived, as we will show that, in addition to being meaningful, they are indeed *achievable*.

Definition 32 (Randomness Recovering PKE [KPP+23b]). *A PKE scheme* $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is randomness recovering (rr-PKE), if there is a randomness recovering PPT decryption algorithm* rrDec *such that* $\mathsf{rrDec}(\mathsf{sk}, \mathsf{ct}) = (\mathsf{m}, r)$*, where* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$*, and* $\mathsf{ct} := \mathsf{Enc}(\mathsf{pk}, \mathsf{m}; r)$*.*

The idea of the AME compiler from rr-PKE in [KPP+23b] is to embed the anamorphic message into the randomness r used in the encryption. More precisely, let Π be an IND-CPA-secure PKE scheme, and SE be a PR-CT-secure symmetric encryption scheme. In the anamorphic triplet Σ, the public and secret keys $(\mathsf{pk}, \mathsf{sk})$ are generated as in Π, and the double key dk is a secret key of SE. The anamorphic encryption algorithm, given a normal message m and an anamorphic message $\hat{\mathsf{m}}$, returns the ciphertext $\mathsf{ct} := \mathsf{Enc}(\mathsf{pk}, \mathsf{m}; r)$, where $r \leftarrow \mathsf{SE.Enc}(\mathsf{dk}, \hat{\mathsf{m}})$. The anamorphic decryption, given a ciphertext ct, first invokes the randomness-recovering decryption algorithm to obtain $\mathsf{rrDec}(\mathsf{sk}, \mathsf{ct}) = (\mathsf{m}, r)$, and then decrypts r using dk to obtain $\mathsf{SE.Dec}(\mathsf{dk}, r) = \hat{\mathsf{m}}$. It was proved in [KPP+23b] that PR-CT security of SE directly implies STE-xCPA1 security of $\mathsf{AME} = (\Pi, \Sigma)$.

Now we consider enhancing this generic compiler to the CCA setting. It seems that we are done if the security properties of the underlying PKE and SE schemes are enhanced from CPA to CCA. However, we argue that it is not STE-xCCA1-secure and SIND-xCCA1-secure yet, for the following reasons.

1. **Anamorphic decryption oracle.** For a challenge ciphertext $\mathsf{ct}^* = \mathsf{Enc}(\mathsf{pk}, \mathsf{m}; r)$ where r is either a true randomness or a symmetric encryption of $\mathsf{SE.Enc}(\mathsf{dk}, \hat{\mathsf{m}})$, the adversary can first decrypt ct^* to obtain r, and then reuse this r to encrypt another message $\mathsf{m}' \neq \mathsf{m}$ to obtain ciphertext $\mathsf{ct}' = \mathsf{Enc}(\mathsf{pk}, \mathsf{m}'; r)$. Therefore, with the result of $\mathsf{aDec}(\mathsf{sk}, \mathsf{dk}, \mathsf{ct}')$ from anamorphic decryption oracle, which, in the normal mode is $\perp$ and in the anamorphic mode is $\hat{\mathsf{m}}$, the adversary is able to distinguish the two modes, and STE-xCCA1 does not hold at all.
2. **Leakage from the double key.** To prove SIND-xCCA1 security, we need to argue that a standard encryption of m is indistinguishable from an anamorphic encryption of $(\mathsf{m}, \hat{\mathsf{m}})$. Namely, $\mathsf{Enc}(\mathsf{pk}, \mathsf{m}; r_0)$ with r_0 randomly sampled from the randomness space is indistinguishable from $\mathsf{Enc}(\mathsf{pk}, \mathsf{m}; r_1)$ with $r_1 \leftarrow \mathsf{SE.Enc}(\mathsf{dk}, \hat{\mathsf{m}})$. At first glance, the two games appear to be identical. But unfortunately, this is not the case since no freshness of r_1 is guaranteed (against the adversary), due to the leakage of the double key dk.

We propose several novel methods to solve these problems.

1. To solve the first problem caused by the anamorphic decryption oracle, we replace SE with an authenticated encryption scheme AE,[14] and modify the scheme such that the randomness r is a symmetric encryption of both m and $\hat{\mathsf{m}}$, i.e., $r \leftarrow \mathsf{AE.Enc}(\mathsf{dk}, (\mathsf{m}||\hat{\mathsf{m}}))$.[15] Meanwhile, in anamorphic decryption w.r.t. a ciphertext $\mathsf{ct} = \mathsf{Enc}(\mathsf{pk}, \mathsf{m}; r)$, it returns $\perp$ directly if, $\mathsf{AE.Dec}(\mathsf{dk}, r) = (\mathsf{m}', \hat{\mathsf{m}})$ and $\mathsf{m}' \neq \mathsf{m}$. Now the adversary who does not know dk can neither generate a randomness that did not decrypt to the failure indicator (the security of AE), nor reuse the randomness r to obtain a different ciphertext, since the ciphertext is uniquely determined by pk, m, and r, and m is fixed once r is determined in the anamorphic encryption mode.
2. To solve the second problem caused by the imperfect randomness due to the leakage of dk, we build an underlying authenticated encryption from IC, and put additional randomness $\hat{r}$ in the encryption. Therefore, even if dk (i.e., the secret key of the symmetric encryption) and the plaintext are known to the adversary, it still cannot predict $r := \mathsf{IC}_{\mathsf{dk}}(\mathsf{m}, \hat{\mathsf{m}}, \hat{r})$, which is randomly sampled from the output space of IC (i.e., the randomness space of the PKE scheme Π). Consequently, we are able to argue the equivalence of the two games in the proof of SIND-xCCA1.

As discussed above, we need an authenticated encryption scheme based on an ideal cipher. One of the simplest approaches to achieve this is following the Encode-then-Encipher (EtE) paradigm [BR00]. We note that EtE requires the maximum message length of EtE to be smaller than the input message length of the inner ideal cipher. To extend the input domain, one may introduce a more *generic* authenticated encryption mode that supports arbitrary length messages with a security proof in the ideal cipher model such as Galois Counter-Mode (GCM) scheme [HTT18] or indifferentiable authenticated encryption [BF18]. Another way of addressing the length issue is assuming a stronger primitive instead of an ideal cipher, e.g., an ideal *accordion cipher mode* [CDD+24], or, in short, an accordion mode.[16] Nonetheless, in this work we stick to the simpler case, which is enough to show feasibility: applying EtE to an ideal cipher.

Let $\mathsf{IC} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be an ideal cipher and $n = n_1 + n_2 + \lambda_1 + \lambda_2$. The authenticated encryption scheme AE with plaintext space $\{0,1\}^{n_1+n_2}$ is constructed as follows.

– $\mathsf{AE.KGen}(1^\lambda)$: The key generation algorithm returns a key dk randomly sampled from $\{0,1\}^k$.

[14] The concurrent work by Jaeger and Stracovsky [JS24] mentioned the use of authentication encryption to achieve robustness, but in the syntax of anamorphic signatures.

[15] It is also possible to use AE with labels and compute r as an encryption of $\hat{\mathsf{m}}$ with m being the label. This would save space for the anamorphic plaintext.

[16] Accordion mode is a relatively new primitive, an ongoing proposal of NIST at the time of writing. We can see it as a variable-length extension of an ideal cipher, which could plausibly be a construction based on an ideal cipher. Therefore, it could be an equivalent assumption to the random oracle model or the ideal cipher model [CPS08,HKT11] in a single-stage game setting [RSS11]. By assuming an ideal accordion mode, EtE based on the accordion mode naturally supports an arbitrary length input. We believe this will be an interesting model in the near future if accordion mode becomes popular.

– $\mathsf{AE.Enc}(\mathsf{dk}, (\mathsf{m}||\hat{\mathsf{m}}))$: The encryption algorithm samples a random string $\hat{r} \leftarrow \{0,1\}^{\lambda_1}$, and returns $r := \mathsf{IC.Enc}(\mathsf{dk}, (\mathsf{m}||\hat{\mathsf{m}}||\hat{r}||0^{\lambda_2}))$.
– $\mathsf{AE.Dec}(\mathsf{dk}, r)$: The decryption algorithm first invokes $\mathsf{IC.Dec}(\mathsf{dk}, r)$ to get $(\mathsf{m}'||\hat{\mathsf{m}}'||\hat{r}'||\mu)$. If $\mu = 0^{\lambda_2}$, it returns $(\mathsf{m}', \hat{\mathsf{m}}')$. Otherwise, it returns $\perp$.

Now we are ready to describe the compiler of secure AME schemes from rr-PKE. Let $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a CCA-secure rr-PKE scheme, and AE be an authenticated encryption scheme from ideal cipher IC constructed as above, we design the anamorphic triplet Σ as follows:

– The anamorphic key generation algorithm $\mathsf{aKGen}(1^\lambda)$ returns $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk})$, where $(\mathsf{pk}, \mathsf{sk}) \leftarrow \Pi.\mathsf{KGen}(1^\lambda)$, and $\mathsf{dk} \leftarrow \mathsf{AE.KGen}(1^\lambda)$.
– The anamorphic encryption algorithm $\mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}})$ returns $\mathsf{ct} := \Pi.\mathsf{Enc}(\mathsf{pk}, \mathsf{m}; r)$, where $r \leftarrow \mathsf{AE.Enc}(\mathsf{dk}, (\mathsf{m}||\hat{\mathsf{m}}))$.
– The anamorphic decryption algorithm $\mathsf{aDec}(\mathsf{sk}, \mathsf{dk}, \mathsf{ct})$ first invokes the randomness recovering algorithm to get $\mathsf{rrDec}(\mathsf{sk}, \mathsf{ct}) = (\mathsf{m}, r)$, runs $\mathsf{AE.Dec}(\mathsf{dk}, r)$ to obtain $(\mathsf{m}', \hat{\mathsf{m}})$, and then returns $\hat{\mathsf{m}}$ if $\mathsf{m} = \mathsf{m}'$ and $\perp$ otherwise (including the case $\mathsf{AE.Dec}$ returns $\perp$).

The correctness of Σ follows directly from the correctness of Π and AE.

Theorem 28 (Security of AME from rrPKE). *If Π is a CCA-secure rr-PKE scheme, IC is an ideal cipher and AE is a secure AE scheme constructed from IC as above, then the AME scheme $\mathsf{AME} = (\Pi, \Sigma)$ with the anamorphic triplet described above has STE-xCCA1 security, SIND-xCCA1 security, and SIND-nCCA security.*

Proof. We first prove the STE-xCCA1 security of AME. To prove the indistinguishability of the experiment $\text{STE-xCCA1}_{\Pi,\Sigma,1}(\mathcal{A})$ (the anamorphic mode) from the experiment $\text{STE-xCCA1}_{\Pi,\Sigma,0}(\mathcal{A})$ (the normal mode), we introduce a hybrid game $\text{Hybrid}_{\Pi,\Sigma}(\mathcal{A})$ which is the same as $\text{STE-xCCA1}_{\Pi,\Sigma,1}(\mathcal{A})$, except that all anamorphic decryption queries from $\mathcal{A}$ output failure indicator $\perp$.

