A preliminary version of this paper appears in Advances in Cryptology - ASIACRYPT 2023. This is the full version, with major revision.

Sender-Anamorphic Encryption Reformulated: Achieving Robust and Generic Constructions *

Yi Wang †
wangyi14@nudt.edu.cn

Rongmao Chen ‡
chromao@nudt.edu.cn

Xinyi Huang §
xinyi@ust.hk

Moti Yung ¶
moti@cs.columbia.edu

November 28, 2023

Abstract

Motivated by the violation of two fundamental assumptions in secure communication - receiver-privacy and sender-freedom - by a certain entity referred to as "the dictator", Persiano et al. introduced the concept of Anamorphic Encryption (AME) for public key cryptosystems (EUROCRYPT 2022). Specifically, they presented receiver/sender-AME, directly tailored to scenarios where receiver privacy and sender freedom assumptions are compromised, respectively. In receiver-AME, entities share a double key to communicate in anamorphic fashion, raising concerns about the online distribution of the double key without detection by the dictator. The sender-AME with no shared secret is a potential candidate for key distribution. However, the only such known schemes (i.e., LWE and Dual LWE encryptions) suffer from an intrinsic limitation and cannot achieve reliable distribution.

Here, we reformulate the sender-AME, present the notion of ℓ-sender-AME and formalize the properties of (strong) security and robustness. Robustness refers to guaranteed delivery of duplicate messages to the intended receiver, ensuring that decrypting normal ciphertexts in an anamorphic way or decrypting anamorphic ciphertexts with an incorrect duplicate secret key results in an explicit abort signal. We first present a simple construction for pseudo-random and robust public key encryption that shares the similar idea of public-key stegosystem by von Ahn and Hopper (EUROCRYPT 2004). Then, inspired by Chen et al.'s malicious algorithm-substitution attack (ASA) on key encapsulation mechanisms (KEM) (ASIACRYPT 2020), we give a generic construction for hybrid PKE with special KEM that encompasses well-known schemes, including ElGamal and Cramer-Shoup cryptosystems.

The constructions of ℓ-sender-AME motivate us to explore the relations between AME, ASA on PKE, and public-key stegosystem. The results show that a strongly secure ℓ-sender-AME is such a strong primitive that implies reformulated receiver-AME, public-key stegosystem, and generalized ASA on PKE. By expanding the scope of sender-anamorphic encryption and establishing its robustness, as well as exploring the connections among existing notions, we advance secure communication protocols under challenging conditions.

*In this full version, some subtlties in the proof of Theorem 4.1, Theorem 4.4 and 5.1 have been revised. Please see the corresponding theorems for mode details.

†National University of Defense Technology
‡National University of Defense Technology
§The Hong Kong University of Science and Technology (Guangzhou)
¶Columbia University & Google

Contents

1 Introduction

In the realm of cryptosystems, there is an implicit assumption that the receiver's secret key remains confidential (referred to as the receiver-privacy assumption), and the sender has the freedom to choose the message to be sent (referred to as the sender-freedom assumption). However, in reality, these fundamental assumptions can be completely violated by a controlling entity known as "the dictator" who possesses the ability to access any individual's secret key and censor the content of messages. Achieving both private and unrestricted communication in such a setting seems futile.

To address this critical issue, Persiano, Phan, and Yung introduced a new concept called "Anamorphic Encryption" [26]. This notion allows a well-established public-key cryptosystem to enable entities to encrypt differently hidden messages in what is called an anamorphic manner, thus evading the censorship imposed by the dictator. Specifically, they defined two variants of anamorphic encryption: receiver-anamorphic encryption and sender-anamorphic encryption, which provide secure communication while eliminating the reliance on the receiver-privacy and sender-freedom assumptions, respectively. Receiver-anamorphic encryption aims to facilitate secure communication in the face of a violating receiver-privacy assumption. By utilizing this technique, entities can engage in anamorphic communication, where the recipient's privacy is protected even in the presence of the dictator. On the other hand, sender-anamorphic encryption focuses on addressing the violation of the sender-freedom assumption. In this case, the sender can encrypt messages in an anamorphic manner, allowing them to transmit information without being constrained or controlled by the dictator.

Obviously, it is impossible to achieve the confidentiality of encrypted message against the dictator when the receiver only holds one secret key and has to reveal this key to the dictator. So, the receiver-anamorphic encryption requires that every pair of sender and receiver must share a double key that is unknown to the dictator. This double key is used to encode/retrieve a secret message into/from an anamorphic ciphertext in symmetric way, which raises a rather important problem: *How to distribute the double key for every entity pair without being detected by the dictator?* A trivial solution is offline key exchange which is extremely inefficient but most unlikely to be caught by the dictator who monitors the online communication constantly. In [26], the authors mentioned that the two-step bootstrap technique [19] of Horel et al. allows two entities to exchange a random string which is used to generate the double key. However, this technique involves the execution of pseudorandom key exchange protocol which is suspicious to the dictator who might ban the usage of such protocol.

The sender-anamorphic encryption formalized in the setting of no-shared secret [26] could be a potential candidate for realizing covert and efficient key distribution (using multiple receiver situations). In particular, when the dictator instructs Alice to send forced message $\mathtt{m}_0$ to Carol, Alice might intend to send duplicate message $\mathtt{m}_1$ (e.g., double key) to Bob. The sender-anamorphic encryption allows Alice to generate randomness via a special coin-toss faking algorithm $\mathsf{fRandom}$ that takes as input forced public key $\mathtt{fpk}$ (i.e., Carol's public key), duplicate public key $\mathtt{dpk}$ (i.e., Bob's public key), forced message $\mathtt{m}_0$ and duplicate message $\mathtt{m}_1$. Then, Alice encrypts forced message $\mathtt{m}_0$ with the selected randomness (produced by $\mathsf{fRandom}$) using forced public key $\mathtt{fpk}$, and obtains an anamorphic ciphertext $\mathtt{act}$ which gives duplicate message $\mathtt{m}_1$ when it is decrypted with duplicate secret key $\mathtt{dsk}$ (i.e., Bob's secret key). Finally, Alice sends ciphertext $\mathtt{act}$ to Carol only via public communication channel such that Bob can observe this ciphertext and retrieve the duplicate message $\mathtt{m}_1$. It is worth noting that the only difference between anamorphic and normal ciphertexts is the distribution of underlying randomness.

Non-Robustness of Sender-Anamorphic Encryption. Persiano et al. [26] pointed out that not every public-key encryption scheme (PKE) can be sender-anamorphic in the setting of

no-shared secret, and listed three sufficient conditions, including *common randomness property, message recovery from randomness* and *equal distribution of plaintext*, for a 1-bit PKE to be sender-anamorphic. So far, the only known sender-anamorphic encryptions are the LWE [28] and the Dual LWE [14] encryptions.

One can note that in these two encryptions, decrypting a normal ciphertext with incorrect secret key would return 0 or 1 with equal probability. This feature is undesirable and incurs the following two problems when applying them to distribute the double key.

- (**Misreading of normal ciphertexts**) Assume that Alice actually does want to send an ℓ-bit message $\mathtt{m}$ to Carol, and generates ℓ normal ciphertexts with uniformly sampled randomnesses for message $\mathtt{m}$. In this case, Bob cannot decide whether the observed ciphertexts are normal or anamorphic, and might take the decryption results of normal ciphertexts using his secret key as the duplicate message from Alice.
- (**Misreading of anamorphic ciphertexts**) Assume that Alice sends ℓ anamorphic ciphertexts, which include ℓ-bit duplicate message for Bob, to Carol, and there is a user Dave who also observes these anamorphic ciphertexts. In this case, Dave cannot tell whether these ciphertexts are intended for himself or not, and might take the decryption results of anamorphic ciphertexts using his secret key as the duplicate message from Alice.

To circumvent these problems, it is required that decrypting *normal ciphertext in anamorphic way* or *anamorphic ciphertext with wrong duplicate secret key* should produce an explicit abort signal. However, this demand leads to a contradiction! In particular, the anamorphic ciphertext can be viewed as a normal ciphertext with proper randomness, and the decryption algorithm always returns a bit for normal ciphertext. So, we cannot expect that the decryption algorithm would return an abort for anamorphic ciphertext.

Our observation regarding the non-robust nature of sender-anamorphic encryption was initially inspired by a recent notable work [4], which insightfully identified a similar issue within the context of receiver-anamorphic encryption. Specially, the security definition for receiver-anamorphic encryption in [26] did not consider the case where normal ciphertext is decrypted in an anamorphic way. Consequently, the work [4] defined the property of robustness for receiver-anamorphic encryption, and presented a range of novel constructions applicable to both general PKE schemes and special PKE schemes.

Motivating Question: Sender-Anamorphic Encryption with Robustness? The aforementioned problem seems to be unsolvable under the model of sender-anamorphic encryption with no shared key. That is, it looks impossible to construct "robust" sender-anamorphic encryption. We remark that compared with [4], our definition of robustness for sender-anamorphic encryption also consider an additional case that decrypting anamorphic ciphertexts with wrong duplicate secret key would produce explicit abort signal. Hence, we turn to reformulate the original definition of sender-anamorphic encryption, and try to explore feasible solutions in the tweaked model.

Recall that the sender of the original sender-anamorphic encryption is required to encode both forced and duplicate messages into one ciphertext. Given the fact that every entity in the cryptosystem usually sends more than one ciphertext to the others, we relax this requirement by allowing encoding duplicate message across multiple ciphertexts. In this way, the sender has to collect multiple pairs of forced public key and message to generate randomnesses for anamorphic ciphertexts. Intuitively, it seems that the sender might fail to generate proper randomnesses when the dictator asks the sender to encrypt only one forced message and to send the ciphertext each time. Fortunately, it would not be a problem if the generation of the

i-th randomness depends on the first i pairs of forced public key and message only, and it is possible to construct such coin-toss faking algorithm.

On the side of the receiver, the retrieval of duplicate message needs an alternative decryption algorithm that takes as input a set of ciphertexts and duplicate secret key. In particular, this decryption algorithm might provide explicit abort signal when these ciphertexts do not include any duplicate message (i.e., normal ciphertexts) or the duplicate secret key does not match these ciphertexts (i.e., anamorphic ciphertexts). With such an algorithm, it is possible to overcome the problem of misreading, achieving robustness for sender-anamorphic encryption and realize reliable duplicate message distribution. Therefore, a natural question that we mainly consider in this work is:

How to reformulate and construct sender-anamorphic encryption satisfying robustness in the setting of no shared key?

1.1 Our Contributions

In this work, we provide an affirmative answer to the above question. Specifically, we reformulate the syntax of sender-anamorphic encryption by permitting encoding duplicate message across multiple anamorphic ciphertexts (while obviously keeping the cryptosystem building block intact), and then formalize the properties of security and robustness. To show the feasibility of such a primitive, we present a simple construction for pseudo-random and robust PKE. The core idea is similar to the public-key stegosystem by von Ahn and Hopper [32]. Then, inspired by Chen et al.'s asymmetric algorithm-substitution attack (ASA) on key encapsulation mechanism (KEM) [10], we give a generic construction for hybrid PKE with special module-level KEM that encompasses well-known schemes including ElGamal and Cramer-Shoup cryptosystems.

These constructions have enlightened us to investigate whether the reformulated sender-anamorphic encryption might have some relations with existing notions. Indeed, it might be difficult to explore the direct relationship among them individually. So, we present the reformulated version of receiver-anamorphic encryption, and introduce the notion of generalized ASA against PKE as in [8] to facilitate the investigation.

We can now present the summary of main contributions of this work:

- We introduce the notion of ℓ-sender-anamorphic encryption (ℓ-sender-AME), and define the properties of (strong) security and robustness.
- We present two constructions of secure and robust ℓ-sender-AME. One is for pseudo-random and robust PKE and the other is for hybrid PKE with special module-level KEM which can be instantiated with well-known schemes including ElGamal and Cramer-Shoup cryptosystems.
- We explore the relations between ℓ-sender-AME and other related primitives, as shown in Fig. 1, including reformulated receiver-anamorphic encryption (ℓ-receiver-AME), public-key stegosystem, and generalized ASA on PKE.

1.2 Results Overview

A Reformulated Model for Robustness: ℓ-Sender-AME. In sender-AME model, the syntax of PKE is only augmented with a coin-toss faking algorithm that generates randomness according to one pair of forced public key and message, and the duplicate public key and message. In our reformulated model, the coin-toss faking algorithm would take more than one

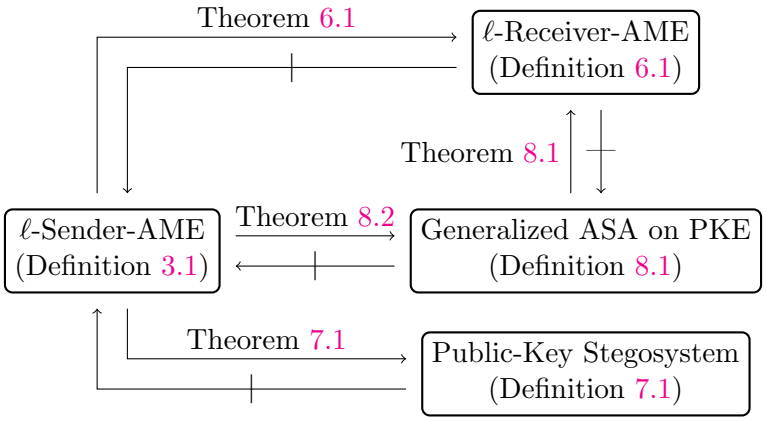

Figure 1: Relations between ℓ-sender-AME and other related primitives.

pair of forced public key and message, and there exists an alternative decryption algorithm that retrieves the duplicate message from a set of anamorphic ciphertexts using duplicate secret key. We name the reformulated notion "ℓ-sender-AME," where ℓ denotes the number of anamorphic ciphertexts required for duplicate message embedding. Clearly, this definition includes the original sender-AME when $\ell = 1$ and the alternative decryption algorithm is the decryption algorithm of PKE.

There are two properties defined for ℓ-sender-AME: *security* and *robustness*. The definition of security is extended from that for original sender-AME trivially. Roughly, it indicates that the dictator who knows all the forced public keys cannot distinguish normal and anamorphic ciphertexts outputted by the encryption oracle with overwhelming advantage. Moreover, we consider a strong variant of security by providing the dictator all the forced secret keys and duplicate public key, which also implies the violation of the receiver-privacy assumption to some extent. This strong security permits us to evade the surveillance by the dictator who violates both receiver-privacy and sender-freedom assumptions. As mentioned above, the meaning of robustness is twofold: the alternative decryption algorithm would return abort when 1) the inputted ciphertexts are normal or 2) the inputted anamorphic ciphertexts and duplicate secret key do not match.

Generic Constructions of ℓ-Sender-AME. In this work, we present two generic constructions of ℓ-sender-AME satisfying both security and robustness. One is for pseudo-random and robust PKE, and the other is for hybrid PKE with special module-level KEM that encompasses well-known schemes including ElGamal and Cramer-Shoup cryptosystems.