We analyze the difference between $\text{STE-xCCA1}_{\Pi,\Sigma,1}(\mathcal{A})$ and $\text{Hybrid}_{\Pi,\Sigma}(\mathcal{A})$ as follows. For simplicity we consider single challenge setting here and the multiple challenge setting follows via the standard hybrid argument. Let $(\mathsf{m}^*, \hat{\mathsf{m}}^*)$ be the challenge query by $\mathcal{A}$, and $\mathsf{ct}^* := \Pi.\mathsf{Enc}(\mathsf{pk}, \mathsf{m}^*; r^*)$ be the corresponding challenge ciphertext, where $r^* \leftarrow \mathsf{AE.Enc}(\mathsf{dk}, (\mathsf{m}^*||\hat{\mathsf{m}}^*))$. If the adversary $\mathcal{A}$ queries a ciphertext $\mathsf{ct} \neq \mathsf{ct}^*$ such that $\mathsf{aDec}(\mathsf{sk}, \mathsf{dk}, \mathsf{ct}) \neq \perp$, then $\mathcal{A}$ is able to distinguish $\text{STE-xCCA1}_{\Pi,\Sigma,1}(\mathcal{A})$ from $\text{Hybrid}_{\Pi,\Sigma}(\mathcal{A})$. Assume $\mathsf{rrDec}(\mathsf{sk}, \mathsf{ct}) = (\mathsf{m}, r)$, we further discuss in the following cases.

– Case 1. $\mathsf{m} = \mathsf{m}^*$ and $r = r^*$. This implies that $\mathsf{ct} = \mathsf{ct}^*$ and the oracle will return $\perp$ directly.
– Case 2. $\mathsf{m} \neq \mathsf{m}^*$ but $r = r^*$. Since $\mathsf{AE.Dec}(\mathsf{dk}, r) = (\mathsf{m}^*, \hat{\mathsf{m}}^*)$, the anamorphic decryption oracle will return $\perp$ since $\mathsf{m} \neq \mathsf{m}^*$.
– Case 3. $r \neq r^*$. The anamorphic decryption does not return $\perp$ only if $\mathsf{AE.Dec}(\mathsf{dk}, r) \neq \perp$, which means that the adversary generates a distinct but valid ciphertext, and hence breaks the security of AE.

Therefore, $\text{STE-xCCA1}_{\Pi,\Sigma,1}(\mathcal{A})$ and $\text{Hybrid}_{\Pi,\Sigma}(\mathcal{A})$ are indistinguishable to the adversary, due to the security of AE.

Then we analyze the difference between $\text{Hybrid}_{\Pi,\Sigma}(\mathcal{A})$ and $\text{STE-xCCA1}_{\Pi,\Sigma,0}(\mathcal{A})$. Now both anamorphic decryption oracles return $\perp$ all the time, and the only difference between these two games is the randomness r used in encryption, which is randomly sampled in $\text{STE-xCCA1}_{\Pi,\Sigma,0}(\mathcal{A})$, and $r \leftarrow \mathsf{AE.Enc}(\mathsf{dk}, (\mathsf{m}||\hat{\mathsf{m}}))$ in $\text{Hybrid}_{\Pi,\Sigma}(\mathcal{A})$. Due to the security (ciphertext pseudorandomness) of AE, which is implied by the property of ideal cipher IC, they are indistinguishable to the adversary $\mathcal{A}$. This finishes the proof of STE-xCCA1 security of AME.

Next we prove SIND-xCCA1, which requires that $\mathsf{Enc}(\mathsf{pk}, \mathsf{m}; r_0)$ is indistinguishable from $\mathsf{Enc}(\mathsf{pk}, \mathsf{m}; r_1)$ to the adversary $\mathcal{A}$ holding the double key dk, where r_0 is randomly sampled from the randomness space, and $r_1 \leftarrow \mathsf{AE.Enc}(\mathsf{dk}, (\mathsf{m}||\hat{\mathsf{m}}))$. Recall that dk is independent from $(\mathsf{pk}, \mathsf{sk})$, and $\mathsf{AE.Enc}(\mathsf{dk}, (\mathsf{m}||\hat{\mathsf{m}})) = \mathsf{IC.Enc}(\mathsf{dk}, (\mathsf{m}||\hat{\mathsf{m}}||\hat{r}_1||0^{\lambda_2}))$, where $\hat{r}_1 \in \{0,1\}^\lambda$ is a random string sampled in the encryption. Let Q_{IC} denote

the number of queries to the ideal cipher IC by the adversary. Unless the adversary has queried $\mathsf{IC.Enc}$ on $(\mathsf{m}||\hat{\mathsf{m}}||\hat{r}_1||0^{\lambda_2})$, which happens with probability at most $Q_{\mathsf{IC}} \cdot 2^{-\lambda_1}$, $r_1 = \mathsf{IC.Enc}(\mathsf{dk}, (\mathsf{m}||\hat{\mathsf{m}}||\hat{r}_1||0^{\lambda_2}))$ is a fresh and random string in the ciphertext space (i.e., the randomness space of PKE scheme Π). Consequently, the experiment SIND-xCCA1$_{\Pi,\Sigma,0}(\mathcal{A})$ is identical to the experiment SIND-xCCA1$_{\Pi,\Sigma,1}(\mathcal{A})$, completing the proof of SIND-xCCA1 security.

Finally, SIND-nCCA security of AME is directly implied by the CCA security of the underlying PKE scheme, since the symmetric key dk is independent from the public and secret key pair $(\mathsf{pk}, \mathsf{sk})$ of the PKE scheme. Taking all together, Theorem 28 holds consequently. □

AME from rr-PKE in Practice. Recall that the core question in anamorphic encryption, as pointed out by the original work [PPY22], is not merely *"can we construct an AME scheme?"*, but rather: *"are any existing encryption schemes also anamorphic?"* In other words, their goal is to identify anamorphic extensions of existing PKE schemes. In practice, many CCA-secure PKE schemes are built from the popular Fujisaki-Okamoto transform [FO99,FO13] applied to a CPA-secure PKE/KEM scheme, including the NIST post-quantum standard ML-KEM (in the KEM setting rather than PKE). The fact that the Fujisaki-Okamoto transform allows for randomness recovery supports the practical relevance of our rr-PKE-based AME constructions.

However, we note that the widely used (KEM+DEM)-hybrid encryption paradigm [CS03] does not achieve our strongest CCA security guarantees. In particular, STE-xCCA1 security is not guaranteed, since the dictator, after observing an anamorphic encryption $(\mathsf{ct}_{\mathsf{KEM}}, \mathsf{ct}_{\mathsf{DEM}})$ can reuse $\mathsf{ct}_{\mathsf{KEM}}$—which hides the anamorphic message—while replacing $\mathsf{ct}_{\mathsf{DEM}}$ with a fresh component to create a new anamorphic ciphertext. We leave constructing CCA-secure AME from arbitrary CCA-secure PKE schemes (even in the presence of idealized functionalities) as interesting future work.

6 Security of Asymmetric AME against Collusion

In this section, we focus on asymmetric AME and explore the security against the collusion of the dictator and anamorphic senders.

6.1 Security Notions

We now define CIND-xCPA1, CIND-aCPA2 and their CCA security extensions, which are only (well-)defined in the asymmetric setting.

CIND-xCPA1 and CIND-xCCA1. Fig. 29 shows the CIND-xCPA1 security game (left) and the CIND-xCCA1 game (right) for asymmetric anamorphic encryption. At a high level, it captures a scenario where an anamorphic sender trusted with the (sender) double key sdk colludes with the dictator who has exfiltrated sk, and together they attempt to distinguish between normal encryptions sent by an honest user and anamorphic encryptions sent (by another user trusted with the double key) to a anamorphic user, i.e., a user who has run anamorphic key generation. Their failure to do so ensures that the dictator cannot collude with a sender to determine whether or not other users are sending anamorphic messages to the recipient. In the CPA setting no auxiliary oracle for anamorphic encryption is needed, as the adversary can simulate this using the $(\mathsf{pk}, \mathsf{sdk})$ it obtained. In the CCA setting, the adversary is additionally equipped with an anamorphic decryption oracle $\textsc{aDec}$. To our knowledge, these two definitions have not been considered in prior literature.

Definition 33 (CIND-xCPA1 and CIND-xCCA1 Security of Asymmetric AME). *An asymmetric AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$ *is CIND-X-secure for* $X \in \{xCPA1, xCCA1\}$*, if for every PPT adversary, the advantage*

$$\mathsf{Adv}^{\text{cind-x}}_{\Pi,\Sigma}(\mathcal{A}) := |\Pr[\text{CIND-X}_{\Pi,\Sigma,1}(\mathcal{A}) = 1] - \Pr[\text{CIND-X}_{\Pi,\Sigma,0}(\mathcal{A}) = 1]|$$

is negligible in λ*, where* CIND-X$_{\Pi,\Sigma,b}(\mathcal{A})$ *are defined in Fig. 29.*

We have the following trivial implications.

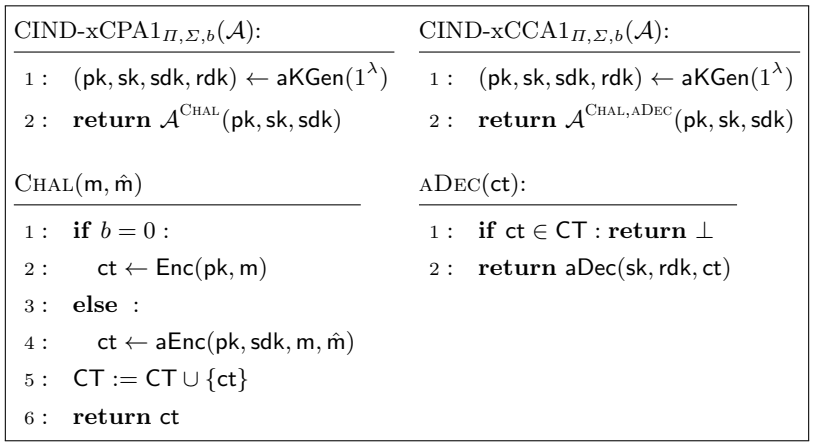


Fig. 29. CIND-xCPA1 and CIND-xCCA1 security games for an asymmetric AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

$$\text{CPA setting: DIND-xCPA1} \Longleftarrow \text{CIND-xCPA1} \Longrightarrow \text{SIND-xCPA1, and}$$
$$\text{CCA setting: DIND-xCCA1} \Longleftarrow \text{CIND-xCCA1} \Longrightarrow \text{SIND-xCCA1.}$$

CIND-aCPA2 and CIND-aCCA2. Fig. 30 shows the CIND-aCPA2 security game (left) and the CIND-aCCA2 for asymmetric anamorphic encryption. At a high level, it captures a scenario where an anamorphic sender trusted with the (sender) double key sdk colludes with the dictator who has exfiltrated sk, and together they attempt to learn information about an anamorphic message contained within an anamorphic ciphertext sent to an anamorphic user, i.e., a user who has run anamorphic key generation. This definition captures the above via a standard IND-CPA-style definition for the anamorphic message, where the adversary can influence both the plain and anamorphic messages. In the CPA setting no auxiliary oracle for anamorphic encryption is needed, as the adversary can simulate this using the (pk, sdk) it obtained. In the CCA setting, the adversary is additionally equipped with an anamorphic decryption oracle ADEC.

The CIND-aCPA2 notion was first proposed in [CGM24a] as Fully Asymmetric property. A single-challenge version of this definition was considered in an early version of [BPRS25], which was uploaded to ePrint after this work was first uploaded. There, it is captured by the game labeled GsIND-CPA.

Definition 34 (CIND-aCPA2 and CIND-aCCA2 Security of Asymmetric AME). *An asymmetric AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$ *is CIND-X-secure for* $X \in \{aCPA2, aCCA2\}$*, if for every PPT adversary, the advantage*

$$\mathsf{Adv}^{\text{cind-x}}_{\Pi,\Sigma}(\mathcal{A}) := |\Pr[\text{CIND-X}_{\Pi,\Sigma,1}(\mathcal{A}) = 1] - \Pr[\text{CIND-X}_{\Pi,\Sigma,0}(\mathcal{A}) = 1]|$$

is negligible in λ*, where* $\text{CIND-X}_{\Pi,\Sigma,b}(\mathcal{A})$ *are defined in Fig. 30.*

It is easy to see that CIND-xCPA1 $\Longrightarrow$ CIND-aCPA2 and CIND-xCCA1 $\Longrightarrow$ CIND-aCCA2.

6.2 Relations with Others

Now we investigate the relationships among CIND-xCPA1, CIND-aCPA2, and other security notions. The results also hold in the CCA setting.

It is easy to see that CIND-xCPA1 is strictly stronger than SIND-xCPA1 and DIND-xCPA1, and similarly CIND-aCPA2 is strictly stronger than SIND-aCPA2 and DIND-aCPA2. Then, what about the relation between STE-xCPA1 and CIND-(xCPA1/aCPA2)? In the following theorem, we will see that those notions are incomparable.

Theorem 29 (STE-xCPA1 $\nLeftrightarrow$ CIND-(xCPA1/aCPA2), and STE-xCCA1 $\nLeftrightarrow$ CIND-(xCCA1/aCCA2)). *For an asymmetric AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$*, STE-xCPA1 does not imply CIND-(xCPA1/aCPA2), and vice versa. Furthermore, STE-xCCA1 does not imply CIND-(xCCA1/aCCA2), and vice versa.*

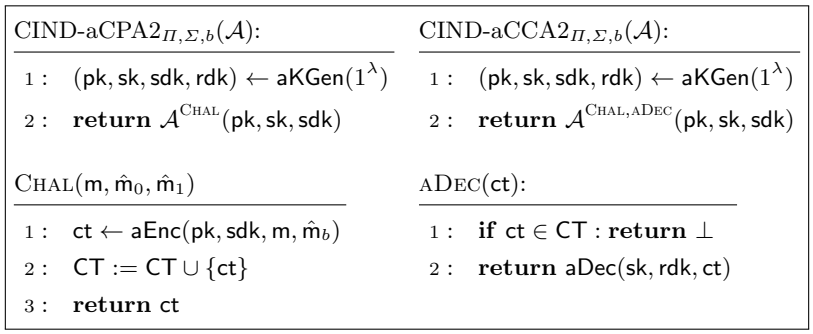


Fig. 30. CIND-aCPA2 and CIND-aCCA2 security games for an asymmetric AME scheme $\mathsf{AME} = (\Pi, \Sigma)$.

Proof. The proof of STE-xCPA1 $\not\Rightarrow$ CIND-(xCPA1/aCPA2) is addressed by a symmetric AME scheme (a specific case of asymmetric AME when $\mathsf{sdk} = \mathsf{rdk}$), where holding both sk and $\mathsf{sdk} = \mathsf{rdk}$ enables one to totally read both the normal message and the anamorphic message in a ciphertext. The proof also applies in the CCA setting.

The other direction is proven by the counterexample of an asymmetric AME scheme where key pairs by the (normal) key generation and anamorphic key generation are distinguishable. For example, the secret key sk output by KGen is always appended with a bit 0, while the secret key sk output by aKGen is appended with a bit 1. □

Relations with Security Notions against the Anamorphic Sender. Now we investigate the relations between the two new notions (CIND-xCPA1 and CIND-aCPA2) and security notions against the anamorphic sender (SIND-nCPA and SIND-aCPA(1/2/3)).

Theorem 30 (SIND-(nCPA/aCPA1/aCPA3) $\not\Rightarrow$ CIND-aCPA2). *For an asymmetric AME scheme $\mathsf{AME} = (\Pi, \Sigma)$, SIND-nCPA, SIND-aCPA1, or SIND-aCPA3 do not imply CIND-aCPA2. Furthermore, SIND-nCCA, SIND-aCCA1, or SIND-aCCA3 do not imply CIND-aCCA2.*

Proof. The separation in the CPA setting can be easily seen by the special case $\mathsf{rdk} = \mathsf{sdk}$. In this case, even if SIND-nCPA (or SIND-aCPA1 or SIND-aCPA3) holds, it is trivially not CIND-aCPA2 because an adversary has both sk and $\mathsf{rdk} = \mathsf{sdk}$ so that it can freely run anamorphic decryption. The proof in the CCA setting follows in a similar way. □

Theorem 31 (SIND-(nCPA/aCPA1/aCPA3) $\not\Leftarrow$ CIND-xCPA1). *For an asymmetric AME scheme $\mathsf{AME} = (\Pi, \Sigma)$, CIND-xCPA1 does not imply SIND-nCPA, SIND-aCPA1, or SIND-aCPA3. Furthermore, CIND-xCCA1 does not imply SIND-nCCA, SIND-aCCA1, or SIND-aCCA3.*

Proof. In the CPA setting, the proof can be done via the following counterexample. Let Π be a PR-CT secure PKE scheme with public key sdk and secret key rdk. We construct an AME scheme where the normal encryption returns $(\mathsf{m}||str)$ with str some random string, and the anamorphic encryption returns $(\mathsf{m}||\Pi.\mathsf{Enc}(\mathsf{sdk},\hat{\mathsf{m}}))$. The construction is CIND-xCPA1 secure due to the PR-CT security of Π. However, it is trivially insecure as SIND-nCPA, SIND-aCPA1, or SIND-aCPA3.

In the CCA setting, the proof can be done by a counterexample of AME scheme from rr-PKE, where the normal encryption returns $(\mathsf{m}||str_1||str_2)$ with str_1, str_2 some random strings, and the anamorphic encryption returns $(\mathsf{m}||\Pi.\mathsf{Enc}(\mathsf{sdk},\hat{\mathsf{m}};r)||H(\mathsf{m},\hat{\mathsf{m}},r))$, and $H(\cdot)$ is a random oracle. □

Since CIND-xCPA1 (resp., CIND-xCCA1) is strictly stronger than CIND-aCPA2 (resp., CIND-aCCA2), we immediately obtain the following two corollaries.

Corollary 3. *For an asymmetric AME scheme $\mathsf{AME} = (\Pi, \Sigma)$, SIND-nCPA, SIND-aCPA1, or SIND-aCPA3 do not imply CIND-xCPA1. Moreover, SIND-nCCA, SIND-aCCA1, or SIND-aCCA3 do not imply CIND-xCCA1.*

Corollary 4. *For an asymmetric AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$*, CIND-aCPA2 does not imply SIND-nCPA, SIND-aCPA1, or SIND-aCPA3. Moreover, CIND-aCCA2 does not imply SIND-nCCA, SIND-aCCA1, or SIND-aCCA3.*

7 AME Schemes Satisfying All Asymmetric Security Notions

In this subsection we modify the generic compiler from rr-PKE in Section 5 to the asymmetric setting, and describe how to additionally achieve CIND-xCCA1 security.

7.1 Achieving CIND-xCCA1 Security (Only)

The high-level idea is to replace the authenticated encryption scheme (which encrypts the anamorphic message into "randomness" used in the PKE scheme) with a PKE scheme Π', so that the double key can be split into two parts, the public key sdk of Π' that is used in anamorphic encryption, and the secret key rdk of Π' that is used in anamorphic decryption. To guarantee that such a replacement will not sacrifice the security of AME, Π' is required to hold both the CCA security and ciphertext pseudorandomness (cf. Def. 4).

More precisely, let Π be an IND-CCA-secure rr-PKE scheme, and Π' be a PR-CCA-secure PKE scheme, we design the anamorphic triplet Σ associated with Π as follows:

- The anamorphic key generation algorithm $\mathsf{aKGen}(1^\lambda)$ returns $(\mathsf{pk}, \mathsf{sk}, \mathsf{sdk}, \mathsf{rdk})$, where $(\mathsf{pk}, \mathsf{sk}) \leftarrow \Pi.\mathsf{KGen}(1^\lambda)$, and $(\mathsf{sdk}, \mathsf{rdk}) \leftarrow \Pi'.\mathsf{KGen}(1^\lambda)$.
- The anamorphic encryption algorithm $\mathsf{aEnc}(\mathsf{pk}, \mathsf{sdk}, \mathsf{m}, \hat{\mathsf{m}})$ returns $\mathsf{ct} := \Pi.\mathsf{Enc}(\mathsf{pk}, \mathsf{m}; r)$, where $r \leftarrow \Pi'.\mathsf{Enc}(\mathsf{sdk}, (\mathsf{m}||\hat{\mathsf{m}}))$.[17]
- The anamorphic decryption algorithm $\mathsf{aDec}(\mathsf{sk}, \mathsf{rdk}, \mathsf{ct})$ invokes the randomness recovering algorithm to get $\Pi.\mathsf{rrDec}(\mathsf{sk}, \mathsf{ct}) = (\mathsf{m}, r)$, runs $\Pi'.\mathsf{Dec}(\mathsf{rdk}, r)$ to obtain $(\mathsf{m}', \hat{\mathsf{m}})$, and returns $\hat{\mathsf{m}}$ if $\mathsf{m} = \mathsf{m}'$ and $\perp$ otherwise (including the case $\Pi'.\mathsf{Dec}$ returns $\perp$).

The correctness of Σ follows directly from the correctness of Π and Π'.

Remark 10 (Secure AE vs. CCA-Secure PKE). In Section 5, we argue that in the rr-PKE paradigm, the CCA security of the underlying symmetric encryption scheme is insufficient for STE-xCCA1 security (cf. Def. 23), and we instead resort to an AE scheme. However, in the asymmetric setting, CCA security and ciphertext pseudorandomness are sufficient for CIND-xCCA1 security (cf. Def. 33). This discrepancy stems from the fundamental incomparability between these two notions.

In the STE-xCCA1 security game, the goal of the adversary is to distinguish the normal mode from the anamorphic mode. Since the anamorphic decryption oracle always returns $\perp$ in the normal mode, the adversary wins as long as it can generate a valid anamorphic ciphertext—equivalently—a valid and fresh ciphertext of the underlying SE scheme. Consequently, CCA security alone is insufficient, and a stronger AE scheme is required to guarantee authenticity. While in the CIND-xCCA1 security game, the adversary already knows the scheme is working in the anamorphic mode from sdk (the public key of Π'), and its attacking goal is to tell if a *specific* ciphertext is a normal one or an anamorphic one. Therefore, the CCA security of the underlying PKE scheme Π' is sufficient for the CIND-xCCA1 security.

To additionally achieve STE-xCCA1 security, however, one must incorporate authenticity—analogous to the role of AE in the symmetric setting. In Section 7.2, we construct an asymmetric AME scheme with both CIND-xCCA1 and STE-xCCA1 from rr-PKE and the random oracle.