Construction I: *Pseudo-random and Robust PKE.* Pseudo-random PKE produces ciphertext that is indistinguishable from random bits [32]. This feature allows us to embed the ciphertext of duplicate message across multiple ciphertexts without being detected by the dictator. The robustness of PKE [1] ensures that a ciphertext cannot be valid under two different decryption keys, which contributes to the robustness of ℓ-sender-AME.

In more details, we encode the i-th bit of the ciphertext of duplicate message into the i-th anamorphic ciphertext using the rejection sampling technique. That is, repeating randomness sampling until the desired bit can be publicly derived from the i-th anamorphic ciphertext. Even though anyone can retrieve the ciphertext of a duplicate message from multiple anamorphic ciphertexts, the dictator who does not hold the duplicate secret key cannot decide if the obtained string is a ciphertext or a random bit string by the pseudo-randomness of PKE.

So far, there are only a few pseudo-random PKEs [32, 23]. Although most well-known public-key cryptosystems do not satisfy pseudo-randomness, it is possible to transform the ciphertext of

duplicate message into a pseudo-random string. Here are some optional approaches: removing the distribution bias caused by the algebra [34], applying the covert key exchange technique [36] or encoding elliptic-curve point to be indistinguishable from uniform random string [9, 31].

Construction II: *Hybrid PKE with special KEM.* Hybrid PKE relies on KEM which encapsulates the key for data encapsulation mechanism (DEM) which is used to encrypt the plaintext. Inspired by the generic ASA on KEM of Chen et al. [10], we encode the $(i-1)$-th bit b_{i-1} of duplicate message across the key encapsulation part of the $(i-1)$-th and i-th anamorphic ciphertexts C_{i-1} and C_i. In particular, let r_i denote the underlying randomness of key ciphertext C_i, we generate randomness r_i from r_{i-1} and duplicate public key. If $b_{i-1} = 1$, we add a perturbation to r_i. Otherwise, we do nothing to r_i. In the process of alternative decryption, we first recover randomness r'_i from C_{i-1} and duplicate secret key by the universal decryptability of KEM, and compute the perturbed randomness r''_i of r'_i. Then, we compare C_i with C'_i and C''_i derived from r'_i and r''_i respectively. If $C_i = C'_i$, then $b_{i-1} = 0$. If $C_i = C''_i$, then $b_{i-1} = 1$. Otherwise, the alternative decryption algorithm returns abort. By the key-pseudo-randomness of KEM, it is hard to detect the correlation between C_{i-1} and C_i for the dictator who does not know the duplicate secret key.

We show the robustness of this construction as following: For normal ciphertext, its randomness is uniformly distributed over appropriate space. The probability of the event that r_i is correlated to r_{i-1} and the duplicate public/secret key is negligible. Thus, decrypting normal ciphertexts in anamorphic way would produce explicit abort. For anamorphic ciphertexts, decrypting them with incorrect duplicate secret key does not reveal the correlation between key ciphertexts, and the unintended receiver would receive abort symbol.

We remark that it is possible to encode t-bits string ($t > 1$) into two successive random fields at the cost of exponential time complexity (in the string length) for alternative decryption algorithm. In particular, the coin-toss faking algorithm converts t-bits string into element in randomness space and then adds it to randomness r_i, while the alternative decryption algorithm has to derive at most 2^t ciphertexts to compare with inputted ciphertext C_i. Thus, such variation is only feasible for short bit string. Another concern with this encoding is that the perturbation over randomnesses disables us from providing a valid explanation on the selection of randomness (e.g., revealing the pre-image of randomness under one-way hash function specified by the dictator in case it is demanded).

Relations Between Existing Notions. The constructions above suggest that there exist some inner connections among anamorphic encryption, public-key stegosystem, and algorithm-substitution attack.

Relation between ℓ-Sender-AME and ℓ-Receiver-AME. In receiver-AME, parties share a double key that is used to encode both normal and anamorphic plaintexts into an anamorphic ciphertext or to retrieve anamorphic plaintext from anamorphic ciphertext. Analogous to ℓ-sender-AME, we present the notion of ℓ-receiver-AME where the anamorphic plaintext is encoded across ℓ anamorphic ciphertexts using double key. Once the double key is set as the forced public keys and duplicate public/secret key pair in ℓ-sender-AME, the anamorphic ciphertexts of ℓ-receiver-AME can be generated using the biased randomnesses outputted by the coin-toss faking algorithm in ℓ-sender-AME.

Note that the security model for sender/receiver-AME only captures the violation of sender-freedom or receiver-privacy assumption. It seems hard to reduce one security notion to another. We overcome this problem by strengthening the ability of the adversary in the security game for ℓ-sender-AME. In the game of strong security, the adversary knows all the forced secret keys and the duplicate public key. Now, we get the following result.

Theorem 1.1. (Informal). *For any strongly secure ℓ-sender-AME* PKE*, it is also a secure*

ℓ-receiver-AME.

It is obvious that not any ℓ-receiver-AME is an ℓ-sender-AME, as the double key in ℓ-receiver-AME may not be parsed as a public/secret key pair.

Relation between ℓ-Sender-AME and Public-Key Stegosystem. Parties in public-key stegosystem with no shared secret key are able to communicate in steganographic way over a public channel such that no eavesdropper of the channel can detect the existence of hidden messages. The notion of ℓ-sender-AME is similar to the public-key stegosystem in the sense of, both, setting and target. Note that the pseudo-random PKE was first proposed to construct public-key stegosystem, and our first ℓ-sender-AME construction is designed for pseudo-random PKE. It is naturally leading us to the following result.

Theorem 1.2. (Informal). *For any strongly secure ℓ-sender-AME* PKE*, there exists a public-key stegosystem* PKS *built over* PKE.

Conversely, it seems that every public-key stegosystem should imply an ℓ-sender-AME. However, constructing public-key stegosystem is more diverse than that of ℓ-sender-AME. For instance, the chosen-stegotext secure stegosystem in [32] uses a sender's secret key to generate stegotexts, while the input of coin-toss faking algorithm in ℓ-sender-AME does not involve any secret keys.

Relations between AME and Generalized ASA on PKE. The attacker in ASA on PKE substitutes the encryption algorithm in PKE with subverted version so as to recover the underlying plaintext from subverted ciphertext using subversion key, which is rather different from the goal of anamorphic encryption. Instead, we present the notion of generalized ASA on PKE as in [8] where the subverted encryption algorithm also encodes a subliminal message into the ciphertext and the attacker's goal is extracting this subliminal message from subverted ciphertext. The original ASA on PKE is a special case of generalized ASA on PKE where the subliminal message equals to the plaintext for encryption.

Depending on how the subversion key is generated and used, the generalized ASA on PKE can be classified into two types: symmetric and asymmetric. In symmetric ASA, there is only one subversion key used by both subverted encryption and extraction algorithms. Recall that, in ℓ-receiver-AME, the same double key is used to encode and extract anamorphic plaintext. In this case, the subversion key corresponds to the double key. Furthermore, we shows that a symmetric ASA on PKE implies the underlying PKE is an ℓ-receiver-AME.

Theorem 1.3. (Informal). *Let* PKE *be a CPA secure PKE. If there exists a symmetric and generalized ASA on* PKE*, then* PKE *is also a secure ℓ-receiver-AME.*

In ASA on PKE, the subversion key is independent of the public and secret key for subverted encryption algorithm. Once the generation of double key in ℓ-receiver-AME relies on the secret key, it might be impossible to construct subversion key generation algorithm with anamorphic key generation algorithm. Thus, not every ℓ-receiver-AME implies a symmetric ASA on PKE.

In asymmetric ASA, the subversion key is a pair of public/secret key, where the public subversion key is hardwired into the subverted encryption algorithm and the extraction of subliminal message requires secret subversion key. Note that if the duplicate public/secret key pair in ℓ-sender-AME is also a subversion key pair for ASA, we can built an ASA on PKE from ℓ-sender-AME.

Theorem 1.4. (Informal). *Let* PKE *be a strongly secure ℓ-sender-AME. Then there exists an asymmetric and generalized ASA on* PKE.

Conversely, not every asymmetric ASA on PKE directly implies that the underlying PKE is an ℓ-sender-AME. In particular, when the subversion key generation of ASA is different from the key generation of PKE, it is unlikely to build the coin-toss faking algorithm following the core idea of ASA, as this algorithm only takes normal public keys and messages as input.

1.3 Related Work

Anamorphic Cryptography. Since the notion of anamorphic encryption was first proposed in 2022, several works [4, 22, 21] have been presented to expand the scope of anamorphic cryptography. In particular, Banfi et al. [4] and Kutylowski et al. [22] both refined the definition of original receiver-AME but for different purposes. In [4], the generation of anamorphic key pair and double key is decoupled, and the authors aims to build receiver-AME satisfying robustness. In [22], both anamorphic public key and double key, instead of double key only, are used to encode normal and anamorphic plaintexts into an anamorphic ciphertext. This refinement permits some wide range of cryptosystems, including RSA-OAEP [7], Goldwasser-Micali [15], Paillier [25], ElGamal [13] and Cramer-Shoup [11], to be receiver-anamorphic. Kutylowski et al. [21] introduced the notion of anamorphic signature scheme where a signature is embedded with an anamorphic message, which is only readable to the one owning the double key.

Steganography. The work of Simmons [30] initiated the study of steganography forty years ago. In 2002, the first complexity-theoretic model of provably secure steganography was present by Hopper et al. [18]. Further, von Ahn and Hopper [32] explored the public-key variant of steganography and proposed the notion of public-key stegosystem and steganographic key exchange protocol. Backes and Cachin [3] enhanced the security model of public-key stegosystem to defend against active attacks. Note that our results in this work might provide a new way to build public-key stegosystem from ℓ-sender-AME.

Algorithm-Substitution Attack. At CRYPTO 2014, Bellare et al. introduced the notion of algorithm-substitution attack [6], which considers the ability of attackers to subvert the implementation of cryptographic algorithms in reality. This concept dates back to the notion of kleptography [34] by Young and Yung. In a series of works [34, 35, 36] on kleptography, the subversion of the key generation algorithm enables attacker to recover the secret information exclusively. The Snowden revelations in 2013 reignited the enthusiasm of the academic community about this topic [6, 12, 5, 2, 8, 29, 10]. Although the ASA on encryption scheme helps the dictator to conduct mass-surveillance without being detected, our work shows that the ordinary users can also take advantage of the rationale behind ASA to communicate in a covert way against the dictator. Namely, the users employ malicious cryptography against the malice of the dictator!

2 Preliminaries

NOTATIONS. For any $n \in \mathbb{N}^+$, we denote the negligible function over n as $\mathsf{negl}(n)$. For any $i \in \mathbb{N}^+$, $[i]$ denotes integer set $\{1, 2, \cdots, i\}$. For any $i, j \in \mathbb{N}$ with $i < j$, $[i, j]$ denotes integer set $\{i, i+1, \cdots, j\}$. For any non-empty set $\mathcal{X}$, $x \leftarrow_{\$} \mathcal{X}$ denotes sampling x from $\mathcal{X}$ uniformly at random. For any randomized algorithm $\mathsf{Alg}(x)$, $y \leftarrow_{\$} \mathsf{Alg}(x)$ denotes the random output of $\mathsf{Alg}(x)$. For any deterministic algorithm $\mathsf{Alg}(x)$, $y := \mathsf{Alg}(x)$ denotes the deterministic output of $\mathsf{Alg}(x)$. For n elements $\mathtt{a}_1, \mathtt{a}_2, \cdots, \mathtt{a}_n$, we denote the set $\{\mathtt{a}_i\}_{i\in[n]}$ as $\mathtt{A}$.

2.1 Public-Key Encryption (PKE)

A public-key encryption scheme PKE consists of following algorithms:

- $\mathsf{Setup}(1^n)$ takes as input 1^n, and returns the public parameter $\mathtt{pp}$ which is an implicit input of encryption and decryption algorithms.
- $\mathsf{Gen}(\mathtt{pp})$ takes as input $\mathtt{pp}$ and returns a public/secret key pair $(\mathtt{pk}, \mathtt{sk})$.
- $\mathsf{Enc}(\mathtt{pk}, \mathtt{m})$ takes as input $\mathtt{pk}$ and a plaintext $\mathtt{m}$, and returns a ciphertext $\mathtt{ct}$.
- $\mathsf{Dec}(\mathtt{sk}, \mathtt{ct})$ take as input $\mathtt{sk}$ and $\mathtt{ct}$, and returns $\mathtt{m}'$ or an abort symbol $\perp$.

Correctness. PKE is correct if, let $\mathcal{M}$ be the plaintext space, for any $\mathtt{m} \in \mathcal{M}$,

$$\Pr\left[\mathsf{Dec}(\mathtt{sk}, \mathtt{ct}) \neq \mathtt{m} : \begin{array}{l} \mathtt{pp} \leftarrow_{\$} \mathsf{Setup}(1^n) \\ (\mathtt{pk}, \mathtt{sk}) \leftarrow_{\$} \mathsf{Gen}(\mathtt{pp}) \\ \mathtt{ct} \leftarrow_{\$} \mathsf{Enc}(\mathtt{pk}, \mathtt{m}) \end{array}\right] \leq \mathsf{negl}(n)\,.$$

Security. PKE is CPA secure if for any PPT adversary $\mathcal{A}_1$ and $\mathcal{A}_2$,

$$\left|\Pr\left[b = b' : \begin{array}{l} \mathtt{pp} \leftarrow_{\$} \mathsf{Setup}(1^n) \\ (\mathtt{pk}, \mathtt{sk}) \leftarrow_{\$} \mathsf{Gen}(\mathtt{pp}) \\ (\mathtt{m}_0, \mathtt{m}_1, \mathtt{st}) \leftarrow \mathcal{A}_1(\mathtt{pp}, \mathtt{pk}) \\ b \leftarrow_{\$} \{0,1\} \\ \mathtt{ct}^* \leftarrow_{\$} \mathsf{Enc}(\mathtt{pk}, \mathtt{m}_b) \\ b' \leftarrow \mathcal{A}_2(\mathtt{st}, \mathtt{ct}^*) \end{array}\right] - \frac{1}{2}\right| \leq \mathsf{negl}(n)\,.$$

The notion of anamorphic encryption is defined over public-key encryption. We recall the definitions of anamorphic encryption in Appendix A.1.

2.2 Entropy Smoothing Hash Functions

Let $\mathcal{H} = \{H_{\hat{k}}\}_{\hat{k} \in \hat{\mathcal{K}}}$ be a keyed hash function family associated with key space $\hat{\mathcal{K}}$, groups X, Y and hash function $H_{\hat{k}} : X \to Y$. We say $\mathcal{H}$ is entropy smoothing if for any PPT adversary $\mathcal{A}$ and $\hat{k} \leftarrow_{\$} \hat{\mathcal{K}}$,

$$\left|\Pr\left[\mathcal{A}(\hat{k}, H_{\hat{k}}(x)) = 1 \middle| x \leftarrow_{\$} X\right] - \Pr\left[\mathcal{A}(\hat{k}, y) = 1 \middle| y \leftarrow_{\$} Y\right]\right| \leq \mathsf{negl}(n)\,.$$

3 Reformulating Sender-Anamorphic Encryption

In this section, we first present the reformulated version of sender-AME in the setting of no shared secret key, and then define the properties of (strong) security and robustness for this new primitive.