Theorem 32 (Security of Asymmetric AME from rrPKE). *If PKE scheme* Π *is IND-CCA-secure and randomness recovering, PKE scheme* Π' *is PR-CCA secure, then the asymmetric AME scheme* $\mathsf{AME} = (\Pi, \Sigma)$ *with the anamorphic triplet described above has CIND-xCCA1 security and SIND-nCCA security.*

[17] Similar to the symmetric setting, we can use a PR-CCA-secure PKE scheme with labels and compute r as an encryption of $\hat{\mathsf{m}}$ with m being the label. This would save space for the anamorphic plaintext.

Proof. To prove the CIND-xCCA1 security of AME, it is sufficient to argue that the adversary, given sdk (the public key of Π') and an anamorphic decryption oracle (the decryption oracle of Π') but not rdk (the secret key of Π'), it still cannot tell an encryption of $(\mathsf{m}, \hat{\mathsf{m}})$ from a random string of the same length of the ciphertext. We then construct a reduction algorithm $\mathcal{A}'$ to show that, if $\mathcal{A}$ can distinguish the experiment $\text{CIND-xCCA1}_{\Pi,\Sigma,1}(\mathcal{A})$ (the anamorphic mode) from the experiment $\text{CIND-xCCA1}_{\Pi,\Sigma,0}(\mathcal{A})$ (the normal mode), then $\mathcal{A}'$ is able to break the PR-CCA security of Π'.

$\mathcal{A}'$ simulates the experiment to $\mathcal{A}$ as follows.

- $\mathcal{A}'$ receives the public key sdk of Π' from its own challenger. It generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \Pi.\mathsf{KGen}(1^\lambda)$, and passes $(\mathsf{pk}, \mathsf{sk}, \mathsf{sdk})$ to $\mathcal{A}$.
- When $\mathcal{A}$ asks a challenge query $(\mathsf{m}^*, \hat{\mathsf{m}}^*)$, the reduction algorithm $\mathcal{A}'$ queries $(\mathsf{m}^*, \hat{\mathsf{m}}^*)$ to its own challenger and obtains r^*, which is either a random string or an encryption $\Pi'.\mathsf{Enc}(\mathsf{sdk}, (\mathsf{m}^*, \hat{\mathsf{m}}^*))$. Then $\mathcal{A}'$ uses this r^* as the randomness to generate $\mathsf{ct}^* := \Pi.\mathsf{Enc}(\mathsf{pk}, \mathsf{m}^*; r^*)$ and passes ct^* as the ciphertext to $\mathcal{A}$.
- When $\mathcal{A}$ asks an anamorphic decryption query with $\mathsf{ct} \neq \mathsf{ct}^*$, $\mathcal{A}'$ first invokes $\Pi.\mathsf{rrDec}(\mathsf{sk}, \mathsf{ct}) = (\mathsf{m}, r)$. If the decryption fails or $r = r^*$ then $\mathcal{A}'$ returns $\perp$ directly. Otherwise, $\mathcal{A}'$ queries its decryption oracle with input r and obtains the result. If the decryption result is $(\mathsf{m}', \hat{\mathsf{m}}')$ such that $\mathsf{m}' = \mathsf{m}$, then $\mathcal{A}'$ returns $\hat{\mathsf{m}}'$ to $\mathcal{A}$; in all other cases, $\mathcal{A}'$ returns $\perp$.

We now analyze the simulation by $\mathcal{A}'$. Recall that in game $\text{CIND-xCCA1}_{\Pi,\Sigma,1}(\mathcal{A})$, the adversary $\mathcal{A}$ is not allowed to ask an anamorphic decryption query with the challenge ciphertext. Since the challenge ciphertext (of Π) is totally determined by the message and the randomness, it is impossible to provide a ciphertext $\mathsf{ct} \neq \mathsf{ct}^*$ such that $\Pi.\mathsf{rrDec}(\mathsf{sk}, \mathsf{ct}) = (\mathsf{m}, r) = (\mathsf{m}^*, r^*)$. If $r = r^*$ but $\mathsf{m} \neq \mathsf{m}^*$, which means that $\mathcal{A}$ reused r^* to encrypt a distinct message m, then $\mathcal{A}'$ directly returns $\perp$ since r^* is either a random string or an encryption of $(\mathsf{m}^*, \cdot)$ so that with negligible probability r^* can be decrypted to $(\mathsf{m}, \cdot)$.

Therefore, if $\mathcal{A}'$ has access to $\text{PR-CCA}_{\Pi',1}(\mathcal{A}')$ (where r^* is a random string), then $\mathcal{A}'$ simulates $\text{CIND-xCCA1}_{\Pi,\Sigma,0}(\mathcal{A})$ for $\mathcal{A}$, and if $\mathcal{A}'$ has access to $\text{PR-CCA}_{\Pi',0}(\mathcal{A}')$ (where r^* is an encryption), then $\mathcal{A}'$ simulates $\text{CIND-xCCA1}_{\Pi,\Sigma,1}(\mathcal{A})$ for $\mathcal{A}$. By the PR-CCA security of Π', CIND-xCCA1 security of AME follows.

The SIND-nCCA security of AME is directly implied by the IND-CCA security of the underlying PKE scheme Π. This concludes the proof. □

7.2 Achieving both CIND-xCCA1 Security and STE-xCCA1 Security

The construction above does not guarantee STE-xCCA1 security, since the underlying PR-CCA-secure PKE scheme provides confidentiality but not authenticity. Specifically, after observing several ciphertexts r of Π' produced by anamorphic encryption, the dictator may be able to forge a fresh r^* that decrypts to a non-failure value. To further achieve STE-xCCA1 security, we must additionally introduce authenticity, as done via AE in Section 5. To this end, we require the PR-CCA-secure PKE Π' to also be randomness-recovering, so that the randomness used in Π' can carry the authentication information. Unlike the symmetric construction in Section 5 where AE serves both to hide $\hat{\mathsf{m}}$ and to provide authenticity, here we can simply use a hash-based message authentication code (MAC), since $\hat{\mathsf{m}}$ is already carried by the PKE scheme Π'. Moreover, in the random oracle model, we prove that the PR-CPA (not CCA) security of Π' suffices.

More precisely, let Π be an IND-CCA-secure rr-PKE scheme, Π' be a PR-CPA-secure rr-PKE scheme with randomness space $\{0,1\}^{\lambda_1+\lambda_2}$, and $\mathsf{H} : \{0,1\}^* \to \{0,1\}^{\lambda_2}$ be a hash function modeled as a random oracle. We design the anamorphic triplet Σ associated with Π as follow. The components added to further achieve STE-xCCA1 security are highlighted in gray.

- The anamorphic key generation algorithm $\mathsf{aKGen}(1^\lambda)$ returns $(\mathsf{pk}, \mathsf{sk}, \mathsf{sdk} = (\mathsf{pk}', \mathsf{k}), \mathsf{rdk} = (\mathsf{sk}', \mathsf{k}))$, where $(\mathsf{pk}, \mathsf{sk}) \leftarrow \Pi.\mathsf{KGen}(1^\lambda)$, $(\mathsf{pk}', \mathsf{sk}') \leftarrow \Pi'.\mathsf{KGen}(1^\lambda)$, and $\mathsf{k} \leftarrow \{0,1\}^\lambda$.
- The anamorphic encryption algorithm $\mathsf{aEnc}(\mathsf{pk}, \mathsf{sdk}, \mathsf{m}, \hat{\mathsf{m}})$ returns $\mathsf{ct} := \Pi.\mathsf{Enc}(\mathsf{pk}, \mathsf{m}; r)$, where $r := \Pi'.\mathsf{Enc}(\mathsf{pk}', \hat{\mathsf{m}}; \hat{r})$, $\hat{r}_1 \leftarrow \{0,1\}^{\lambda_1}$, and $\hat{r} := (\hat{r}_1 || \mathsf{H}(\mathsf{k}, \mathsf{m}, \hat{\mathsf{m}}, \hat{r}_1))$.

– The anamorphic decryption algorithm $\mathsf{aDec}(\mathsf{sk}, \mathsf{rdk}, \mathsf{ct})$ first invokes the randomness recovering algorithm of Π to get $\Pi.\mathsf{rrDec}(\mathsf{sk}, \mathsf{ct}) = (\mathsf{m}, r)$, then runs $\Pi'.\mathsf{rrDec}(\mathsf{sk}', r)$ to obtain $(\hat{\mathsf{m}}, (\hat{r}_1, \hat{r}_2))$. It returns $\hat{\mathsf{m}}$ if $\hat{r}_2 = \mathsf{H}(\mathsf{k}, \mathsf{m}, \hat{\mathsf{m}}, \hat{r}_1)$ and $\perp$ otherwise.

Theorem 33 (Security of Asymmetric AME from rrPKE and RO). *If PKE scheme Π is IND-CCA-secure and randomness recovering, PKE scheme Π' is PR-CPA secure and randomness recovering, and H is a random oracle, then the asymmetric AME scheme $\mathsf{AME} = (\Pi, \Sigma)$ with the anamorphic triplet described above has STE-xCCA1 security, CIND-xCCA1 security and SIND-nCCA security.*

Proof. We first prove STE-xCCA1 security. Note that pk and sk distribute identically in both $\text{STE-xCCA1}_{\Pi,\Sigma,1}(\mathcal{A})$ (the anamorphic mode) and $\text{STE-xCCA1}_{\Pi,\Sigma,0}(\mathcal{A})$ (the normal mode). To prove the indistinguishability of these two games, we introduce a hybrid game $\text{Hybrid}_{\Pi,\Sigma}(\mathcal{A})$ which is the same as $\text{STE-xCCA1}_{\Pi,\Sigma,1}(\mathcal{A})$, except that all anamorphic decryption queries from $\mathcal{A}$ output failure indicator $\perp$.

We analyze the difference between $\text{STE-xCCA1}_{\Pi,\Sigma,1}(\mathcal{A})$ and $\text{Hybrid}_{\Pi,\Sigma}(\mathcal{A})$ as follows. For simplicity we consider single challenge setting here, and the multiple challenge setting follows via the standard hybrid argument. Let $(\mathsf{m}^*, \hat{\mathsf{m}}^*)$ be the challenge query by $\mathcal{A}$, and $\mathsf{ct}^* := \Pi.\mathsf{Enc}(\mathsf{pk}, \mathsf{m}^*; r^*)$ be the corresponding challenge ciphertext, where $r^* := \Pi'.\mathsf{Enc}(\mathsf{pk}', \hat{\mathsf{m}}^*; \hat{r}^*)$, $\hat{r}^* = (\hat{r}_1^* || \hat{r}_2^*)$, $\hat{r}_1^* \leftarrow \{0,1\}^{\lambda_1}$, and $\hat{r}_2^* = \mathsf{H}(\mathsf{k}, \mathsf{m}^*, \hat{\mathsf{m}}^*, \hat{r}_1^*)$. If the adversary $\mathcal{A}$ queries a ciphertext $\mathsf{ct} \neq \mathsf{ct}^*$ such that $\mathsf{aDec}(\mathsf{sk}, \mathsf{rdk}, \mathsf{ct}) \neq \perp$, then $\mathcal{A}$ is able to distinguish $\text{STE-xCCA1}_{\Pi,\Sigma,1}(\mathcal{A})$ from $\text{Hybrid}_{\Pi,\Sigma}(\mathcal{A})$. Assume $\Pi.\mathsf{rrDec}(\mathsf{sk}, \mathsf{ct}) = (\mathsf{m}, r)$ and $\Pi'.\mathsf{rrDec}(\mathsf{sk}', r) = (\hat{\mathsf{m}}, (\hat{r}_1 || \hat{r}_2))$, we further discuss in the following cases.