3.1 ℓ-Sender-Anamorphic Encryption (ℓ-Sender-AME)

Definition 3.1 (ℓ-Sender-Anamorphic Encryption)**.** Let PKE be a public key encryption scheme. For $\ell \in \mathbb{N}^+$, we say PKE is ℓ-sender-anamorphic if 1) there exists a sender-anamorphic extension $(\mathsf{fRandom}, \mathsf{dDec})$:

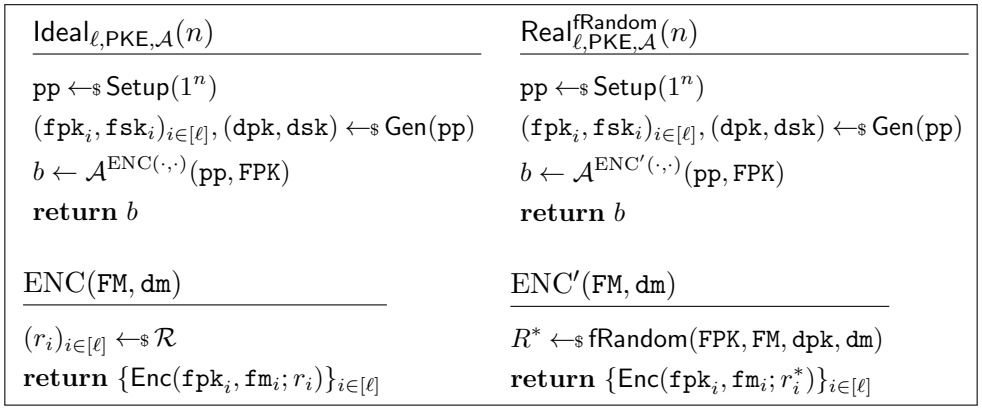

Figure 2: Definition of game $\mathsf{Ideal}_{\ell,\mathsf{PKE},\mathcal{A}}(n)$ and $\mathsf{Real}^{\mathsf{fRandom}}_{\ell,\mathsf{PKE},\mathcal{A}}(n)$.

- $\mathsf{fRandom}(\mathtt{FPK}, \mathtt{FM}, \mathtt{dpk}, \mathtt{dm})$ takes a forced public key set $\mathtt{FPK} = \{\mathtt{fpk}_i\}_{i\in[\ell]}$, a forced plaintext set $\mathtt{FM} = \{\mathtt{fm}_i\}_{i\in[\ell]}$, a duplicate public key $\mathtt{dpk}$ and a duplicate plaintext $\mathtt{dm}$, and returns a randomness set $R = \{r_i\}_{i\in[\ell]}$;
- $\mathsf{dDec}(\mathtt{dsk}, \mathtt{CT})$ takes as input a duplicate secret key $\mathtt{dsk}$ and a ciphertext set $\mathtt{CT}$, and returns the duplicate plaintext $\mathtt{dm}$,

and 2) let $\mathcal{M}$ and $\overline{\mathcal{M}}$ be the forced and duplicate plaintext space respectively, for any $\mathtt{FM} \in \mathcal{M}^\ell$, any $\mathtt{dm} \in \overline{\mathcal{M}}$,

$$\Pr\left[\mathsf{dDec}(\mathtt{dsk}, \mathtt{CT}) \neq \mathtt{dm} : \begin{array}{l} \mathtt{pp} \leftarrow\!\!\$\ \mathsf{Setup}(1^n) \\ (\mathtt{fpk}_i, \mathtt{fsk}_i)_{i\in[\ell]}, (\mathtt{dpk}, \mathtt{dsk}) \leftarrow\!\!\$\ \mathsf{Gen}(\mathtt{pp}) \\ R \leftarrow\!\!\$\ \mathsf{fRandom}(\mathtt{FPK}, \mathtt{FM}, \mathtt{dpk}, \mathtt{dm}) \\ \mathtt{CT} := \{\mathsf{Enc}(\mathtt{fpk}_i, \mathtt{fm}_i; r_i)\}_{i\in[\ell]} \end{array}\right] \leq \mathsf{negl}(n)$$

When $\ell = 1$ and algorithm dDec is the decryption algorithm Dec of PKE, the definition above is the original sender-AME in [26].

3.2 Security

The property of security means that it is hard for anyone who does not possess the duplicate secret key to distinguish a set of ciphertexts generated with uniformly sampled randomnesses from the output of coin-toss faking algorithm with overwhelming advantage.

Definition 3.2 (Secure ℓ-Sender-AME)**.** Let $\mathsf{PKE} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an ℓ-sender-AME associated with extension $(\mathsf{fRandom}, \mathsf{dDec})$. We say PKE is a secure ℓ-sender-AME if 1) PKE is CPA secure, and 2) for any PPT adversary $\mathcal{A}$ in Fig. 2,

$$\left|\Pr[\mathsf{Ideal}_{\ell,\mathsf{PKE},\mathcal{A}}(n) = 1] - \Pr[\mathsf{Real}^{\mathsf{fRandom}}_{\ell,\mathsf{PKE},\mathcal{A}}(n) = 1]\right| \leq \mathsf{negl}(n)\,.$$

If condition 2) still holds when $\mathcal{A}$ also owns $\mathtt{FSK} := \{\mathtt{fsk}_i\}_{i\in[\ell]}$ and $\mathtt{dpk}$, we say PKE is a *strongly* secure ℓ-sender-AME.

In fact, the definition of strongly secure ℓ-sender-AME also captures the violation of receiver-privacy assumption. specifically, the adversary $\mathcal{A}$ owns the forced secret keys of receivers, and can decrypt ciphertexts to obtain the forced plaintexts. As will be shown later, this strong security for sender-AME implies the security for receiver-AME. Hence, we are interested in constructing strongly secure ℓ-sender-AME.

3.3 Robustness

Roughly, the property of robustness indicates that both decrypting normal ciphertexts in the anamorphic way and decrypting anamorphic ciphertexts with incorrect duplicate secret key would return abort signaling explicitly.

Definition 3.3 (Robust ℓ-Sender-AME)**.** Let $\mathsf{PKE} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an ℓ-sender-AME associated with extension $(\mathsf{fRandom}, \mathsf{dDec})$. We say PKE is a robust ℓ-sender-AME if for any PPT adversary $\mathcal{A}$,

$$\Pr\left[\mathsf{dDec}(\mathtt{dsk}, \mathtt{CT}) \neq \perp : \begin{array}{l} \mathtt{pp} \leftarrow\!\!\$\, \mathsf{Setup}(1^n) \\ (\mathtt{fpk}_i, \mathtt{fsk}_i)_{i\in[\ell]}, (\mathtt{dpk}, \mathtt{dsk}) \leftarrow\!\!\$\, \mathsf{Gen}(\mathtt{pp}) \\ \mathtt{FM} \leftarrow \mathcal{A}(\mathtt{pp}, \mathtt{FPK});\ (r_i)_{i\in[\ell]} \leftarrow\!\!\$\, \mathcal{R} \\ \mathtt{CT} := \{\mathsf{Enc}(\mathtt{fpk}_i, \mathtt{fm}_i; r_i)\}_{i\in[\ell]} \end{array}\right] \leq \mathsf{negl}(n)\,, \tag{1}$$

where $\mathtt{FM} \in \mathcal{M}^\ell$, and

$$\Pr\left[\mathsf{dDec}(\mathtt{dsk}', \mathtt{CT}) \neq \perp : \begin{array}{l} \mathtt{pp} \leftarrow\!\!\$\, \mathsf{Setup}(1^n) \\ (\mathtt{fpk}_i, \mathtt{fsk}_i)_{i\in[\ell]}, (\mathtt{dpk}, \mathtt{dsk}) \leftarrow\!\!\$\, \mathsf{Gen}(\mathtt{pp}) \\ (\mathtt{dpk}', \mathtt{dsk}') \leftarrow\!\!\$\, \mathsf{Gen}(\mathtt{pp}) \\ (\mathtt{FM}, \mathtt{dm}) \leftarrow \mathcal{A}(\mathtt{pp}, \mathtt{FPK}, \mathtt{dpk}) \\ R \leftarrow\!\!\$\, \mathsf{fRandom}(\mathtt{FPK}, \mathtt{FM}, \mathtt{dpk}, \mathtt{dm}) \\ \mathtt{CT} := \{\mathsf{Enc}(\mathtt{fpk}_i, \mathtt{fm}_i; r_i)\}_{i\in[\ell]} \end{array}\right] \leq \mathsf{negl}(n)\,, \tag{2}$$

where $\mathtt{FM} \in \mathcal{M}^\ell$ and $\mathtt{dm} \in \overline{\mathcal{M}}$.

4 Construction I: Pseudo-random and Robust PKE

In this section, we show that any pseudo-random and robust PKE is ℓ_{ct}-sender-anamorphic, where ℓ_{ct} is the bit length of ciphertext. The core idea is embedding the ciphertext of duplicate message into multiple normal ciphertexts bit-by-bit using the rejection sampling technique, which is inspired by the construction of public-key stegosystem in [32]. By pseudo-randomness, the dictator cannot distinguish whether observed ciphertexts carry the ciphertext of duplicate message or not. By robustness, decrypting the ciphertext of duplicate message with incorrect duplicate secret key would return an abort symbol.

4.1 Pseudo-random and Robust PKE

We recall the definition of pseudo-randomness for PKE in [32] as below.

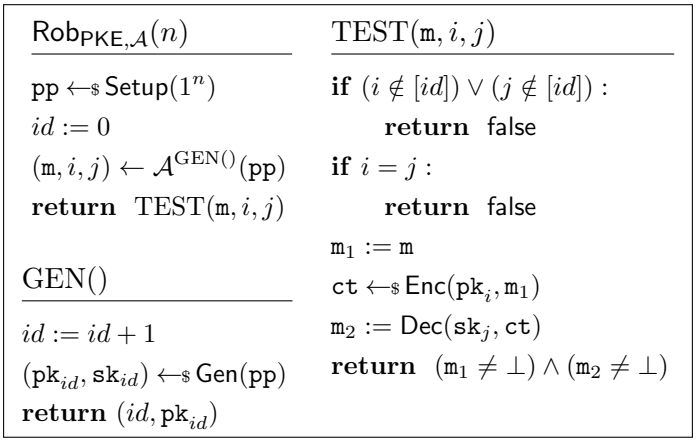

Figure 3: Definition of game $\mathsf{Rob}_{\mathsf{PKE},\mathcal{A}}(n)$.

Definition 4.1 (Pseudo-randomness [32])**.** Let $\mathsf{PKE} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme. We say PKE is *indistinguishable from random bits under chosen plaintext attack (pseudo-random)* if for any PPT adversary $\mathcal{A}_1, \mathcal{A}_2$,

$$\left| \Pr\left[b = b' : \begin{array}{l} \mathtt{pp} \leftarrow\!\!\$\ \mathsf{Setup}(1^n) \\ (\mathtt{pk}, \mathtt{sk}) \leftarrow\!\!\$\ \mathsf{Gen}(\mathtt{pp}) \\ (\mathtt{m}^*, \mathtt{st}) \leftarrow \mathcal{A}_1(\mathtt{pp}, \mathtt{pk}) \\ \mathtt{ct}_0 \leftarrow\!\!\$\ \mathsf{Enc}(\mathtt{pk}, \mathtt{m}^*) \\ \mathtt{ct}_1 \leftarrow\!\!\$\ \{0,1\}^{|\mathtt{ct}_0|} \\ b \leftarrow\!\!\$\ \{0,1\} \\ b' \leftarrow \mathcal{A}_2(\mathtt{st}, \mathtt{ct}_b) \end{array} \right] - \frac{1}{2} \right| \leq \mathsf{negl}(n)\,.$$

The robustness presented by Abdalla et al. [1] for PKE captures the difficulty of generating a ciphertext which is valid under two different decryption keys. Specifically, they formalized four definitions for robustness including weak and strong robustness in the setting of CPA and CCA security respectively (i.e., WROB-CPA, WROB-CCA, SROB-CPA, SROB-CCA). Here we require the PKE to satisfy the WROB-CPA property and recall its definition below.

Definition 4.2 (Robustness [1])**.** Let $\mathsf{PKE} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme. We say PKE is robust if for any PPT adversary $\mathcal{A}$ in Fig. 3,

$$\Pr\left[\mathsf{Rob}_{\mathsf{PKE},\mathcal{A}}(n) = \mathsf{true}\right] \leq \mathsf{negl}(n)\,.$$

So far, there are only a few pseudo-random PKE [32, 23], where Möller's scheme [23] also satisfies robustness. It seems that the coverage of this construction is rather limited. The following description of sender-anamorphic extension indicates that we only need the ciphertext (of duplicate message) can be encoded into a string which is indistinguishable from a uniform random, and here are several approaches [34, 36, 9, 31] to achieve such an encoding.

4.2 Sender-Anamorphic Extension

Let PKE be a pseudo-random and robust PKE with plaintext space $\mathcal{M}$, randomness space $\mathcal{R}$ and ciphertext space $\mathcal{C}$. The length of ciphertext in PKE is denoted by ℓ_{ct}. Fig. 4 depicts the

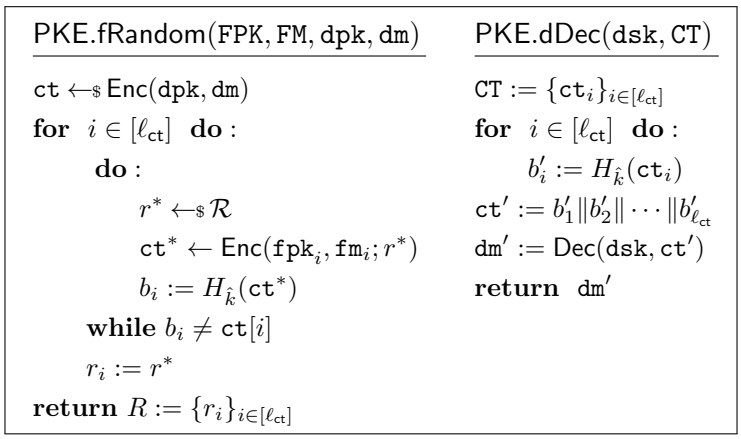

Figure 4: Sender-anamorphic extension for pseudo-random and robust PKE.

details of sender-anamorphic extension for PKE, and PKE is an ℓ_{ct}-sender-AME with duplicate plaintext space $\mathcal{M}$. We point out that the keyed hash function $H_{\hat{k}} : \mathcal{C} \to \{0,1\}$ used in algorithm fRandom and dDec is entropy smoothing and accessible to the public including the dictator.

4.3 Security Analysis

Theorem 4.1 (Robustness[1])**.** *Let* PKE *be a pseudo-random and robust PKE.* PKE *is a robust* ℓ_{ct}*-sender-AME with extension algorithms (in Fig. 4).*

Proof. Let $\mathbf{H}_0$ denote the game for $\mathcal{A}$ in Equation 1. Game $\mathbf{H}_1$ is the same as $\mathbf{H}_0$ except that all the ciphertexts in $\mathtt{CT}$ are sampled from $\{0,1\}^{\ell_{\mathsf{ct}}}$ uniformly, instead of generated by encrypting $\mathtt{FM}$.

Lemma 4.2. *By the pseudo-randomness of* PKE, $\mathbf{H}_0 \approx_c \mathbf{H}_1$.

Proof. Let $\mathbf{H}_{0,0} = \mathbf{H}_0$ and $\mathbf{H}_{0,i}$ be the same as $\mathbf{H}_{0,i-1}$ except $\mathtt{ct}_i$ in $\mathtt{CT}$ is replaced to a bit-string uniformly sampled from $\{0,1\}^{\ell_{\mathsf{ct}}}$ for $i \in [\ell_{\mathsf{ct}}]$. We have $\mathbf{H}_1 = \mathbf{H}_{0,\ell_{\mathsf{ct}}}$. To prove $\mathbf{H}_{0,i-1} \approx_c \mathbf{H}_{0,i}$, we show how to build $\mathcal{B}_1$ and $\mathcal{B}_2$ to break the pseudo-randomness of PKE as follows.

- $\mathcal{B}_1$ receives $(\mathtt{pp}, \mathtt{pk})$, sets $\mathtt{fpk}_i = \mathtt{pk}$, generates $(\mathtt{fpk}_k, \mathtt{fsk}_k)_{k \in [\ell_{\mathsf{ct}}] \setminus \{i\}}$ and $(\mathtt{dpk}, \mathtt{dsk})$ by running $\mathsf{Gen}(\mathtt{pp})$, and sends $(\mathtt{pp}, \mathtt{FPK})$ to $\mathcal{A}$.
- After receiving $\mathtt{FM}$ from $\mathcal{A}$, $\mathcal{B}_1$ forwards $\mathtt{fm}_i$ and state $\mathtt{st}$ to the challenger, and $\mathcal{B}_2$ receives $(\mathtt{ct}^*, \mathtt{st})$. $\mathcal{B}_2$ sets the first $i-1$ ciphertexts in $\mathtt{CT}$ as bit-strings uniformly sampled from $\{0,1\}^{\ell_{\mathsf{ct}}}$, the i-th ciphertext as $\mathtt{ct}^*$, and computes the last $\ell_{\mathsf{ct}} - i$ ciphertexts using Enc.

If $\mathtt{ct}^*$ is a ciphertext of $\mathtt{fm}_i$ under $\mathtt{pk}$, $\mathcal{B}_1$ and $\mathcal{B}_2$ simulates $\mathbf{H}_{0,i-1}$ for $\mathcal{A}$. Otherwise, $\mathtt{ct}^* \leftarrow_{\$} \{0,1\}^{\ell_{\mathsf{ct}}}$ and the simulation is $\mathbf{H}_{0,i}$. □

Game $\mathbf{H}_2$ is the same as $\mathbf{H}_1$ except that challenger samples a bit-string $\mathtt{ct}'$ from $\{0,1\}^{\ell_{\mathsf{ct}}}$ uniformly and computes $\mathtt{dm} := \mathsf{Dec}(\mathtt{dsk}, \mathtt{ct}')$, instead of running dDec.

Lemma 4.3. *By the entropy smoothness of* $\{H_{\hat{k}}\}_{\hat{k} \in \hat{\mathcal{K}}}$, $\mathbf{H}_1 \approx_c \mathbf{H}_2$.

[1]Compared with the conference version, the proof has been enhanced with more details.

Proof. Let $\mathbf{H}_{1,0} = \mathbf{H}_1$ and $\mathbf{H}_{1,i}$ be the same as $\mathbf{H}_{1,i-1}$ except that the i-th bit of $\mathtt{ct}'$ in dDec is uniformly sampled from $\{0,1\}$, instead of $H_{\hat{k}}(\mathtt{ct}_i)$. We have $\mathbf{H}_2 = \mathbf{H}_{1,\ell_{\mathtt{ct}}}$. To prove $\mathbf{H}_{1,i-1} \approx_c \mathbf{H}_{1,i}$, we show how to build $\mathcal{B}$ to break the entropy smoothness of $\{H_{\hat{k}}\}_{\hat{k}\in\hat{\mathcal{K}}}$ as follows.

Let $(\hat{k}, y)$ be the instance that $\mathcal{B}$ receives from its challenger. $\mathcal{B}$ generates public parameter, forced and duplicate public/secret key pairs as in $\mathbf{H}_0$. In the execution of dDec, the first $i-1$ bits of $\mathtt{ct}'$ are uniformly sampled from $\{0,1\}^{i-1}$, the i-th bit of $\mathtt{ct}'$ is set as y and the last $\ell_{\mathtt{ct}} - i$ bits are derived from $\{\mathtt{ct}_j\}_{j\in[i+1,\ell_{\mathtt{ct}}]}$ with $\hat{k}$. If $y = H_{\hat{k}}(\mathtt{ct})$ with $\mathtt{ct} \leftarrow\!\!\$\, \{0,1\}^{\ell_{\mathtt{ct}}}$, $\mathcal{B}$ simulates $\mathbf{H}_{1,i-1}$ for $\mathcal{A}$. Otherwise, $y \leftarrow\!\!\$\, \{0,1\}$ and $\mathcal{B}$ simulates $\mathbf{H}_{1,i}$. □

If the probability of $\mathtt{dm} \neq \perp$ in $\mathbf{H}_2$ is $\epsilon(n)$, that is,

$$\Pr\left[\mathsf{Dec}(\mathtt{dsk},\mathtt{ct}') \neq \perp : \begin{array}{l} \mathtt{pp} \leftarrow\!\!\$\, \mathsf{Setup}(1^n) \\ (\mathtt{dpk},\mathtt{dsk}) \leftarrow\!\!\$\, \mathsf{Gen}(\mathtt{pp}) \\ \mathtt{ct}' \leftarrow\!\!\$\, \{0,1\}^{\ell_{\mathtt{ct}}} \end{array}\right] = \epsilon(n),$$

then the advantage of $\mathcal{B}_1$ and $\mathcal{B}_2$ breaking the pseudo-randomness of PKE is $\epsilon(n)$. In specific, after receiving challenge ciphertext $\mathtt{ct}_b$, $\mathcal{B}_2$ samples $(\mathtt{pk}', \mathtt{sk}') \leftarrow\!\!\$\, \mathsf{Gen}(\mathtt{pp})$ and runs $\mathsf{Dec}(\mathtt{sk}', \mathtt{ct}_b)$. If $\mathtt{ct}_b$ is a valid ciphertext, by the robustness of PKE, the probability of $\mathsf{Dec}(\mathtt{sk}', \mathtt{ct}_b) \neq \perp$ is negligible. If $\mathtt{ct}_b$ is a random bit-string, the probability of $\mathsf{Dec}(\mathtt{sk}', \mathtt{ct}_b) \neq \perp$ is $\epsilon(n)$. By the pseudo-randomness of PKE, $\epsilon(n)$ must be a negligible function, and Equation 1 holds.

Let $\mathbf{G}$ denote the game for $\mathcal{A}$ in Equation 2. To prove this equation, we show how to break the robustness of PKE with $\mathcal{A}$ as follows.

Let $\mathcal{A}^*$ be the adversary in game $\mathsf{Rob}_{\mathsf{PKE},\mathcal{A}^*}(n)$. $\mathcal{A}^*$ receives $\mathtt{pp}$ from its challenger and makes at most q GEN queries. Then, $\mathcal{A}^*$ generates $(\mathtt{fpk}_i, \mathtt{fsk}_i)_{i\in[\ell_{\mathtt{ct}}]} \leftarrow\!\!\$\, \mathsf{Gen}(\mathtt{pp})$ and sets $\mathtt{dpk}$ as the public key $\mathtt{pk}_i$ of the i-th GEN query, where $i \leftarrow\!\!\$\, [q]$. After receiving $(\mathtt{FM}, \mathtt{dm})$ from $\mathcal{A}$, $\mathcal{A}^*$ sets $\mathtt{m}$ as $\mathtt{dm}$, picks random $j \leftarrow\!\!\$\, [q]$ and returns $(\mathtt{m}, i, j)$ to the challenger.

This simulation is perfect. If the probability of $\mathsf{dDec}(\mathtt{dsk}', \mathtt{CT}) \neq \perp$ in $\mathbf{G}$ is not negligible, the robustness of PKE does not hold. □

Theorem 4.4 (Security[2])**.** *Let* PKE *be a pseudo-random and robust PKE. Then,* PKE *is a secure* ℓ_{ct}*-sender-AME with extension algorithms in Fig. 4.*

Proof. By the correctness of PKE, one can easily verify that PKE is ℓ_{ct}-sender-anamorphic with extension algorithms in Fig. 4. The pseudo-randomness of PKE implies CPA security.

We now prove that for any PPT adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ distinguishing games $\mathsf{Ideal}_{\ell_{\mathsf{ct}},\mathsf{PKE},\mathcal{A}}(n)$ and $\mathsf{Real}^{\mathsf{fRandom}}_{\ell_{\mathsf{ct}},\mathsf{PKE},\mathcal{A}}(n)$ is negligible. Assume that $\mathcal{A}$ makes at most q encryption queries (providing a forced plaintext set $\mathtt{FM}$ and a duplicate plaintext $\mathtt{dm}$ each time).

Let $\mathbf{H}_0 = \mathsf{Real}^{\mathsf{fRandom}}_{\ell_{\mathsf{ct}},\mathsf{PKE},\mathcal{A}}(n)$, and $\mathbf{H}_i$ is the same as $\mathbf{H}_{i-1}$ except that challenger samples $(r_j)_{j\in[\ell_{\mathtt{ct}}]} \leftarrow\!\!\$\, \mathcal{R}$ to encrypt $\{\mathtt{fm}_j\}_{j\in[\ell_{\mathtt{ct}}]}$ in the i-th encryption query for $i \in [q]$. We have $\mathbf{H}_q = \mathsf{Ideal}_{\ell_{\mathsf{ct}},\mathsf{PKE},\mathcal{A}}(n)$.

Lemma 4.5. *By the pseudo-randomness of* PKE*,* $\mathbf{H}_{i-1} \approx_c \mathbf{H}_i$ *for* $i \in [q]$*.*

Proof. Let $\mathbf{H}^*_{i-1}$ be the same as $\mathbf{H}_{i-1}$ except that $\mathtt{ct}$ is uniformly sampled from $\{0,1\}^{\ell_{\mathtt{ct}}}$ in $\mathsf{fRandom}$. Let $\mathbf{H}^*_{i-1,0} = \mathbf{H}^*_{i-1}$ and $\mathbf{H}^*_{i-1,j}$ be the same as $\mathbf{H}^*_{i-1,j-1}$ except that, in $\mathsf{fRandom}$, the

[2]The proof has been revised to specify that PKE is only provable to be secure rather than being strongly secure (although we do not find valid attack) as claimed in the conference version. The main difficulty for proving strongly security is that adversary $\mathcal{B}_1^{i,j}$, who sets the public key in pseudo-randomness game as the j-th forced public key, is unable to provide the j-th forced secret key to $\mathcal{A}$.

ciphertext of $\mathtt{fm}_j$ is replaced to random string sampled from $\{0,1\}^{\ell_{\mathsf{ct}}}$ for $j \in [\ell_{\mathsf{ct}}]$. In $\mathbf{H}^*_{i-1,\ell_{\mathsf{ct}}}$, the sampling of $(r_j)_{j\in[\ell_{\mathsf{ct}}]}$ is independent of $\mathtt{FPK}, \mathtt{FM}, \mathtt{dpk}, \mathtt{dm}$. Thus, $\mathbf{H}^*_{i-1,\ell_{\mathsf{ct}}}$ is equivalent to $\mathbf{H}_i$.

To prove $\mathbf{H}_{i-1} \approx_c \mathbf{H}^*_{i-1}$, we build adversary $\mathcal{B}^i_1$ and $\mathcal{B}^i_2$ to break the pseudo-randomness of PKE. In specific, $\mathcal{B}^i_1$ receives $(\mathtt{pp}, \mathtt{pk})$ and returns challenge plaintext $\mathtt{m}^*$ and state $\mathtt{st}$. $\mathcal{B}^i_2$ receives $(\mathtt{st}, \mathtt{ct}_b)$ and has to guess the bit b.

Adversary $\mathcal{B}^i_1$ and $\mathcal{B}^i_2$ simulate $\mathbf{H}_{i-1}$ or $\mathbf{H}^*_{i-1}$ for $\mathcal{A}$ as follows.

- $\mathcal{B}^i_1$ runs $\mathsf{Gen}(\mathtt{pp})$ to generate ℓ_{ct} pairs of forced public/secret key $(\mathtt{fpk}_j, \mathtt{fsk}_j)$, and sets $\mathtt{pk}$ as the duplicate public key $\mathtt{dpk}$;
- $\mathcal{B}^i_1$ provides $(\mathtt{pp}, \mathtt{FPK} := \{\mathtt{fpk}_j\}_{j\in[\ell_{\mathsf{ct}}]})$ to $\mathcal{A}$;
- Let $(\mathtt{FM}, \mathtt{dm})$ be the i-th encryption query by $\mathcal{A}$. $\mathcal{B}^i_1$ returns $\mathtt{dm}$ and $\mathtt{st} = (\mathtt{pp}, \mathtt{dpk}, \mathtt{FPK}, \mathtt{FSK})$;
- $\mathcal{B}^i_2$ answers the first $(i-1)$ encryption queries by encrypting $\mathtt{FM}$ with randomnesses generated by $\mathsf{fRandom}(\mathtt{FPK}, \mathtt{FM}, \mathtt{dpk}, \mathtt{dm})$ in Fig. 4, and the last $(q-i)$ queries with randomnesses uniformly sampled from $\mathcal{R}$;
- $\mathcal{B}^i_2$ answers the i-th query by encrypting $\mathtt{FM}$ with R generated by $\mathsf{fRandom}'(\mathtt{FPK}, \mathtt{FM}, \mathtt{ct}_b)$ which is the same as $\mathsf{fRandom}(\mathtt{FPK}, \mathtt{FM}, \mathtt{dpk}, \mathtt{dm})$ except that $\mathtt{ct}$ is replaced by $\mathtt{ct}_b$.

If $\mathtt{ct}_b$ is an encryption of $\mathtt{dm}$ under $\mathtt{pk}$, then $\mathsf{fRandom}'(\mathtt{FPK}, \mathtt{FM}, \mathtt{ct}_b)$ is actually the same as $\mathsf{fRandom}(\mathtt{FPK}, \mathtt{FM}, \mathtt{dpk}, \mathtt{dm})$, and $\mathcal{B}^i_1, \mathcal{B}^i_2$ simulate $\mathbf{H}_{i-1}$ for $\mathcal{A}$.

If $\mathtt{ct}_b$ is uniformly sampled from $\{0,1\}^{\ell_{\mathsf{ct}}}$, then $\mathcal{B}^i_1, \mathcal{B}^i_2$ simulate $\mathbf{H}^*_{i-1}$ for $\mathcal{A}$.