– Case 1. $\mathsf{m} = \mathsf{m}^*$ and $r = r^*$. This implies that $\mathsf{ct} = \mathsf{ct}^*$ and the oracle will return $\perp$ directly.
– Case 2. $\mathsf{m} \neq \mathsf{m}^*$ but $r = r^*$. Recall that $\Pi'.\mathsf{rrDec}(\mathsf{sk}', r) = (\hat{\mathsf{m}}^*, (\hat{r}_1^* || \hat{r}_2^*))$ and $\hat{r}_2^* = \mathsf{H}(\mathsf{k}, \mathsf{m}^*, \hat{\mathsf{m}}^*, \hat{r}_1^*)$. If the anamorphic decryption oracle does not return $\perp$, then $\hat{r}_2^* = \mathsf{H}(\mathsf{k}, \mathsf{m}^*, \hat{\mathsf{m}}^*, \hat{r}_1^*) = \mathsf{H}(\mathsf{k}, \mathsf{m}, \hat{\mathsf{m}}^*, \hat{r}_1^*)$. Since $\mathsf{m} \neq \mathsf{m}^*$, this leads to a collision for random oracle H, which happens with a negligible probability.
– Case 3. $r \neq r^*$. Assume the decryption of Π' does not fail, and $\Pi'.\mathsf{rrDec}(\mathsf{sk}', r) = (\hat{\mathsf{m}}, (\hat{r}_1 || \hat{r}_2))$.
  • Case 3.1. $(\mathsf{m}, \hat{\mathsf{m}}, \hat{r}_1) = (\mathsf{m}^*, \hat{\mathsf{m}}^*, \hat{r}_1^*)$. This implies that $r = r^*$, since $\hat{r}_2$ is totally determined by k, m, $\hat{\mathsf{m}}$, and $\hat{r}_1$. Therefore, $\mathsf{ct} = \mathsf{ct}^*$ and the oracle will return $\perp$.
  • Case 3.2. $(\mathsf{m}, \hat{\mathsf{m}}, \hat{r}_1) \neq (\mathsf{m}^*, \hat{\mathsf{m}}^*, \hat{r}_1^*)$. If the oracle does not return $\perp$, then $\mathsf{H}(\mathsf{k}, \mathsf{m}, \hat{\mathsf{m}}, \hat{r}_1) = \hat{r}_2$. Since H is a random oracle, the information of k is totally hidden to the adversary who has even observed $\hat{r}_2^* = \mathsf{H}(\mathsf{k}, \mathsf{m}^*, \hat{\mathsf{m}}^*, \hat{r}_1^*)$. Therefore, $\mathsf{H}(\mathsf{k}, \mathsf{m}, \hat{\mathsf{m}}, \hat{r}_1) = \hat{r}_2$ happens with a negligible probability.

As a result, $\text{STE-xCCA1}_{\Pi,\Sigma,1}(\mathcal{A})$ and $\text{Hybrid}_{\Pi,\Sigma}(\mathcal{A})$ are indistinguishable to the adversary.

Then we analyze the difference between $\text{Hybrid}_{\Pi,\Sigma}(\mathcal{A})$ and $\text{STE-xCCA1}_{\Pi,\Sigma,0}(\mathcal{A})$. Now both anamorphic decryption oracles return $\perp$ all the time, and the only difference between these two games is the randomness r, which is randomly sampled in $\text{STE-xCCA1}_{\Pi,\Sigma,0}(\mathcal{A})$, and $r := \Pi'.\mathsf{Enc}(\mathsf{pk}', \hat{\mathsf{m}}; (\hat{r}_1 || \mathsf{H}(\mathsf{k}, \mathsf{m}, \hat{\mathsf{m}}, \hat{r}_1)))$ in $\text{Hybrid}_{\Pi,\Sigma}(\mathcal{A})$. Due to the PR-CPA security of Π' and that H is a random oracle, these two games are indistinguishable to the adversary $\mathcal{A}$. This finishes the proof of STE-xCCA1 security.

We then prove the CIND-xCCA1 security in the single challenge setting, and the security in the multi-challenge setting follows by the hybrid argument. Recall that in the $\text{CIND-xCCA1}_{\Pi,\Sigma,1}(\mathcal{A})$ game (the anamorphic mode), the challenge ciphertext ct^* is computed as $\mathsf{ct}^* \leftarrow \Pi.\mathsf{Enc}(\mathsf{pk}, \mathsf{m}^*; r^*)$, where $r^* := \Pi'.\mathsf{Enc}(\mathsf{pk}', \hat{\mathsf{m}}^*; \hat{r}^*)$, $\hat{r}^* = (\hat{r}_1^* || \hat{r}_2^*)$, $\hat{r}_1^* \leftarrow \{0,1\}^{\lambda_1}$, and $\hat{r}_2^* = \mathsf{H}(\mathsf{k}, \mathsf{m}^*, \hat{\mathsf{m}}^*, \hat{r}_1^*)$.

We prove via two hybrid games. Define $\text{Hybrid1}_{\Pi,\Sigma}(\mathcal{A})$ the same as $\text{CIND-xCCA1}_{\Pi,\Sigma,1}(\mathcal{A})$ except that the anamorphic decryption oracle returns $\perp$ for any query ct such that $\Pi.\mathsf{rrDec}(\mathsf{sk}, \mathsf{ct}) = (\mathsf{m}, r)$ and $r = r^*$. Recall that $\mathcal{A}$ is not allowed to ask an anamorphic decryption query on $\mathsf{ct} = \mathsf{ct}^* = \Pi.\mathsf{Enc}(\mathsf{pk}, \mathsf{m}^*; r^*)$. Assume $\Pi'.\mathsf{rrDec}(\mathsf{sk}', r) = (\hat{\mathsf{m}}', (\hat{r}_1 || \hat{r}_2)) \neq \perp$. If $r = r^*$ and the anamorphic decryption oracle does not return $\perp$, then we have $\mathsf{m} \neq \mathsf{m}^*$, and

$$\mathsf{H}(\mathsf{k}, \mathsf{m}, \hat{\mathsf{m}}^*, \hat{r}_1^*) = \hat{r}_2^* = \mathsf{H}(\mathsf{k}, \mathsf{m}^*, \hat{\mathsf{m}}^*, \hat{r}_1^*),$$

which implies a collision on H. Since H is a random oracle, this happens with negligible probability, and consequently $\text{Hybrid1}_{\Pi,\Sigma}(\mathcal{A})$ is indistinguishable from $\text{CIND-xCCA1}_{\Pi,\Sigma,1}(\mathcal{A})$ by $\mathcal{A}$.

Define $\text{Hybrid2}_{\Pi,\Sigma}(\mathcal{A})$ the same as $\text{Hybrid1}_{\Pi,\Sigma}(\mathcal{A})$ except that upon receiving an anamorphic decryption query ct such that $\Pi.\mathsf{rrDec}(\mathsf{sk}, \mathsf{ct}) = (\mathsf{m}, r)$ and $\Pi'.\mathsf{rrDec}(\mathsf{sk}', r) = (\hat{\mathsf{m}}, (\hat{r}_1 || \hat{r}_2))$, if there is no query

$\mathsf{H}(\mathsf{k}, \mathsf{m}, \hat{\mathsf{m}}, \hat{r}_1) = \hat{r}_2$, then the oracle returns $\perp$. It is easy to see that Hybrid2$_{\Pi,\Sigma}(\mathcal{A})$ and Hybrid1$_{\Pi,\Sigma}(\mathcal{A})$ are identical unless $\mathsf{H}(\mathsf{k}, \mathsf{m}, \hat{\mathsf{m}}, \hat{r}_1)$ equals to a fixed value $\hat{r}_2$, which happens with probability $\frac{1}{2^{\lambda_2}}$.

Note that the difference between Hybrid2$_{\Pi,\Sigma}(\mathcal{A})$ and CIND-xCCA1$_{\Pi,\Sigma,0}(\mathcal{A})$ (the normal mode) is how to generate randomness r^* used in the challenge ciphertext. Namely, we have $r^* := \Pi'.\mathsf{Enc}(\mathsf{pk}', \hat{\mathsf{m}}^*; \hat{r}^*)$, where $\hat{r}^* \leftarrow \{0,1\}^{\lambda_1+\lambda_2}$ in Hybrid2$_{\Pi,\Sigma}(\mathcal{A})$, and r^* is uniformly sampled from the randomness space of Π in CIND-xCCA1$_{\Pi,\Sigma,0}(\mathcal{A})$. We now construct a reduction algorithm $\mathcal{A}'$ to show that, if $\mathcal{A}$ can distinguish game Hybrid2$_{\Pi,\Sigma}(\mathcal{A})$ from CIND-xCCA1$_{\Pi,\Sigma,0}(\mathcal{A})$, then $\mathcal{A}'$ is able to break the PR-CPA security of Π'. The reduction proceeds similarly to that in the proof of Theorem 32, and is detailed as follows.

- $\mathcal{A}'$ receives the public key pk' of Π' from its own challenger. It generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \Pi.\mathsf{KGen}(1^\lambda)$, samples $\mathsf{k} \leftarrow \{0,1\}^\lambda$, and passes $(\mathsf{pk}, \mathsf{sk}, \mathsf{sdk} = (\mathsf{pk}', \mathsf{k}))$ to $\mathcal{A}$.
- $\mathcal{A}'$ maintain a list L_{H} to simulate the random oracle H for $\mathcal{A}$.
- When $\mathcal{A}$ asks an anamorphic decryption query ct, $\mathcal{A}'$ uses sk to do randomness recovering decryption $\Pi.\mathsf{rrDec}(\mathsf{sk}, \mathsf{ct}) = (\mathsf{m}, r)$. Then, if there exists a unique item $(\mathsf{k}, \mathsf{m}, \hat{\mathsf{m}}', \hat{r}'_1, \hat{r}'_2) \in L_{\mathsf{H}}$ such that $r = \Pi'.\mathsf{Enc}(\mathsf{pk}', \hat{\mathsf{m}}'; (\hat{r}'_1 || \hat{r}'_2))$, then the oracle returns $\hat{\mathsf{m}}'$. Otherwise, it returns $\perp$.
- When $\mathcal{A}$ asks the challenge query $(\mathsf{m}^*, \hat{\mathsf{m}}^*)$, the reduction algorithm $\mathcal{A}'$ queries $\hat{\mathsf{m}}^*$ to its own challenger and obtains r^*, which is either a random string or an encryption $\Pi'.\mathsf{Enc}(\mathsf{pk}', \hat{\mathsf{m}}^*)$. Then $\mathcal{A}'$ uses this r^* as the randomness to generate $\mathsf{ct}^* := \Pi.\mathsf{Enc}(\mathsf{pk}, \mathsf{m}^*; r^*)$, and passes ct^* as the challenge ciphertext to $\mathcal{A}$.
- The anamorphic decryption query after the challenge query is simulated in the same way, except that it returns $\perp$ immediately if $r = r^*$.