Similarly, we can build adversary $\mathcal{B}^{i,j}_1$ and $\mathcal{B}^{i,j}_2$ to break the pseudo-randomness of PKE to prove $\mathbf{H}^*_{i-1,j-1} \approx_c \mathbf{H}^*_{i-1,j}$. In specific, $\mathcal{B}^{i,j}_1$ receives $(\mathtt{pp}, \mathtt{pk})$, sets $\mathtt{pk}$ as the j-th forced public key for $\mathcal{A}$ and returns $\mathtt{fm}_j$ in the i-th encryption query by $\mathcal{A}$. $\mathcal{B}^{i,j}_2$ answers the i-th query by encrypting $\mathtt{FM}$ with R generated by $\mathsf{fRandom}''(\{\mathtt{fpk}_k, \mathtt{fm}_k\}_{k\in[j+1,\ell_{\mathsf{ct}}]}, \mathtt{ct}_b)$, which is the same as $\mathsf{fRandom}(\mathtt{FPK}, \mathtt{FM}, \mathtt{dpk}, \mathtt{dm})$ except that the ciphertexts of $\mathtt{dm}$ and $\{\mathtt{fm}_k\}_{k\in[j-1]}$ are replaced to random string from $\{0,1\}^{\ell_{\mathsf{ct}}}$ and the ciphertext of $\mathtt{fm}_j$ is replaced to $\mathtt{ct}_b$.

If $\mathtt{ct}_b$ is an encryption of $\mathtt{fm}_j$ under $\mathtt{pk}$, $\mathcal{B}^{i,j}_1$ and $\mathcal{B}^{i,j}_2$ simulates $\mathbf{H}^*_{i-1,j-1}$ for $\mathcal{A}$. Otherwise, the simulation is $\mathbf{H}^*_{i-1,j}$. □

By Lemma 4.5, we have $\mathbf{H}_0 \approx_c \mathbf{H}_q$, from which this theorem follows. □

5 Construction II: Hybrid PKE with Special KEM

In this section, we demonstrate that a wide range of hybrid PKE schemes are strongly secure and robust $(\ell+1)$-sender-AME for duplicate plaintext with ℓ bits. We first depict such PKE schemes with a generic framework, then provide the details of their sender-anamorphic extension, and finally prove the properties of strong security and robustness rigorously.

5.1 Hybrid PKE with Special KEM

To better describe the proposed sender-anamorphic extension for hybrid PKE, we recall the module-level syntax of KEM and related properties (i.e., universal decryptability and key-pseudo-randomness) by Chen et al. [10] as below. In fact, these two properties are the "perfect correctness" and "CPA security" for typical KEM respectively.

- $\mathsf{KEM.Setup}(1^n)$ returns the public parameter $\mathtt{pp}$ including the key space $\mathcal{K}_{\mathsf{KEM}}$ and randomness space $\mathcal{R}_{\mathsf{KEM}}$.

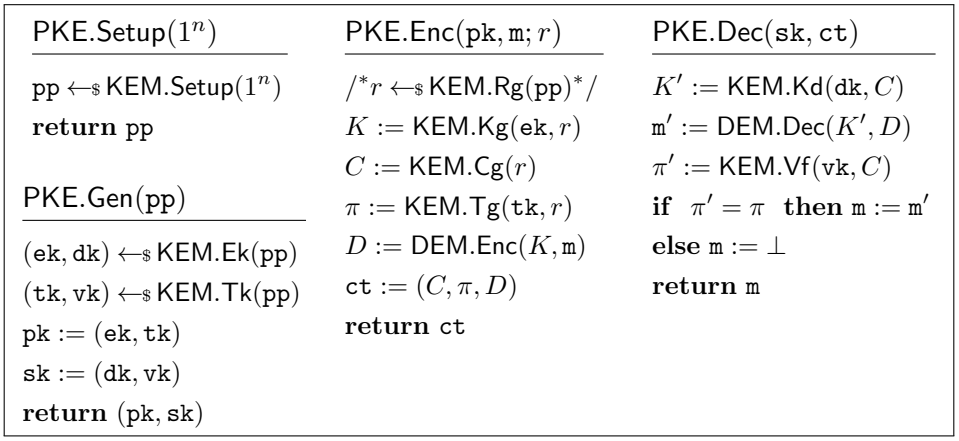

Figure 5: Hybrid PKE built on module-level KEM and DEM.

- $\mathsf{KEM.Gen}(\mathtt{pp})$ returns a key pair $(\mathtt{pk} = (\mathtt{ek}, \mathtt{tk}), \mathtt{sk} = (\mathtt{dk}, \mathtt{vk}))$ by running following sub-algorithms.
  - $\mathsf{KEM.Ek}(\mathtt{pp})$ produces an en/decapsulation key pair $(\mathtt{ek}, \mathtt{dk})$.
  - $\mathsf{KEM.Tk}(\mathtt{pp})$ produces a tag generation/verification key pair $(\mathtt{tk}, \mathtt{vk})$.
- $\mathsf{KEM.Enc}(\mathtt{pk})$ returns key K and ciphertext $\psi = (C, \pi)$ by running following sub-algorithms.
  - $\mathsf{KEM.Rg}(\mathtt{pp})$ picks a randomness r from $\mathcal{R}_{\mathsf{KEM}}$.
  - $\mathsf{KEM.Kg}(\mathtt{ek}, r)$ produces a key $K \in \mathcal{K}_{\mathsf{KEM}}$.
  - $\mathsf{KEM.Cg}(r)$ produces a ciphertext C of key K.
  - $\mathsf{KEM.Tg}(\mathtt{tk}, r)$ produces the ciphertext tag π of C.
- $\mathsf{KEM.Dec}(\mathtt{sk}, \psi = (C, \pi))$ returns the key K or $\perp$ by running following sub-algorithms.
  - $\mathsf{KEM.Kd}(\mathtt{dk}, C)$ produces a key K.
  - $\mathsf{KEM.Vf}(\mathtt{vk}, C)$ produces a ciphertext tag π'.

  If $\pi' = \pi$, $\mathsf{KEM.Dec}$ returns K. Otherwise, it returns $\perp$.

We remark that algorithms related to the ciphertext tag (including $\mathsf{KEM.Tk}$, $\mathsf{KEM.Tg}$ and $\mathsf{KEM.Vf}$) are optional and do not appear in certain KEMs.

Definition 5.1 (Universal Decryptability [10])**.** Let KEM be a module-level KEM. We say KEM satisfies universal decryptability if for any $n \in \mathbb{N}$,

$$\Pr\left[\mathsf{KEM.Kg}(\mathtt{ek}, r) \neq \mathsf{KEM.Kd}(\mathtt{dk}, C) : \begin{array}{l} \mathtt{pp} \leftarrow\!\!\$\ \mathsf{KEM.Setup}(1^n) \\ (\mathtt{ek}, \mathtt{dk}) \leftarrow\!\!\$\ \mathsf{KEM.Ek}(\mathtt{pp}) \\ r \leftarrow\!\!\$\ \mathsf{KEM.Rg}(\mathtt{pp}) \\ C := \mathsf{KEM.Cg}(r) \end{array}\right] \leq \mathsf{negl}(n)\,.$$

Definition 5.2 (Key-Pseudo-Randomness [10])**.** Let KEM be a module-level KEM. We say KEM is key-pseudo-random if for any PPT adversary $\mathcal{A}$,

$$\left|\Pr\left[b = b' : \begin{array}{l} \mathtt{pp} \leftarrow_{\$} \mathsf{KEM.Setup}(1^n) \\ (\mathtt{ek}, \mathtt{dk}) \leftarrow_{\$} \mathsf{KEM.Ek}(\mathtt{pp}) \\ r \leftarrow_{\$} \mathsf{KEM.Rg}(\mathtt{pp}) \\ C := \mathsf{KEM.Cg}(r) \\ b \leftarrow_{\$} \{0,1\}; K_0 \leftarrow_{\$} \mathcal{K}_{\mathsf{KEM}} \\ K_1 := \mathsf{KEM.Kg}(\mathtt{ek}, r) \\ b' \leftarrow \mathcal{A}(\mathtt{pp}, \mathtt{ek}, K_b, C) \end{array}\right] - \frac{1}{2}\right| \leq \mathsf{negl}(n)\,.$$

Here we introduce the homomorphic property for module-level KEM.

Definition 5.3 (Homomorphic Property)**.** Let KEM be a module-level KEM with key space $\mathcal{K}_{\mathsf{KEM}}$ and randomness space $\mathcal{R}_{\mathsf{KEM}}$. We say KEM is homomorphic if for any $n \in \mathbb{N}^+$, $\mathtt{pp} \leftarrow_{\$} \mathsf{KEM.Setup}(1^n)$, $(\mathtt{ek}, \mathtt{dk}) \leftarrow_{\$} \mathsf{KEM.Ek}(\mathtt{pp})$, any $r_1, r_2 \in \mathcal{R}_{\mathsf{KEM}}$,

1) $\mathsf{KEM.Kg}(\mathtt{ek}, r_1 \oplus r_2) = \mathsf{KEM.Kg}(\mathtt{ek}, r_1) \odot \mathsf{KEM.Kg}(\mathtt{ek}, r_2)$ where $\oplus$ and $\odot$ are operations defined over $\mathcal{R}_{\mathsf{KEM}}$ and $\mathcal{K}_{\mathsf{KEM}}$ respectively, and

2) $\mathsf{KEM.Cg}(r_1 \oplus r_2) = \mathsf{KEM.Cg}(r_1) \otimes \mathsf{KEM.Cg}(r_2)$ where $\otimes$ is an operation defined over the support of $\mathsf{KEM.Cg}(r)$.

Let KEM be a module-level KEM satisfying universal decryptability, key-pseudo-randomness and homomorphic property, and $\mathsf{DEM} = (\mathsf{Enc}, \mathsf{Dec})$ be the data encapsulation mechanism (DEM) that is a symmetric encryption scheme that has indistinguishable encryptions in the presence of an eavesdropper (see Appendix A.2). Fig. 5 shows the hybrid PKE PKE built on KEM and DEM.

Some KEMs including Cramer-Shoup KEMs under DDH, DCR and QR assumptions [11], Kurosawa-Desmedt KEM [20] and Hofheinz-Kiltz KEM [17] have been demonstrated to be universally decryptable and key-pseudo-random in [10]. One can refer to the details of these KEMs in [10] and verify that all these KEMs also satisfy the homomorphic property. Analogous to the DDH-based Cramer-Shoup KEM, the ElGamal KEM also meets the requirement.

5.2 Sender-Anamorphic Extension

Fig. 6 depicts the details of sender-anamorphic extension for the above hybrid PKE. Also, this extension requires an entropy smoothing keyed hash function $H_{\hat{k}}$ that maps the key of KEM into the randomness space $\mathcal{R}_{\mathsf{KEM}}$. We assume $\mathcal{R}_{\mathsf{KEM}}$ is an additive and cyclic group and $1_{\mathcal{R}_{\mathsf{KEM}}}$ denotes a generator of $\mathcal{R}_{\mathsf{KEM}}$.

We note that some cryptosystems are built over such hybrid PKE (e.g., Naor-Yung [24] and double-strand [16, 27, 33] paradigms) are strongly secure and robust 1-sender-AMEs for 1-bit duplicate message. See Appendix B for more details.

5.3 Security Analysis

Let PKE be a hybrid PKE in Fig. 5 with KEM satisfying universal decryptability, key-pseudo-randomness and homomorphic property and DEM that has indistinguishable encryptions in the presence of an eavesdropper.

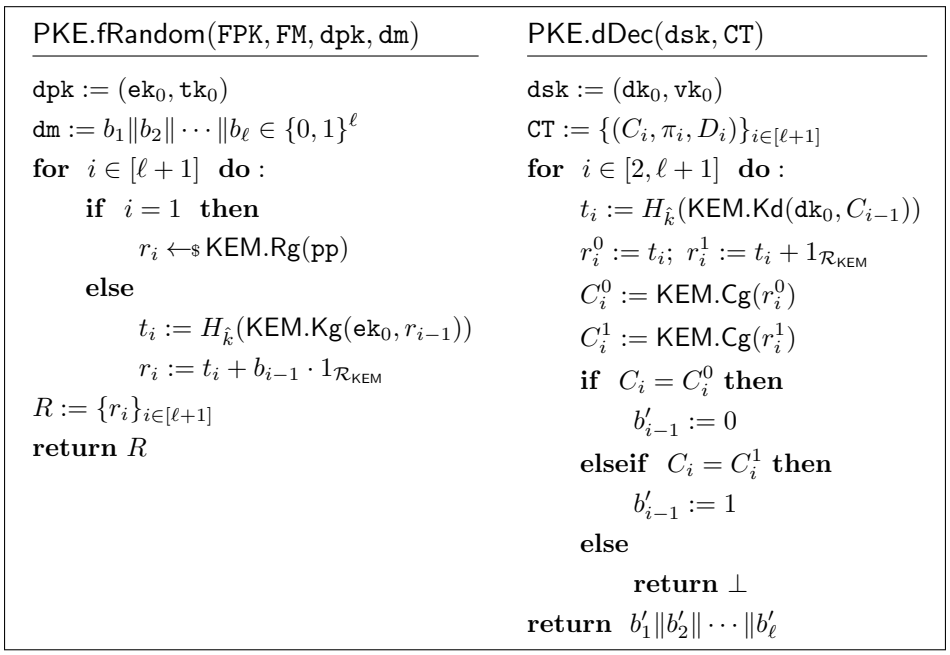

Figure 6: Sender-anamorphic extension for hybrid PKE with special KEM.

Theorem 5.1 (Robustness[3])**.** PKE *in Fig. 5 is a robust* $(\ell+1)$*-sender-AME with extension algorithms in Fig. 6.*

Proof. Let $\mathbf{H}_0$ denote the game for $\mathcal{A}$ in Equation 1. Game $\mathbf{H}_1$ is the same as $\mathbf{H}_0$ except that $t_i := H_{\hat{k}}(\mathsf{KEM.Kg}(\mathtt{ek}_0, r_{i-1}))$ in dDec for every $i \in [2, \ell+1]$, instead of $t_i := H_{\hat{k}}(\mathsf{KEM.Kd}(\mathtt{dk}_0, C_{i-1}))$. Since the challenger performs the encryption of FM, it is possible to obtain all the randomnesses and compute t_i using $\mathtt{ek}_0$. By the universal decryptability of KEM, game $\mathbf{H}_1$ is statistically indistinguishable from $\mathbf{H}_0$.

Game $\mathbf{H}_2$ is the same as $\mathbf{H}_1$ except that $t_i := H_{\hat{k}}(K_i)$ with $K_i \leftarrow\!\!\$\, \mathcal{K}_{\mathsf{KEM}}$ for every $i \in [2, \ell+1]$.

Lemma 5.2. *By the key-pseudo-randomness of* KEM, $\mathbf{H}_1 \approx_c \mathbf{H}_2$.