We first analyze the anamorphic decryption oracle simulated by $\mathcal{A}'$. Due to the (perfect) correctness of Π', if the decryption does not fail, then there exists a unique message-randomness pair $(\hat{\mathsf{m}}', (\hat{r}'_1 || \hat{r}'_2))$ such that $r = \Pi'.\mathsf{Enc}(\mathsf{pk}', \hat{\mathsf{m}}'; (\hat{r}'_1 || \hat{r}'_2))$. If the anamorphic decryption oracle does not return $\perp$, then by the rule introduced in Hybrid2$_{\Pi,\Sigma}(\mathcal{A})$, $\mathcal{A}$ must have queried $\mathsf{H}(\mathsf{k}, \mathsf{m}, \hat{\mathsf{m}}', \hat{r}'_1) = \hat{r}'_2$, which implies the perfect simulation of the anamorphic decryption oracle by $\mathcal{A}'$.

If $\mathcal{A}'$ has access to PR-CPA$_{\Pi',1}(\mathcal{A}')$ (where r^* is a random string), then $\mathcal{A}'$ simulates CIND-xCCA1$_{\Pi,\Sigma,0}(\mathcal{A})$ for $\mathcal{A}$, and if $\mathcal{A}'$ has access to PR-CPA$_{\Pi',0}(\mathcal{A}')$ (where r^* is an encryption), then $\mathcal{A}'$ simulates Hybrid2$_{\Pi,\Sigma}(\mathcal{A})$ for $\mathcal{A}$. By the PR-CPA security of Π', CIND-xCCA1 security of AME follows.

Finally, the SIND-nCCA security of AME is directly implied by the IND-CCA security of the underlying PKE scheme Π. This concludes the proof. □

Acknowledgments. We would like to thank the anonymous reviewers for their insightful feedback. Wonseok Choi was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2026-25494383). Vassilis Zikas was supported in part by AnalytiXIN, JPMorgan Chase, and Sunday Group, Inc.

References

ABG+26. Gennaro Avitabile, Vincenzo Botta, Emanuele Giunta, Marcin Mielniczuk, and Francesco Migliaro. The malice of ELFs: Practical anamorphic-resistant encryption without random oracles. In *Eurocrypt 2026*, 2026.

BDPR98. Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. pages 26–45, 1998.

BF18. Manuel Barbosa and Pooya Farshim. Indifferentiable authenticated encryption. pages 187–220, 2018.

BGH+24. Fabio Banfi, Konstantin Gegier, Martin Hirt, Ueli Maurer, and Guilherme Rito. Anamorphic encryption, revisited. pages 3–32, 2024.

BHR12. Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. Foundations of garbled circuits. pages 784–796, 2012.

BKP14. Olivier Blazy, Eike Kiltz, and Jiaxin Pan. (Hierarchical) identity-based encryption from affine message authentication. pages 408–425, 2014.

BN08. Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. 21(4):469–491, October 2008.

BPRS25. Shalini Banerjee, Tapas Pal, Andy Rupp, and Daniel Slamanig. Simple public key anamorphic encryption and signature using multi-message extensions. Cryptology ePrint Archive, Report 2025/370, 2025.

BR00. Mihir Bellare and Phillip Rogaway. Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. pages 317–330, 2000.

CCGM25. Davide Carnemolla, Dario Catalano, Emanuele Giunta, and Francesco Migliaro. Anamorphic resistant encryption: the good, the bad and the ugly. pages 472–503, 2025.

CDD+24. Yu Long Chen, Michael Davidson, Morris Dworkin, Jinkeon Kang, John Kelsey, Yu Sasaki, Meltem Sönmez Turan, Donghoon Chang, Nicky Mouha, MD Largo, et al. Proposal of requirements for an accordion mode. In *Discussion Draft for the NIST Accordion Mode Workshop 2024*, 2024.

CGM24a. Dario Catalano, Emanuele Giunta, and Francesco Migliaro. Anamorphic encryption: New constructions and homomorphic realizations. pages 33–62, 2024.

CGM24b. Dario Catalano, Emanuele Giunta, and Francesco Migliaro. Limits of black-box anamorphic encryption. pages 352–383, 2024.

CGM25. Dario Catalano, Emanuele Giunta, and Francesco Migliaro. Generic anamorphic encryption, revisited: New limitations and constructions. pages 275–303, 2025.

CPS08. Jean-Sébastien Coron, Jacques Patarin, and Yannick Seurin. The random oracle model and the ideal cipher model are equivalent. pages 1–20, 2008.

CS03. Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. 33(1):167–226, 2003.

DG25. Yevgeniy Dodis and Eli Goldin. Anamorphic-resistant encryption; or why the encryption debate is still alive. pages 440–471, 2025.

DL25a. Amit Deo and Benoît Libert. Anamorphic signatures with dictator and recipient unforgeability for long messages. pages 370–401, 2025.

DL25b. Amit Deo and Benoît Libert. Fully asymmetric anamorphic homomorphic encryption from LWE. Cryptology ePrint Archive, Report 2025/328, 2025.

DL26. Amit Deo and Benoit Libert. Fully asymmetric anamorphic homomorphic encryption from LWE. In *Eurocrypt 2026*, 2026.

FO99. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. pages 537–554, 1999.

FO13. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. 26(1):80–101, January 2013.

Fro94. A Michael Froomkin. Metaphor is the key: cryptography, the clipper chip, and the constitution. *U. Pa. L. Rev.*, 143:709, 1994.

HKT11. Thomas Holenstein, Robin Künzler, and Stefano Tessaro. The equivalence of the random oracle model and the ideal cipher model, revisited. pages 89–98, 2011.

HLQG18. Shuai Han, Shengli Liu, Baodong Qin, and Dawu Gu. Tightly CCA-secure identity-based encryption with ciphertext pseudorandomness. 86(3):517–554, 2018.

HTT18. Viet Tung Hoang, Stefano Tessaro, and Aishwarya Thiruvengadam. The multi-user security of GCM, revisited: Tight bounds for nonce randomization. pages 1429–1440, 2018.

JS24. Joseph Jaeger and Roy Stracovsky. Dictators? Friends? Forgers. - breaking and fixing unforgeability definitions for anamorphic signature schemes. pages 105–137, 2024.

KPP+23a. Miroslaw Kutylowski, Giuseppe Persiano, Duong Hieu Phan, Moti Yung, and Marcin Zawada. Anamorphic signatures: Secrecy from a dictator who only permits authentication! pages 759–790, 2023.

KPP+23b. 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. 2023(4):170–183, October 2023.

KW19. Venkata Koppula and Brent Waters. Realizing chosen ciphertext security generically in attribute-based encryption and predicate encryption. pages 671–700, 2019.

LLHG23. Xiangyu Liu, Shengli Liu, Shuai Han, and Dawu Gu. Fine-grained verifier NIZK and its applications. pages 482–511, 2023.

Möl04. Bodo Möller. A public-key encryption scheme with pseudo-random ciphertexts. pages 335–351, 2004.

MX23. Varun Maram and Keita Xagawa. Post-quantum anonymity of Kyber. pages 3–35, 2023.

PPY22. Giuseppe Persiano, Duong Hieu Phan, and Moti Yung. Anamorphic encryption: Private communication against a dictator. pages 34–63, 2022.

PPY24. Giuseppe Persiano, Duong Hieu Phan, and Moti Yung. Public-key anamorphism in (CCA-secure) public-key encryption and beyond. pages 422–455, 2024.

RSS11. Thomas Ristenpart, Hovav Shacham, and Thomas Shrimpton. Careful with composition: Limitations of the indifferentiability framework. pages 487–506, 2011.

WHL24. Weihao Wang, Shuai Han, and Shengli Liu. Anamorphic authenticated key exchange: Double key distribution under surveillance. pages 168–200, 2024.

Woo25. Emma Woollacott. UK Orders Apple To Open Up Encrypted Messages Worldwide. https://www.forbes.com/sites/emmawoollacott/2025/02/07/uk-orders-apple-to-open-up-encrypted-messages-worldwide/, 2025.

A Additional Results in the CPA Setting

In this section, we discuss not well-defined and impossible definitions in the CPA setting.

A.1 Not Well-Defined Definitions

We first identify the settings that are not well-defined and explain why they are such. Broadly, these cases occur whenever the challenge oracle asks the adversary to distinguish anamorphic ciphertexts, yet only normal key generation was run so the double keys necessary for anamorphic encryption do not exist.

The three not well-defined definitions are STE-aCPA1, STE-aCPA2, and STE-aCPA3 (refer to Fig. 7 and Table 6). In all of these definitions, the key generation setting is dictator stealthiness STE, meaning that normal key generation KGen is run when $b = 0$ and anamorphic key generation aKGen is run when $b = 1$. In more detail and considering symmetric anamorphic encryption for simplicity, when $b = 0$, the game generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$, and when $b = 1$, the game generates $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{aKGen}(1^\lambda)$. Now, in the aCPA1, aCPA2, and aCPA3 challenge oracle settings, the challenge oracle always computes $\mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}})$ where $\mathsf{m}, \hat{\mathsf{m}}$ are dependent on b and the adversary's left-or-right query. Note, however, that when $b = 0$, the double key dk has not been generated as only normal key generation was run, hence aEnc cannot be computed and the definitions are not well-defined. This reasoning extends to asymmetric anamorphic encryption, because the sending double key sdk is not generated when $b = 0$ yet is needed to compute $\mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{sdk}, \mathsf{m}, \hat{\mathsf{m}})$.

A.2 Impossible Definitions

Impossible Definitions. We next identify the settings that are impossible (for schemes that achieve correctness) and explain why they are such. Broadly, these cases occur whenever the challenge oracle takes in a particular left-or-right input that the adversary can decrypt given the key they obtain. Below, we will assume for simplicity that we require schemes to be perfectly correct but the reasoning extends to schemes that achieve imperfect correctness. Recall that we require an anamorphic encryption scheme to satisfy effectively four correctness properties, which we informally outline and give placeholder names to below (see also Def. 9 and Def. 10). These names will help us in our upcoming discussion.

- **Standard:** for $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$ and $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$ for arbitrary m, we have that $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \mathsf{m}$. This is standard encryption correctness.
- **Type 1:** for $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{aKGen}(1^\lambda)$ and $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$ for arbitrary m, we have that $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \mathsf{m}$.
- **Type 2:** for $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{aKGen}(1^\lambda)$ and $\mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}})$ for arbitrary m and $\hat{\mathsf{m}}$, we have that $\mathsf{aDec}(\mathsf{sk}, \mathsf{dk}, \mathsf{ct}) = \hat{\mathsf{m}}$.
- **Type 3:** for $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{aKGen}(1^\lambda)$ and $\mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}})$ for arbitrary m and $\hat{\mathsf{m}}$, we have that $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \mathsf{m}$.

We begin with STE-xCPA2 and STE-nCPA. Here, the key generation setting is dictator stealthiness STE, meaning that either normal or anamorphic key generation is run depending on b. In more detail and considering symmetric anamorphic encryption, when $b = 0$, the game generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$, and when $b = 1$, the game generates $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{aKGen}(1^\lambda)$. The dictator receives $(\mathsf{pk}, \mathsf{sk})$. Now for both the xCPA2 and nCPA challenge oracle settings, the adversary gets to provide (among any other inputs) two normal messages $\mathsf{m}_0, \mathsf{m}_1$ that are encrypted when $b = 0$ and $b = 1$ respectively. An adversary can simply decrypt any challenge ciphertext ct using sk to recover m_b. While the attack is conceptually simple, there is nuance in which correctness properties the adversary leverages, which depends on the challenge oracle setting. We first consider xCPA2, where, given an adversary's query $(\mathsf{m}_0, \mathsf{m}_1, \hat{\mathsf{m}})$, normal encryption $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_0)$ is run when $b = 0$ and anamorphic encryption $\mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}_1, \hat{\mathsf{m}})$ is run when $b = 1$. When $b = 0$, standard correctness implies that $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_0)) = \mathsf{m}_0$ and when $b = 1$, Type 3 correctness implies that $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \mathsf{Dec}(\mathsf{sk}, \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}_1, \hat{\mathsf{m}})) = \mathsf{m}_1$. In nCPA, the adversary queries $(\mathsf{m}_0, \mathsf{m}_1)$ and receives a normal encryption $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_b)$. When $b = 0$, standard correctness

implies that $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_0)) = \mathsf{m}_0$ and when $b = 1$, Type 1 correctness implies that $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_1)) = \mathsf{m}_1$. Observe that these attacks also apply to asymmetric anamorphic encryption, because the adversary only uses sk (and so it does not matter whether there is a shared double key or separate sending and receiving double keys).

We now consider the DIND-xCPA2, DIND-nCPA, DIND-aCPA1, and DIND-aCPA3 definitions. Similar to the definitions just considered, the impossibility of achieving these definitions for correct anamorphic encryption schemes follows from the fact that the key generation setting captures an adversarial dictator who obtains the secret key sk, and the challenge oracle settings allow this adversary to pass in differing left-or-right messages that the dictator can recover using sk. In more detail, the key generation setting is dictator indistinguishability DIND, so the game generates $(\mathsf{pk}, \mathsf{sk}, \mathsf{dk}) \leftarrow \mathsf{aKGen}(1^\lambda)$ and gives $(\mathsf{pk}, \mathsf{sk})$ to the adversary. In the xCPA2 challenge oracle setting, recall that the adversary queries $(\mathsf{m}_0, \mathsf{m}_1, \hat{\mathsf{m}})$ to the challenge oracle and receives a normal encryption $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_0)$ when $b = 0$ and anamorphic encryption $\mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}_1, \hat{\mathsf{m}})$ when $b = 1$. Note that Type 1 correctness implies that $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_0)) = \mathsf{m}_0$ and Type 3 correctness implies that $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \mathsf{Dec}(\mathsf{sk}, \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}_1, \hat{\mathsf{m}})) = \mathsf{m}_1$, hence allowing the adversary to decrypt regardless the value of b. In nCPA, the adversary queries $(\mathsf{m}_0, \mathsf{m}_1)$ and receives a normal encryption $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_b)$. Here, Type 1 correctness applies for both values of b and $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_b)) = \mathsf{m}_b$. Finally, for aCPA1 and aCPA3, the adversary queries $(\mathsf{m}_0, \mathsf{m}_1, \dots)$ to the challenge oracle, where $\dots$ may be a fixed or left-or-right anamorphic messages, and they receive $\mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}_b, \dots)$. Here, Type 3 correctness applies for both values of b and $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \mathsf{Dec}(\mathsf{sk}, \mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}_b, \dots)) = \mathsf{m}_b$. Like before, these attacks additionally apply to asymmetric anamorphic encryption, because the adversary only uses sk (and so it does not matter whether there is a shared double key or separate sending and receiving double keys). To finish, we note that the aCPA2 setting is not impossible, because the normal message is fixed.

Finally, we consider the CIND-xCPA2, CIND-nCPA, CIND-aCPA1, and CIND-aCPA3 definitions, which only apply to asymmetric anamorphic encryption. The impossibilities follow essentially similarly from the attacks on DIND-xCPA2, DIND-nCPA, DIND-aCPA1, and DIND-aCPA3, but we detail them for completeness. Here, the key generation setting is colluder indistinguishability CIND, so the game generates $(\mathsf{pk}, \mathsf{sk}, \mathsf{sdk}, \mathsf{rdk}) \leftarrow \mathsf{aKGen}(1^\lambda)$ and gives $(\mathsf{pk}, \mathsf{sk}, \mathsf{sdk})$ to the adversary. In the xCPA2 challenge oracle setting, the adversary queries $(\mathsf{m}_0, \mathsf{m}_1, \hat{\mathsf{m}})$ to the challenge oracle and receives a normal encryption $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_0)$ when $b = 0$ and anamorphic encryption $\mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{sdk}, \mathsf{m}_1, \hat{\mathsf{m}})$ when $b = 1$. Note that Type 1 correctness implies that $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_0)) = \mathsf{m}_0$; furthermore Type 3 correctness implies that $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \mathsf{Dec}(\mathsf{sk}, \mathsf{aEnc}(\mathsf{pk}, \mathsf{sdk}, \mathsf{m}_1, \hat{\mathsf{m}})) = \mathsf{m}_1$, hence allowing the adversary to decrypt regardless the value of b. In nCPA, the adversary queries $(\mathsf{m}_0, \mathsf{m}_1)$ and receives a normal encryption $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_b)$. Here, Type 1 correctness always applies so $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_b)) = \mathsf{m}_b$. Lastly, for aCPA1 and aCPA3, the adversary queries $(\mathsf{m}_0, \mathsf{m}_1, \dots)$ to the challenge oracle, where $\dots$ may be a fixed or left-or-right anamorphic messages, and they receive $\mathsf{ct} \leftarrow \mathsf{aEnc}(\mathsf{pk}, \mathsf{sdk}, \mathsf{m}_b, \dots)$. Here, Type 3 correctness applies for both values of b and $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \mathsf{Dec}(\mathsf{sk}, \mathsf{aEnc}(\mathsf{pk}, \mathsf{sdk}, \mathsf{m}_b, \dots)) = \mathsf{m}_b$.

Note that we already ruled out certain key generation settings as impossible (marked in Table 4), so they are not included in the analysis above. Specifically, we do not consider an adversary who gets the secret key sk when only KGen is ever run, and we do not consider colluder indistinguishability for symmetric anamorphic encryption. In these cases, the adversary receives all keys possible and can thus simulate any party it wants. While the first two cases are simply the insecurity of normal public-key encryption under an adversary who obtains the secret key, the last case has the explicit attacks in the xCPA1 and aCPA2 challenge oracle settings where the adversary can use dk to decrypt anamorphic ciphertexts to distinguish left-or-right anamorphic messages. This kind of attack relies on Type 2 correctness.

B Related Work and Discussions

B.1 Related Work

A recent line of research has explored anamorphic cryptography, including new constructions and primitives [BPRS25,BGH+24,CGM24a,DL25b,KPP+23a,KPP+23b,PPY22,PPY24,WHL24], (feasibility and) impossi-

bility results [CGM24b,CGM25,DG25], and improved anamorphic definitions over those considered in prior work [BGH+24,CGM24a,JS24,KPP+23b]. Our work fully explores this last line of research.

Below, we note that two works concurrent to this work also made observations regarding the insufficiency of anamorphic robustness [BGH+24] under a "dictator."

Comparison with Concurrent Work [JS24]. In the concurrent work [JS24], Jaeger and Stracovsky propose new security notions for anamorphic signatures (and encryption in the full version). Our work is primarily focused on enumerating and exploring relations between security notions, proposing constructions as feasibility results or counterexamples along the way. By comparison, [JS24] mainly considers which of the definitions are satisfied by schemes from the literature on anamorphic cryptography (we do so in Table 3), and conduct only a preliminary investigation of relations between security notions. In the full version of their work, they map their results to the encryption setting, culminating in their proof of the relation IND-CAMA $\wedge$ DUF-CASA $\Rightarrow$ IND-CASA, whereas in this work we show STE-xCPA1 $\wedge$ ROB-CT+ $\Rightarrow$ STE-xCCA1.

The authors note that it is not philosophically clear how to define STE-xCCA1. Their reasoning is as follows: it is not clear how the decryption oracle should be defined in the 'normal' world when there is no double key. By considering AME schemes with explicit rejection in decryption, we believe that our definition (Def. 23) is appropriate.

By contrast to our work, the authors consider *stateful* AME schemes, i.e., where anamorphic encryption algorithms (and signatures in their case) additionally take a state as input which is then updated and output. They however do not consider asymmetric AME as we do.

Their notion of RUF-CAMA security for signatures roughly corresponds to ROB-CT+ security, except that the adversary is given the double key dk. For encryption, this does not make sense (nor does ROB-PT security), since anamorphic encryption takes the double key as input, and thus the adversary can trivially forge a ciphertext and win. The authors do not consider additional security notions where the double key is leaked to the adversary (security against the anamorphic sender); we consider multiple.

Comparison with Concurrent Work [DG25]. Another concurrent work by Dodis and Goldin [DG25] also addresses robustness against *adversarial* ciphertexts generated by the dictator. Their notion of strong unforgeability essentially corresponds to our ROB-CT+, which serves as a necessary condition for achieving CCA security.

B.2 Further Discussion

Security in the Semi-Adaptive Setting. Recently, Catalano, Giunta, and Migliaro [CGM25] introduced so-called semi-adaptive security, where the adversary is restricted to perform all oracle queries before obtaining sk. The semi-adaptive notion is proposed to circumvent the impossibility result of black-box constructing AME (from rejection sampling [PPY22]) in order to provide some positive results. We emphasize that the semi-adaptive security in [CGM25] is strictly weaker than the basic STE-xCPA1 notion considered in this paper (and also all other works on AME [PPY22,CGM24a,BGH+24,KPP+23b,PPY24,WHL24]).

All notions in our work consider the adaptive setting, where the adversary can query oracles adaptively after observing sk and/or dk. Additionally, all the (non-)implication relationships proven in this work also hold in the semi-adaptive setting, due to the independence of challenge queries and sk/dk in the proofs.

Issues in Sharing the Double Key Privately. A basic assumption in anamorphic encryption is that the double key (or the sender double key) is privately shared between the anamorphic sender and the receiver. The issue of sharing the double key secretly was recently studied by Wang et al. in [WHL24], where a so-called anamorphic authenticated key exchange protocol was introduced to allow two parties to assign a double key under the surveillance of the dictator. Additionally, a public-key anamorphic scheme [CGM24b] can also facilitate the distribution of a double key, since the anamorphic sender is able to do anamorphic encryption with only the public key. Therefore, a fresh double key can be carried on as an anamorphic message and transmitted to the receiver secretly.