Proof. Let $\mathbf{H}_{1,1} = \mathbf{H}_1$ and $\mathbf{H}_{1,i}$ be the same as $\mathbf{H}_{1,i-1}$ except that $t_i := H_{\hat{k}}(K_i)$ with $K_i \leftarrow\!\!\$\, \mathcal{K}_{\mathsf{KEM}}$ and $i \in [2, \ell+1]$. We have $\mathbf{H}_2 = \mathbf{H}_{1,\ell+1}$. To prove $\mathbf{H}_{1,i-1} \approx_c \mathbf{H}_{1,i}$, we show how to break the key-pseudo-randomness of KEM by $\mathcal{D}_{i-1,1}$ and $\mathcal{D}_{i-1,2}$ distinguishing $\mathbf{H}_{1,i-1}$ and $\mathbf{H}_{1,i}$ with non-negligible advantage in Fig. 7. Specifically, ciphertext $\mathtt{ct}_{i-1}$ derived from C^* using secret key $\mathtt{fsk}_{i-1}$ is valid. If $K^* = \mathsf{KEM.Kg}(\mathtt{ek}^*, r^*)$ and r^* is the underlying randomness of C^*, $\mathcal{B}_{i-1}$ simulates $\mathbf{H}_{1,i-1}$. If $K^* \leftarrow\!\!\$\, \mathcal{K}_{\mathsf{KEM}}$, $\mathcal{B}_{i-1}$ simulates $\mathbf{H}_{1,i}$. $\mathcal{B}_{i-1}$ breaks the key-pseudo-randomness of KEM by forwarding the bit b outputted by $\mathcal{D}_{i-1,2}$. □

Game $\mathbf{H}_3$ is the same as $\mathbf{H}_2$ except that $t_i \leftarrow\!\!\$\, \mathcal{R}_{\mathsf{KEM}}$ for every $i \in [2, \ell+1]$.

Lemma 5.3. *By the entropy smoothness of hash function family* $\{H_{\hat{k}}\}_{\hat{k}\in\hat{\mathcal{K}}}$, $\mathbf{H}_2 \approx_c \mathbf{H}_3$.

Proof. Let $\mathbf{H}_{2,1} = \mathbf{H}_2$ and $\mathbf{H}_{2,i}$ be the same as $\mathbf{H}_{2,i-1}$ except that $t_i \leftarrow\!\!\$\, \mathcal{R}_{\mathsf{KEM}}$ with $i \in [2, \ell+1]$. We have $\mathbf{H}_3 = \mathbf{H}_{2,\ell+1}$. If there exists adversary $\mathcal{D}_{i-1,1}$ and $\mathcal{D}_{i-1,2}$ distinguishing $\mathbf{H}_{2,i-1}$ and $\mathbf{H}_{2,i}$ with overwhelming advantage, we show $\mathcal{B}_{i-1}$ that breaks the entropy smoothness of $\{H_{\hat{k}}\}_{\hat{k}\in\hat{\mathcal{K}}}$. In particular, let $(\hat{k}^*, y^*)$ be the instance, $\mathcal{B}_{i-1}$ simulates the game for $\mathcal{D}_{i-1,1}$ and computes CT

[3]Compared with the conference version, the proof has been enhanced with more details.

$\mathcal{B}_{i-1}(\mathsf{pp}, \mathtt{ek}^*, K^*, C^*)$

$(\mathtt{fpk}_i, \mathtt{fsk}_i)_{i\in[\ell+1]} \leftarrow\!\!\$\ \mathsf{Gen}(\mathsf{pp})$
$(\mathtt{FM}, \mathtt{st}) \leftarrow \mathcal{D}_{i-1,1}(\mathsf{pp}, \mathtt{FPK})$
$(r_j)_{j\in[\ell+1]\setminus\{i-1\}} \leftarrow\!\!\$\ \mathcal{R}$
$\mathtt{ct}_{i-1} := \mathsf{Enc}^*(\mathtt{fsk}_{i-1}, C^*, \mathtt{fm}_{i-1})$
$\mathtt{CT} := \{\mathsf{Enc}(\mathtt{fpk}_j, \mathtt{fm}_j; r_j)\}_{j\in[\ell+1]\setminus\{i-1\}} \cup \{\mathtt{ct}_{i-1}\}$
$\mathsf{flag} := \mathsf{dDec}^*(\mathtt{ek}^*, (r_j)_{j\in[i,\ell]}, \mathtt{CT}) \neq \bot$
$b \leftarrow \mathcal{D}_{i-1,2}(\mathsf{flag}, \mathtt{st})$
return b

$\mathsf{Enc}^*(\mathtt{sk}, C, \mathtt{m})$

$\mathtt{sk} := (\mathtt{dk}, \mathtt{vk})$
$\pi := \mathsf{KEM.Vf}(\mathtt{vk}, C)$
$K := \mathsf{KEM.Kd}(\mathtt{dk}, C)$
$D := \mathsf{DEM.Enc}(K, \mathtt{m})$
$\mathtt{ct} := (C, \pi, D)$
return $\mathtt{ct}$

$\mathsf{dDec}^*(\mathtt{ek}^*, (r_j)_{j\in[i,\ell]}, \mathtt{CT})$

$\mathtt{CT} := \{(C_i, \pi_i, D_i)\}_{i\in[\ell+1]}$
for $j \in [2, \ell+1]$ **do**:
  if $j < i$ **then**
    $K_j \leftarrow\!\!\$\ \mathcal{K}_{\mathsf{KEM}}$
    $t_j := H_{\hat{k}}(K_j)$
  elseif $j = i$ **then**
    $t_j := H_{\hat{k}}(K^*)$
  else
    $t_j := H_{\hat{k}}(\mathsf{KEM.Kg}(\mathtt{ek}^*, r_{j-1}))$
  $r_j^0 := t_j;\ r_j^1 := t_j + 1_{\mathcal{R}_{\mathsf{KEM}}}$
  $C_j^0 := \mathsf{KEM.Cg}(r_j^0)$
  $C_j^1 := \mathsf{KEM.Cg}(r_j^1)$
  if $C_j = C_j^0$ **then**
    $b'_{j-1} := 0$
  elseif $C_j = C_j^1$ **then**
    $b'_{j-1} := 1$
  else
    return $\bot$
return $b'_1 \| b'_2 \| \cdots \| b'_\ell$

Figure 7: Adversary $\mathcal{B}_{i-1}$ breaking the key-pseudo-randomness of KEM in the proof of Lemma 5.2

as in $\mathbf{H}_0$. Recall that, in $\mathbf{H}_2$, $t_j := H_{\hat{k}}(K_j)$ with $K_j \leftarrow\!\!\$\ \mathcal{K}_{\mathsf{KEM}}$ for every $i \in [2, \ell+1]$. Here, $\mathcal{B}_{i-1}$ samples $t_j \leftarrow\!\!\$\ \mathcal{R}_{\mathsf{KEM}}$ for $j \in [2, i-1]$, sets $t_i := y^*$ and computes $t_j := H_{\hat{k}^*}(K_j)$ with $K_j \leftarrow\!\!\$\ \mathcal{K}_{\mathsf{KEM}}$ for $j \in [i+1, \ell+1]$. If $y^* \leftarrow\!\!\$\ \mathcal{R}_{\mathsf{KEM}}$, then $\mathcal{B}_{i-1}$ simulates $\mathbf{H}_{2,i}$. Otherwise, $\mathcal{B}_{i-1}$ simulates $\mathbf{H}_{2,i-1}$. □

Recall that the output of dDec is not $\bot$ when $C_i = C_i^0$ or $C_i = C_i^1$ for any $i \in [2, \ell+1]$. That is, $r_i = t_i$ or $r_i = t_i + 1$ for any $i \in [2, \ell+1]$ by the homomorphic property of KEM. Otherwise, $\mathsf{KEM.Cg}(r_i - t_i)$ or $\mathsf{KEM.Cg}(r_i - (t_i + 1))$ is the identity in the support of KEM.Cg, while $r_i - t_i$ or $r_i - (t_i + 1)$ is not 0. In $\mathbf{H}_3$, both r_i and t_i are uniformly sampled from $\mathcal{R}_{\mathsf{KEM}}$. The probability that dDec does not return $\bot$ is at most $(2/|\mathcal{R}_{\mathsf{KEM}}|)^\ell$ which is negligible over n.

Let $\mathbf{G}$ denote the game for $\mathcal{A}$ in Equation 2. In game $\mathbf{G}$, $r_i := H_{\hat{k}}(\mathsf{KEM.Kg}(\mathtt{ek}_0, r_{i-1})) + b_{i-1} \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}$ for $i \in [2, \ell+1]$. Let $\mathtt{dsk}' := (\mathtt{dk}'_0, \mathtt{vk}'_0)$, then $t_i := H_{\hat{k}}(\mathsf{KEM.Kd}(\mathtt{dk}'_0, C_{i-1}))$. By the universal decryptability of KEM, $t_i := H_{\hat{k}}(\mathsf{KEM.Kg}(\mathtt{ek}'_0, r_{i-1}))$. Since both $\mathtt{dpk}$ and $\mathtt{dpk}'$ are randomly generated, the probability that $\mathtt{ek}_0 = \mathtt{ek}'_0$ is negligible. Obviously, $\mathsf{KEM.Kg}(\mathtt{ek}_0, r_{i-1})) \neq \mathsf{KEM.Kg}(\mathtt{ek}'_0, r_{i-1}))$. By the entropy smoothness of $\{H_{\hat{k}}\}_{\hat{k}\in\hat{\mathcal{K}}}$, the probability that $r_i = t_i$ or $r_i = t_i + 1$ is negligible. □

Theorem 5.4 (Security)**.** PKE *is a strongly secure* $(\ell+1)$*-sender-AME with extension algorithms in Fig. 6.*

Proof. We first prove that PKE is $(\ell+1)$-sender-anamorphic and claim that $b'_i = b_i$ for $i \in [\ell]$.

In $\mathtt{ct}_i := (C_i, \pi_i, D_i)$,

$$\begin{aligned} C_i &= \mathsf{KEM.Cg}(r_i) \\ &= \mathsf{KEM.Cg}(H_{\hat{k}}(\mathsf{KEM.Kg}(\mathtt{ek}_0, r_{i-1})) + b_{i-1} \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}) \\ &= \mathsf{KEM.Cg}(H_{\hat{k}}(\mathsf{KEM.Kd}(\mathtt{dk}_0, C_{i-1})) + b_{i-1} \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}) \\ &= \mathsf{KEM.Cg}(r_i^{b_{i-1}}) = C_i^{b_{i-1}}, \end{aligned}$$

where $i \in [2, \ell+1]$. By the universal decryptability of KEM, the probability of event that the third equality does not hold is negligible. Note that if $b_{i-1} = 0$ then $C_i = C_i^0$ and $b'_{i-1} = 0$. Otherwise, we have $b_{i-1} = 1$, $C_i = C_i^1$ and $b'_{i-1} = 1$.

Since KEM is key-pseudo-random and DEM has indistinguishable encryptions in the presence of an eavesdropper, PKE is CPA secure. We defer the proof of CPA security to Appendix A.3.

We now prove that for any PPT adversary $\mathcal{A}$, the advantage of adversary $\mathcal{A}$ distinguishing games $\mathsf{Ideal}_{\ell+1,\mathsf{PKE},\mathcal{A}}(n)$ and $\mathsf{Real}^{\mathsf{fRandom}}_{\ell+1,\mathsf{PKE},\mathcal{A}}(n)$ is negligible. Assume that adversary $\mathcal{A}$ makes at most q encryption queries, and provides a forced plaintext set $\mathtt{FM} := \{\mathtt{fm}_i\}_{i\in[\ell+1]}$ and a duplicate plaintext $\mathtt{dm}$ in each query.

Let $\mathbf{H}_0 = \mathsf{Real}^{\mathsf{fRandom}}_{\ell+1,\mathsf{PKE},\mathcal{A}}(n)$, and $\mathbf{H}_i$ is the same as $\mathbf{H}_{i-1}$ except that challenger samples $(r_i)_{i\in[\ell+1]} \leftarrow\!\!\$\, \mathcal{R}_{\mathsf{KEM}}$ to encrypt forced messages $\{\mathtt{fm}_j\}_{j\in[\ell+1]}$ in the i-th encryption query for $i \in [q]$. Obviously, $\mathbf{H}_q = \mathsf{Ideal}_{\ell+1,\mathsf{PKE},\mathcal{A}}(n)$.

Let $\mathbf{H}_{i-1,0} = \mathbf{H}_{i-1}$ for $i \in [q]$, and $\mathbf{H}_{i-1,j}$ is the same as $\mathbf{H}_{i-1,j-1}$ except that the challenger samples $r_j \leftarrow\!\!\$\, \mathcal{R}_{\mathsf{KEM}}$ to encrypt $\mathtt{fm}_j$ in the i-th encryption query for $j \in [\ell+1]$. Obviously, $\mathbf{H}_{i-1,\ell+1} = \mathbf{H}_i$. Note that $r_1^* \leftarrow\!\!\$\, \mathsf{KEM.Rg}(\mathsf{pp})$, namely, $r_1^* \leftarrow\!\!\$\, \mathcal{R}_{\mathsf{KEM}}$, in algorithm $\mathsf{fRandom}$, we have $\mathbf{H}_{i-1,0} = \mathbf{H}_{i-1,1}$ for $i \in [q]$.

Let $\mathbf{H}'_{i-1,j-1}$ be the same as $\mathbf{H}_{i-1,j-1}$ except that the challenger computes $r_j := H_{\hat{k}}(K) + b_{j-1} \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}$, where $K \leftarrow\!\!\$\, \mathcal{K}_{\mathsf{KEM}}$ and b_{j-1} is the $(j-1)$-th bit of $\mathtt{dm}$, to encrypt $\mathtt{fm}_j$ in the i-th encryption query for $j \in [2, \ell+1]$.

Now, there is a series of games between $\mathbf{H}_0$ and $\mathbf{H}_q$ as below.

$$\begin{array}{lllllll} \{ & \mathbf{H}_0(\mathbf{H}_{0,0}), & \mathbf{H}_{0,1}, & \mathbf{H}'_{0,1}, & \cdots, & \mathbf{H}_{0,\ell}, & \mathbf{H}'_{0,\ell}, \\ & \mathbf{H}_1(\mathbf{H}_{0,\ell+1}, \mathbf{H}_{1,0}), & \mathbf{H}_{1,1}, & \mathbf{H}'_{1,1}, & \cdots, & \mathbf{H}_{1,\ell}, & \mathbf{H}'_{1,\ell}, \\ & \cdots, & & & & & \\ & \mathbf{H}_{q-1}(\mathbf{H}_{q-2,\ell+1}, \mathbf{H}_{q-1,0}), & \mathbf{H}_{q-1,1}, & \mathbf{H}'_{q-1,1}, & \cdots, & \mathbf{H}_{q-1,\ell}, & \mathbf{H}'_{q-1,\ell}, \\ & \mathbf{H}_q(\mathbf{H}_{q-1,\ell+1})\} & & & & & \end{array}$$

Lemma 5.5. *By the key-pseudo-randomness of* KEM, $\mathbf{H}'_{i-1,j-1} \approx_c \mathbf{H}_{i-1,j-1}$ *for* $i \in [q]$ *and* $j \in [2, \ell+1]$.

Proof. We show how to build an adversary $\mathcal{D}_{i-1,j-1}$ breaking the key-pseudo-randomness of module-level KEM. Specifically, $\mathcal{D}_{i-1,j-1}$ receives $(\mathtt{pp}, \mathtt{ek}^*, K^*, C^*)$ and has to guess the bit b. If $b = 0$, K^* is uniformly sampled from $\mathcal{K}_{\mathsf{KEM}}$. Otherwise, $K^* = \mathsf{KEM.Kg}(\mathtt{ek}^*, r)$ with $r \leftarrow\!\!\$\, \mathsf{KEM.Rg}(\mathsf{pp})$ and $C^* = \mathsf{KEM.Cg}(r)$.

Adversary $\mathcal{D}_{i-1,j-1}$ simulates $\mathbf{H}_{i-1,j-1}$ or $\mathbf{H}'_{i-1,j-1}$ for $\mathcal{A}$ as shown in Fig. 8. In particular, the duplicate public key $\mathtt{dpk}^*$ is a valid public key, as the generation of encapsulation key $\mathtt{ek}$ and tag generation key $\mathtt{tk}$ in PKE is independent.

$\mathcal{D}_{i-1,j-1}$ runs algorithm dEnc to generate ciphertexts $\{\mathtt{ct}_i\}_{i\in[\ell+1]}$ for the i-th encryption query. In $\mathtt{ct}_{j-1}$, $C_{j-1} = C^* + b_{j-2} \cdot \mathsf{KEM.Cg}(1_{\mathcal{R}_{\mathsf{KEM}}})$, $D_{j-1} = \mathsf{DEM.Enc}(K_{j-1}, \mathtt{fm}_{j-1})$, π_{j-1}

$\mathcal{D}_{i-1,j-1}(\mathtt{pp}, \mathtt{ek}^*, K^*, C^*)$

$(\mathtt{fpk}_i, \mathtt{fsk}_i)_{i\in[\ell+1]} \leftarrow\!\!\$\ \mathsf{Gen}(\mathtt{pp})$
$((\mathtt{ek}_0, \mathtt{tk}_0), \mathtt{dsk}) \leftarrow\!\!\$\ \mathsf{Gen}(\mathtt{pp})$
$\mathtt{dpk}^* := (\mathtt{ek}^*, \mathtt{tk}_0)$
$b \leftarrow \mathcal{A}^{\mathrm{ENC}(\cdot,\cdot)}(\mathtt{pp}, \mathtt{FPK}, \mathtt{FSK}, \mathtt{dpk}^*)$
return b

ENC($\mathtt{FM}, \mathtt{dm}$)

/*the k-th encryption query*/
if $k = i$ **then**
  return dEnc($\mathtt{FPK}, \mathtt{FM}, \mathtt{dpk}^*, \mathtt{dm}$)
if $k \le i-1$ **then**
  $(r_i)_{i\in[\ell+1]} \leftarrow\!\!\$\ \mathcal{R}_{\mathsf{KEM}}$
else $\cdots\cdots$ /*$k > i^*$/
  $R \leftarrow\!\!\$\ \mathsf{fRandom}(\mathtt{FPK}, \mathtt{FM}, \mathtt{dpk}^*, \mathtt{dm})$
return $\{\mathsf{Enc}(\mathtt{fpk}_i, \mathtt{fm}_i; r_i)\}_{i\in[\ell+1]}$

dEnc($\mathtt{FPK}, \mathtt{FM}, \mathtt{dpk}^*, \mathtt{dm}$)

$\mathtt{dm} := b_1 b_2 \cdots b_\ell \in \{0,1\}^\ell$; $b_0 := 0$
for $i \in [\ell+1]$ **do**:
  if $i < j-1$ **then**
    $r_i \leftarrow\!\!\$\ \mathcal{R}_{\mathsf{KEM}}$
    $\mathtt{ct}_i := \mathsf{Enc}(\mathtt{fpk}_i, \mathtt{fm}_i; r_i)$
  elseif $i = j-1$ **then**
    $C_i := C^* + b_{i-1} \cdot \mathsf{KEM.Cg}(1_{\mathcal{R}_{\mathsf{KEM}}})$
    $\pi_i := \mathsf{KEM.Vf}(\mathtt{vk}_i, C_i)$
    $K_i := \mathsf{KEM.Kd}(\mathtt{dk}_i, C_i)$
    $D_i := \mathsf{DEM.Enc}(K_i, \mathtt{fm}_i)$
    $\mathtt{ct}_i := (C_i, \pi_i, D_i)$
  elseif $i = j$ **then**
    $s_i := b_{i-2} \cdot \mathsf{KEM.Kg}(\mathtt{ek}^*, 1_{\mathcal{R}_{\mathsf{KEM}}})$
    $t_i := H_{\hat{k}}(K^* + s_i)$
    $r_i := t_i + b_{i-1} \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}$
    $\mathtt{ct}_i := \mathsf{Enc}(\mathtt{fpk}_i, \mathtt{fm}_i; r_i)$
  else $\cdots\cdots$ /*$i > j^*$/
    $t_i := H_{\hat{k}}(\mathsf{KEM.Kg}(\mathtt{ek}^*, r_{i-1}))$
    $r_i := t_i + b_{i-1} \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}$
    $\mathtt{ct}_i := \mathsf{Enc}(\mathtt{fpk}_i, \mathtt{fm}_i; r_i)$
return $\mathtt{CT} := \{\mathtt{ct}_i\}_{i\in[\ell+1]}$

Figure 8: Adversary $\mathcal{D}_{i-1,j-1}$ breaking the key-pseudo-randomness of KEM in the proof of Lemma 5.5.

and K_{j-1} are derived from C_{j-1} using $\mathtt{fsk}_{j-1} = (\mathtt{dk}_{j-1}, \mathtt{vk}_{j-1})$. Note that $C^* = \mathsf{KEM.Cg}(r)$, we have $C_{j-1} = \mathsf{KEM.Cg}(r_{j-1}) = \mathsf{KEM.Cg}(r + b_{j-2} \cdot 1_{\mathcal{R}_{\mathsf{KEM}}})$ by the homomorphic property of KEM.Cg, and $\mathtt{ct}_{j-1} = \mathsf{Enc}(\mathtt{fpk}_{j-1}, \mathtt{fm}_{j-1}; r_{j-1})$. In $\mathtt{ct}_j$, the underlying randomness $r_j = H_{\hat{k}}(K^* + b_{j-2} \cdot \mathsf{KEM.Kg}(\mathtt{ek}^*, 1_{\mathcal{R}_{\mathsf{KEM}}})) + b_{j-1} \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}$.

If $K^* = \mathsf{KEM.Kg}(\mathtt{ek}^*, r)$, by the homomorphic property of KEM.Kg,

$$\begin{aligned} r_j &= H_{\hat{k}}(\mathsf{KEM.Kg}(\mathtt{ek}^*, r + b_{j-2} \cdot 1_{\mathcal{R}_{\mathsf{KEM}}})) + b_{j-1} \cdot 1_{\mathcal{R}_{\mathsf{KEM}}} \\ &= H_{\hat{k}}(\mathsf{KEM.Kg}(\mathtt{ek}^*, r_{j-1})) + b_{j-1} \cdot 1_{\mathcal{R}_{\mathsf{KEM}}}. \end{aligned}$$

$\mathtt{ct}_j$ is computed in the same way as in fRandom, and $\mathcal{D}_{i-1,j-1}$ simulates $\mathbf{H}_{i-1,j-1}$ for $\mathcal{A}$. Otherwise, $K^* \leftarrow\!\!\$\ \mathcal{K}_{\mathsf{KEM}}$, $K^* + b_{j-2} \cdot \mathsf{KEM.Kg}(\mathtt{ek}^*, 1_{\mathcal{R}_{\mathsf{KEM}}})$ is uniformly distributed over $\mathcal{K}_{\mathsf{KEM}}$ and $\mathcal{D}_{i-1,j-1}$ simulates $\mathbf{H}'_{i-1,j-1}$ for $\mathcal{A}$. □

Lemma 5.6. *By the entropy smoothness of hash function family $\mathcal{H} = \{H_{\hat{k}}\}_{\hat{k}\in\hat{\mathcal{K}}}$ with $H_{\hat{k}} : \mathcal{K}_{\mathsf{KEM}} \to \mathcal{R}_{\mathsf{KEM}}$, $\mathbf{H}_{i-1,j} \approx_c \mathbf{H}'_{i-1,j-1}$ for $i \in [q]$ and $j \in [2, \ell+1]$.*

Proof. We show how to build an adversary $\mathcal{B}_{i-1,j-1}$ breaking the entropy smoothness of hash function family $\mathcal{H}$ with $H_{\hat{k}} : \mathcal{K}_{\mathsf{KEM}} \to \mathcal{R}_{\mathsf{KEM}}$. Specifically, $\mathcal{B}_{i-1,j-1}$ receives $(\hat{k}, y)$ and has to decide whether $y = H_{\hat{k}}(x)$ with $x \leftarrow\!\!\$\ \mathcal{K}_{\mathsf{KEM}}$ or $y \leftarrow\!\!\$\ \mathcal{R}_{\mathsf{KEM}}$.

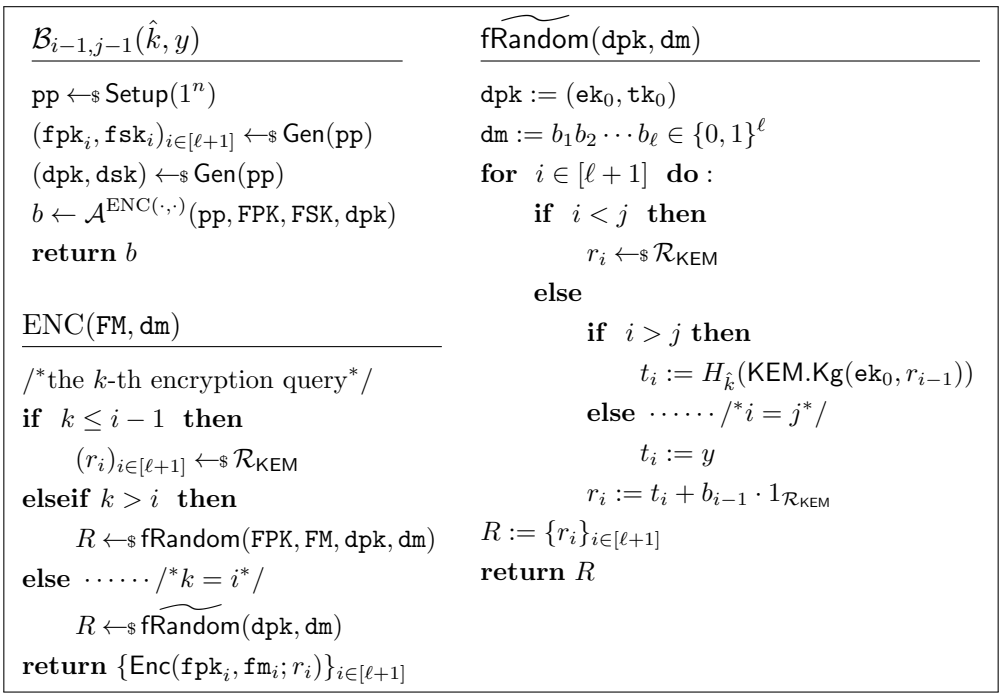

Figure 9: Adversary $\mathcal{B}_{i-1,j-1}$ breaking the entropy smoothness of hash function family $\mathcal{H}$ in the proof of Lemma 5.6.

Adversary $\mathcal{B}_{i-1,j-1}$ simulates $\mathbf{H}_{i-1,j-1}$ or $\mathbf{H}'_{i-1,j-1}$ for $\mathcal{A}$ as shown in Fig. 9. In particular, $\mathcal{B}_{i-1,j-1}$ runs algorithm $\widetilde{\mathsf{fRandom}}$ to generate R for the i-th encryption query. In algorithm $\widetilde{\mathsf{fRandom}}$, the value of t_j is set as y.

If y is uniformly sampled from $\mathcal{R}_{\mathsf{KEM}}$, then randomness r_j is also uniformly sampled from $\mathcal{R}_{\mathsf{KEM}}$, and $\mathcal{B}_{i-1,j-1}$ simulates $\mathbf{H}_{i-1,j}$ for $\mathcal{A}$. If $y = H_{\hat{k}}(x)$ with $x \leftarrow\!\!\$\ \mathcal{K}_{\mathsf{KEM}}$, then $\mathcal{B}_{i-1,j-1}$ simulates $\mathbf{H}'_{i-1,j-1}$ for $\mathcal{A}$. □

By Lemma 5.5, Lemma 5.6 and $\mathbf{H}_{i-1,0} = \mathbf{H}_{i-1,1}$ for $i \in [q]$, we have $\mathbf{H}_0 \approx_c \mathbf{H}_q$. That is, for any PPT adversary $\mathcal{A}$, the advantage of adversary $\mathcal{A}$ distinguishing games $\mathsf{Ideal}_{\ell+1,\mathsf{PKE},\mathcal{A}}(n)$ and $\mathsf{Real}^{\mathsf{fRandom}}_{\ell+1,\mathsf{PKE},\mathcal{A}}(n)$ is negligible. □

6 Relation between ℓ-Receiver/Sender-AME

6.1 ℓ-Receiver-Anamorphic Encryption (ℓ-Receiver-AME)

Definition 6.1 (ℓ-Receiver-Anamorphic Encryption)**.** Let PKE be a public key encryption scheme. We say PKE is ℓ-receiver-anamorphic if 1) there exists a receiver-anamorphic extension $(\mathsf{aSetup}, \mathsf{aGen}, \mathsf{aEnc}, \mathsf{aDec})$

- $\mathsf{aSetup}(1^n)$ takes as input 1^n, and produces the public parameter $\mathtt{pp}$;
- $\mathsf{aGen}(\mathtt{pp})$ takes as input $\mathtt{pp}$, and produces ℓ anamorphic public/secret key pairs $(\mathtt{apk}_i, \mathtt{ask}_i)_{i\in[\ell]}$, and a double key $\mathtt{dkey}$;
- $\mathsf{aEnc}(\mathtt{dkey}, \mathtt{M}, \bar{\mathtt{m}})$ takes as input the double key $\mathtt{dkey}$, a normal plaintext set $\mathtt{M} = \{\mathtt{m}_i\}_{i\in[\ell]}$ and an anamorphic plaintext $\bar{\mathtt{m}}$, and produces an anamorphic ciphertext set $\mathtt{ACT} = \{\mathtt{act}_i\}_{i\in[\ell]}$;

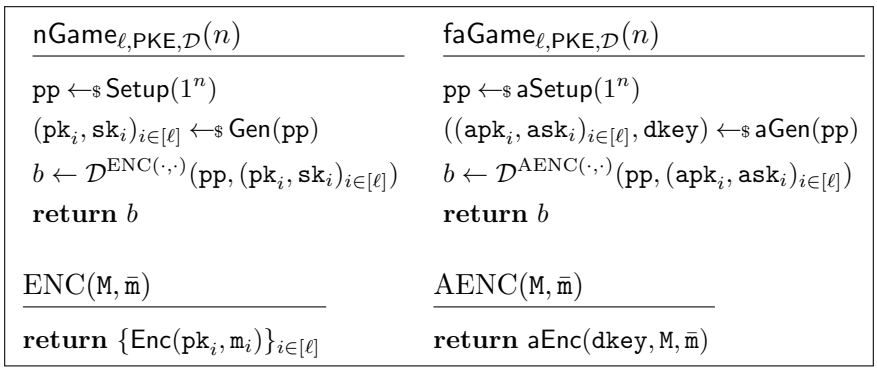

Figure 10: Definitions of game $\mathsf{nGame}_{\ell,\mathsf{PKE},\mathcal{D}}(n)$ and $\mathsf{faGame}_{\ell,\mathsf{PKE},\mathcal{D}}(n)$.