Issues in Fake Anamorphic Senders. As we discussed in the introduction, the CIND-xCPA1 and CIND-xCCA1 security notions are motivated by the scenario in which the anamorphic sender and the dictator collude. One may raise the following question:

What if one can generate a fake *sender double key to impersonate the anamorphic sender? Could it be harmful if the one hands over this fake key to the dictator?*

This is actually a desirable property since it means that the dictator cannot distinguish an anamorphic ciphertext and a normal ciphertext even in the presence of sdk and a fake one sdk', respectively. If it is hard to link a fake key and a ciphertext, whenever sdk' is revealed to the dictator, it might imply the ciphertext is indeed anamorphic and the dictator may know this. But if it is easy to link a ciphertext with a fake key in general, then the dictator cannot know whether a given ciphertext-key pair is correct or not. Hence, if a scheme has this property, the anamorphic sender and receiver enjoy one more layer of barrier from the dictatorship; even if sdk' leaks, they can claim that the key is fake and they are innocent.

Sender-Anamorphic Encryption. Persiano et al. [PPY22] proposed sender-anamorphic encryption, which allows the sender to privately send a duplicate message out when it is forced to send a "fake" message to some specified receiver. In this work we focus on the study of receiver-AME. An interesting question would be to perform a similar classification and study of sender-AME.

C Motivations for Additional Security Considerations

An encryption scheme (either a normal encryption scheme or an anamorphic triplet) is deployed by the receiver who samples the public key, secret key, and possibly a double key, and wishes to guarantee confidentiality of messages sent to them. Therefore, as shown in Fig. 31, AME involves four distinct parties: the receiver, the dictator, the anamorphic sender(s), and the eavesdroppers. Security against eavesdroppers is addressed by standard IND-CPA (or IND-CCA) security, while security against the dictator was first formalized in [PPY22] and subsequently explored in [BGH+24,CGM24a,KPP+23b,PPY24,CGM25,BPRS25,DL25b,DG25].

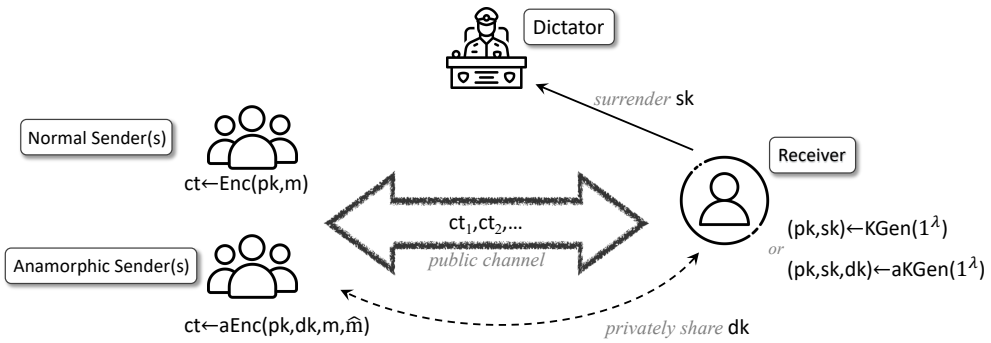


Fig. 31. The workflow of a (symmetric) AME scheme.

Anamorphic Sender: Trusted Ally, or Potential Attacker. Should the receiver allow anamorphic senders to read *normal* messages inside of ciphertexts? We argue that they should *not* unless they also have access to the secret key. Consider a concrete example, the ElGamal anamorphic triplet of Kutylowski et al. [KPP+23b], where the double key dk consists of a symmetric key K *and* the secret key x of public key $X = g^x$. That is, sharing the double key poses a significant threat to the security of the receiver, since the anamorphic sender holding dk is able to read all (normal) messages and potential anamorphic messages initially sent to the receiver. This raises the following concern:

Does the double key alone compromise the confidentiality of the normal and/or anamorphic message, in either a normal or anamorphic ciphertext, and, if so, to what extent can we avoid this?

Very few works [KPP+23b,CGM24a] (including the concurrent works [BPRS25,DL25b,JS24]) considered security against the anamorphic sender. Kutylowski et al. [KPP+23b] examined the security of anamorphic messages against the anamorphic sender, and Catalano, Giunta, and Migliaro [CGM24a] considered so-called "fully asymmetric security" (and only for *asymmetric* AME) where an anamorphic encryption of $(\mathsf{m}_0, \hat{\mathsf{m}}_0)$ is indistinguishable from an anamorphic encryption of $(\mathsf{m}_1, \hat{\mathsf{m}}_1)$ to the anamorphic sender.

AME under Chosen-Ciphertext Attacks. In (public-key) encryption, the gold standard for security is IND-CCA2 security, i.e., security even if the adversary has access to an decryption oracle that takes as input some (potentially adversarial) ciphertext and returns its decryption. When considering security against the dictator, at first glance, CCA security may seem equivalent to CPA security, since the secret key sk is known to the dictator who can thus perform decryption themselves. Of course in AME (under our syntax), there is anamorphic decryption that requires not only the secret key but also the double key, the latter being unknown to the dictator. Therefore, a natural question is:

To what extent can we provide security guarantees when an attacker has access to a regular and/or anamorphic decryption oracle?

Indeed if the encryption scheme is deployed via normal key generation, then there is no double key and hence no anamorphic decryption at all. Therefore, our new STE-xCCA1 security notion requires that the adversary, holding the secret key sk, cannot distinguish the anamorphic mode where there is an anamorphic decryption oracle that inputs ct and returns $\mathsf{aDec}(\mathsf{sk}, \mathsf{dk}, \mathsf{ct})$, from the normal mode where there is also a decryption oracle but always returns $\perp$.[18]

Let us see why CCA security, for example STE-xCCA1 security, is practically important in more detail. Suppose that the anamorphic sender Alice sends a ciphertext ct to the receiver Bob, where ct is an anamorphic encryption of

$$\text{normal message } \mathsf{m} = \text{“Glory to the dictator!”},$$

and

$$\text{anamorphic message } \hat{\mathsf{m}} = \text{“Down with the dictator! Protest at } \underline{\mathbf{10}} \text{ am.”}$$

We assume a dictator that surveils all users by decrypting every ciphertext that is sent. Suppose that message m looks innocent and the dictator decides to pass ciphertext ct to Bob. However, due to either benign faults or modification from a suspicious dictator, ciphertext ct has been changed to another ciphertext ct' where the normal message remains the same, but the anamorphic message changes to

$$\text{modified } \hat{\mathsf{m}}' = \text{“Down with the dictator! Protest at } \underline{\mathbf{8}} \text{ am.”}$$

Then, after receiving ct', Bob performs anamorphic decryption to obtain the modified anamorphic message $\hat{\mathsf{m}}'$ (this modification can be unknown to the dictator), and views it as call to action from Alice. Due to the inconsistent anamorphic messages contained in ct and ct', the use of such an anamorphic channel could be exposed to the dictator, and consequently the revolution might fail.

Relations with Robustness. To prevent a normal ciphertext from being misunderstood as containing an anamorphic message, [BGH+24] defined robustness for AME. Informally, an AME scheme has robustness if applying anamorphic decryption to a ciphertext generated from $\mathsf{Enc}(\mathsf{pk}, \mathsf{m})$ will output a failure indicator except with negligible probability. The definition of robustness has been considered in subsequent work [CGM24a,CGM24b,WHL24,JS24,DG25] and has been regarded as a standard notion for AME schemes. It seems that STE-xCCA1 security defined above is highly related to robustness, since to invalidate the anamorphic decryption oracle obtained by the adversary (the dictator), the anamorphic decryption should reject all ciphertexts produced by the adversary.

However, robustness as defined in [BGH+24] guarantees only the robustness of ciphertexts from normal encryption, but does not require anything on *adversarial* ciphertexts. In fact, to ensure STE-xCCA1 security, we need a stronger version of robustness where it is infeasible to generate a different ciphertext that could be

[18] In order to achieve STE-xCCA1 security, decryption cannot use *implicit rejection*.

anamorphically decrypted to a non-failure message, even given multiple anamorphic ciphertexts produced via $\mathsf{aEnc}(\mathsf{pk}, \mathsf{dk}, \mathsf{m}, \hat{\mathsf{m}})$ with $\mathsf{m}, \hat{\mathsf{m}}$ chosen by the adversary. We call this stronger notion ROB-CT+, which is a necessary condition for STE-xCCA1 security.

Despite the strength of STE-xCCA1, we emphasize that no existing AME schemes as stated achieve this property. That means the use of an anamorphic channel is in fact unsafe against a dictator making CCA attacks.

Moreover, we believe that threats against the anamorphic sender should also be considered in the CCA setting. Since in our treatment the anamorphic sender has access to the double key dk but not the secret key sk (which, as observed above, is not always true for anamorphic encryption [PPY22,BGH+24], but is more general), the latter being required in both normal decryption and anamorphic decryption, the anamorphic sender (as the adversary) is provided with both normal and anamorphic decryption oracles. Consequently, we extend the security notions of SIND-xCPA(1/2), SIND-nCPA, and SIND-aCPA(1/2/3) in the CPA setting to SIND-xCCA(1/2), SIND-nCCA and SIND-aCCA(1/2/3) in the CCA setting, respectively.

Asymmetric AME: Security Against Collusion. In the typical formulation of anamorphic encryption, the double key is assumed to be a symmetric key shared between the anamorphic sender and the receiver [PPY22,BGH+24]. Later works [KPP+23b,CGM24a,CGM24b,CGM25,BPRS25] considered the scenario that the double key held by the anamorphic sender, used for anamorphic encryption, in general *differs* from the double key held by the receiver used for anamorphic decryption. We call this an asymmetric AME scheme, and denote the sender double key by sdk and the receiver double key by rdk. Symmetric AME can be viewed as a special case of asymmetric AME when $\mathsf{dk} = \mathsf{sdk} = \mathsf{rdk}$.

While most security notions are similar between the symmetric case and the asymmetric case, we study the following strong security requirement:

What if a powerful adversary gains both the secret key sk *and the sender double key* sdk*? What security can be provided in this scenario?*[19]

The case that the adversary obtains both sk and sdk happens if the dictator and the anamorphic sender collude. In this scenario, the powerful adversary not only knows that the encryption scheme was deployed in the anamorphic mode (due to sdk), but also has access to the normal messages in both normal ciphertexts and anamorphic ciphertexts (due to sk). However, the adversary still does not know the anamorphic message concealed in an anamorphic ciphertext, or even, does not know whether a ciphertext is anamorphic or not, since the anamorphic decryption additionally requires the receiver double key rdk which is unknown to the adversary.

One may wonder what is the meaning of considering such a strong attack model. Consider a receiver leading an underground resistance movement against the dictator, and they deploy an (asymmetric) AME scheme so that the resistance can use sdk to send anamorphic messages privately to their leader. However, one member betrays the resistance and secretly hands over their sdk to the dictator. Before issuing an arrest warrant for the leader, the dictator aims to read anamorphic messages contained in the ciphertexts sent from other members of the movement. Furthermore, the dictator may want to identify exactly who belongs to this movement by determining whether they have sent an anamorphic ciphertext to the receiver. To prevent this, the receiver must ensure that the leakage of both sk and sdk does not compromise the confidentiality of anamorphic messages.

We consider CIND-xCPA1 and CIND-aCPA2 security (as well as CIND-xCCA1 and CIND-aCCA2 security in the CCA setting) to address this issue, and explore how to achieve those security notions in Section 6.

[19] Note that [CGM24a] considers CIND-aCPA2 security but not other notions in the collusion setting.