- $\mathsf{aDec}(\mathtt{dkey}, \mathtt{ACT})$ takes as input the double key $\mathtt{dkey}$ and the ciphertext set $\mathtt{ACT}$, and returns the anamorphic plaintext $\bar{\mathtt{m}}$,

and for any $\mathtt{M} \in \mathcal{M}^\ell$, any $\bar{\mathtt{m}} \in \overline{\mathcal{M}}$,

$$\Pr\left[\mathsf{aDec}(\mathtt{dkey}, \mathtt{ACT}) \neq \bar{\mathtt{m}} : \begin{array}{l} \mathtt{pp} \leftarrow\!\!\$\ \mathsf{aSetup}(1^n) \\ ((\mathtt{apk}_i, \mathtt{ask}_i)_{i\in[\ell]}, \mathtt{dkey}) \leftarrow\!\!\$\ \mathsf{aGen}(\mathtt{pp}) \\ \mathtt{ACT} \leftarrow\!\!\$\ \mathsf{aEnc}(\mathtt{dkey}, \mathtt{M}, \bar{\mathtt{m}}) \end{array}\right] \leq \mathsf{negl}(n)\,.$$

Definition 6.2 (Secure ℓ-Receiver-AME)**.** Let PKE be an ℓ-receiver-AME with extension $(\mathsf{aSetup}, \mathsf{aGen}, \mathsf{aEnc}, \mathsf{aDec})$. We say PKE is a secure ℓ-receiver-AME if following conditions hold,

- PKE is CPA secure;
- For any plaintext set $\widehat{M} \in \mathcal{M}^\ell$, $\mathsf{fAME}_{\widehat{M}} = (\mathsf{aSetup}, \mathsf{aGen}_{2\ell+1}, \mathsf{aEnc}_{1,\widehat{M}}, \mathsf{aDec})$ is a symmetric encryption scheme, where aGen_i denotes selecting the ith component of the triplet generated by aGen as the output, $\mathsf{aEnc}_{1,\widehat{M}}$ denotes running aEnc with the normal plaintext set $\mathtt{M} = \widehat{M}$;
- For any PPT adversary $\mathcal{D}$,

$$\left|\Pr\left[\mathsf{nGame}_{\ell,\mathsf{PKE},\mathcal{D}}(n) = 1\right] - \Pr\left[\mathsf{faGame}_{\ell,\mathsf{PKE},\mathcal{D}}(n) = 1\right]\right| \leq \mathsf{negl}(n)\,.$$

6.2 ℓ-Sender-AME $\Rightarrow$ ℓ-Receiver-AME

Theorem 6.1. *Let* PKE *be a strongly secure* ℓ*-sender-AME with extension* $(\mathsf{fRandom}, \mathsf{dDec})$ *and duplicate plaintext space* $\overline{\mathcal{M}}$*.* PKE *is also a secure* ℓ*-receiver-AME with extension in Fig. 11 and anamorphic plaintext space* $\overline{\mathcal{M}}$*.*

Proof. One can easily verify that the correctness of ℓ-sender-AME implies the correctness of ℓ-receiver-AME, and, for any plaintext set $\widehat{M} \in \mathcal{M}^\ell$, $\mathsf{fAME}_{\widehat{M}} = (\mathsf{aSetup}, \mathsf{aGen}_{2\ell+1}, \mathsf{aEnc}_{1,\widehat{M}}, \mathsf{aDec})$ is a symmetric encryption scheme.

Next, we prove that for any PPT adversary $\mathcal{D}$, the advantage of distinguishing games $\mathsf{nGame}_{\ell,\mathsf{PKE},\mathcal{D}}(n)$ and $\mathsf{faGame}_{\ell,\mathsf{PKE},\mathcal{D}}(n)$ in Fig. 10 is negligible.

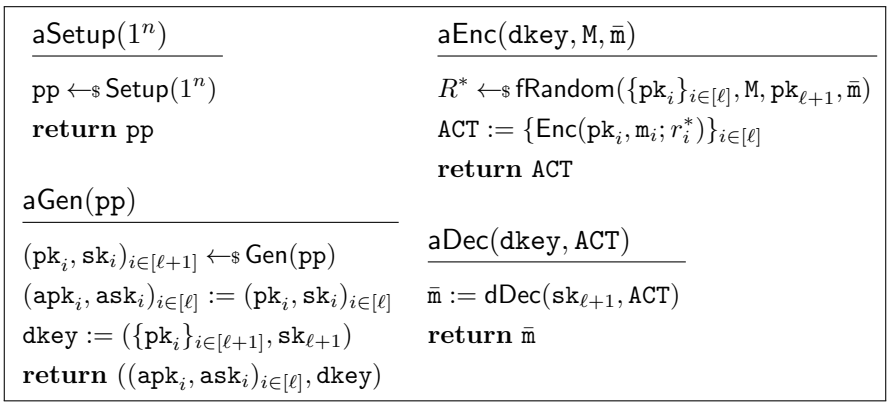

Figure 11: Receiver-anamorphic extension built on sender-anamorphic extension.

We show how to build an adversary $\mathcal{B}$ distinguishing with $\mathsf{Ideal}_{\ell,\mathsf{PKE},\mathcal{B}}(n)$ and $\mathsf{Real}^{\mathsf{fRandom}}_{\ell,\mathsf{PKE},\mathcal{B}}(n)$. In specific, $\mathcal{B}$ receives $(\mathtt{pp}, \mathtt{FPK}, \mathtt{FSK}, \mathtt{dpk})$ and can makes encryption queries with $(\mathtt{FM}, \mathtt{dm})$. The encryption oracle encrypts FM with $R^* \leftarrow\!\!\$\ \mathsf{fRandom}(\mathtt{FPK}, \mathtt{FM}, \mathtt{dpk}, \mathtt{dm})$ in $\mathsf{Real}^{\mathsf{fRandom}}_{\ell,\mathsf{PKE},\mathcal{B}}(n)$ and $(r_i)_{i\in[\ell]} \leftarrow\!\!\$\ \mathcal{R}$ in $\mathsf{Ideal}_{\ell,\mathsf{PKE},\mathcal{B}}(n)$.

$\mathcal{B}$ forwards $(\mathtt{pp}, \mathtt{FPK}, \mathtt{FSK})$ to $\mathcal{D}$. To answer the encryption query $(\mathtt{M}, \bar{\mathtt{m}})$ from $\mathcal{D}$, $\mathcal{B}$ queries its encryption oracle with $(\mathtt{M}, \bar{\mathtt{m}})$ and returns the ciphertexts to $\mathcal{D}$. The simulation is perfect. If the encryption oracle encrypts M with $(r_i)_{i\in[\ell]} \leftarrow\!\!\$\ \mathcal{R}$, $\mathcal{B}$ simulates $\mathsf{nGame}_{\ell,\mathsf{PKE},\mathcal{D}}(n)$ for $\mathcal{D}$. Otherwise, the encryption oracle encrypts M with $R^* \leftarrow\!\!\$\ \mathsf{fRandom}(\mathtt{FPK}, \mathtt{M}, \mathtt{dpk}, \bar{\mathtt{m}})$, $\mathcal{B}$ simulates $\mathsf{faGame}_{\ell,\mathsf{PKE},\mathcal{D}}(n)$ for $\mathcal{D}$. □

Obviously, not every ℓ-receiver-AME is also ℓ-sender-anamorphic. In particular, the construction of ℓ-receiver-AME might rely on the anamorphic setup and key generation algorithms, or the double key for anamorphic encryption and decryption algorithms might cannot be parsed as a public/secret key pair.

7 Relation between ℓ-Sender-AME and Public-Key Stegosystem

In this section, we recall the definition of public-key stegosystem and its security in [32], and rigorously prove that a strongly secure ℓ-sender-AME implies a secure public-key stegosystem.

7.1 Public-Key Stegosystem

Definition 7.1 (Public-Key Stegosystem [32])**.** A public-key stegosystem PKS consists of four algorithms $(\mathsf{sSetup}, \mathsf{sGen}, \mathsf{sEnc}, \mathsf{sDec})$

- $\mathsf{sSetup}(1^n)$ takes as input 1^n, and returns the public parameter pp;
- $\mathsf{sGen}(\mathtt{pp})$ takes as input pp, and returns a public/secret key pair $(\mathtt{pk}, \mathtt{sk})$;
- $\mathsf{sEnc}(\mathtt{pk}, \mathtt{m}, \mathtt{h})$ takes pk, a string $\mathtt{m} \in \{0,1\}^*$ (i.e., the hiddentext) and a history h, and returns a list of documents $\mathtt{ST} = (\mathtt{st}_1, \mathtt{st}_2, \cdots, \mathtt{st}_\ell)$ (i.e., the stegotext). Also, it can access a oracle $M(\mathtt{h})$ that samples a document according to a channel distribution $\mathcal{C}_\mathtt{h}$;
- $\mathsf{sDec}(\mathtt{sk}, \mathtt{ST}, \mathtt{h})$ takes sk, $\mathtt{ST} = (\mathtt{st}_1, \mathtt{st}_2, \cdots, \mathtt{st}_\ell)$ and a history h, and returns the string m.

and for any polynomial $\ell(n)$, any $\mathtt{m} \in \{0,1\}^{\ell(n)}$,

$$\Pr\left[\mathsf{sDec}(\mathtt{sk},\mathtt{ST},\mathtt{h}) \neq \mathtt{m} : \begin{array}{l} \mathtt{pp} \leftarrow\!\!\$\, \mathsf{sSetup}(1^n) \\ (\mathtt{pk},\mathtt{sk}) \leftarrow\!\!\$\, \mathsf{sGen}(\mathtt{pp}) \\ \mathtt{ST} \leftarrow\!\!\$\, \mathsf{sEnc}(\mathtt{pk},\mathtt{m},\mathtt{h}) \end{array}\right] \leq \mathsf{negl}(n)\,.$$

Definition 7.2 (CHA Security [32])**.** Let $\mathsf{PKS} = (\mathsf{sSetup},\mathsf{sGen},\mathsf{sEnc},\mathsf{sDec})$ be a public-key stegosystem. We say PKS is *secure against chosen hiddentext attacks (CHA secure)* over channel $\mathcal{C}$ if for any PPT warden $\mathcal{W}_1, \mathcal{W}_2$,

$$\left|\Pr\left[b = b' : \begin{array}{l} \mathtt{pp} \leftarrow\!\!\$\, \mathsf{sSetup}(1^n) \\ (\mathtt{pk},\mathtt{sk}) \leftarrow\!\!\$\, \mathsf{sGen}(\mathtt{pp}) \\ (\mathtt{m}^*,\mathtt{h}^*,\mathtt{st}) \leftarrow \mathcal{W}_1^{M(\mathtt{h})}(\mathtt{pp},\mathtt{pk}) \\ \mathtt{ST}_0 \leftarrow\!\!\$\, \mathsf{sEnc}(\mathtt{pk},\mathtt{m}^*,\mathtt{h}^*) \\ \mathtt{ST}_1 \leftarrow\!\!\$\, \mathcal{C}_{\mathtt{h}^*}^{|\mathtt{ST}_0|} \\ b \leftarrow\!\!\$\, \{0,1\} \\ b' \leftarrow \mathcal{W}_2^{M(\mathtt{h})}(\mathtt{st},\mathtt{ST}_b) \end{array}\right] - \frac{1}{2}\right| \leq \mathsf{negl}(n)\,.$$

7.2 ℓ-Sender-AME $\Rightarrow$ Public-Key Stegosystem

Let $\mathsf{PKE} = (\mathsf{Setup},\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ be a PKE scheme. We define a channel $\mathcal{C}_{\mathsf{PKE},n}(\ell)$ for PKE with security parameter $n \in \mathbb{N}^+$ and $\ell = \mathsf{poly}(n)$, and specify the distributions for any history $\mathtt{h}$ as below.

- Case 1: $(\mathtt{h} = \emptyset)$. $\mathcal{C}_{\mathsf{PKE},n,\mathtt{h}}(\ell)$ is the distribution of all the public parameters generated by $\mathsf{Setup}(1^n)$;
- Case 2: $(\mathtt{h} = \mathtt{pp}\|(\mathtt{pk}_1,\mathtt{sk}_1)\|\cdots\|(\mathtt{pk}_r,\mathtt{sk}_r)$ with $r \in [0,\ell-1])$. $\mathcal{C}_{\mathsf{PKE},n,\mathtt{h}}(\ell)$ is the distribution of all the key pairs generated by $\mathsf{Gen}(\mathtt{pp})$;
- Case 3: $(\mathtt{h} = \mathtt{pp}\|(\mathtt{pk}_1,\mathtt{sk}_1)\|\cdots\|(\mathtt{pk}_\ell,\mathtt{sk}_\ell)\|\mathtt{m}_1\|\cdots\|\mathtt{m}_r$ with $\mathtt{m}_i \in \mathcal{M}$, $r \in [0,\ell-1])$. $\mathcal{C}_{\mathsf{PKE},n,\mathtt{h}}(\ell)$ is the uniform distribution over message space $\mathcal{M}$;
- Case 4: $(\mathtt{h} = \mathtt{pp}\|(\mathtt{pk}_1,\mathtt{sk}_1)\|\cdots\|(\mathtt{pk}_\ell,\mathtt{sk}_\ell)\|\mathtt{m}_1\|\cdots\|\mathtt{m}_\ell\|\mathtt{ct}_1\|\cdots\|\mathtt{ct}_r$ with $\mathtt{m}_i \in \mathcal{M}$, $\mathtt{ct}_i \leftarrow\!\!\$\, \mathsf{Enc}(\mathtt{pk}_{i'},\mathtt{m}_{i'})$, $i' = ((i-1) \bmod \ell)+1)$. $\mathcal{C}_{\mathsf{PKE},n,\mathtt{h}}(\ell)$ is the distribution of $\mathsf{Enc}(\mathtt{pk}_{r'},\mathtt{m}_{r'})$ where $r' = (r \bmod \ell)+1$.

Theorem 7.1. *Let* $\mathsf{PKE} = (\mathsf{Setup},\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ *be a strongly secure* ℓ*-sender-AME. The public-key stegosystem* $\mathsf{PKS} = (\mathsf{sSetup},\mathsf{sGen},\mathsf{sEnc},\mathsf{sDec})$ *in Fig. 12 over channel* $\mathcal{C}_{\mathsf{PKE},n}(\ell)$ *is CHA secure.*

Proof. We demonstrate how adversary $\mathcal{A}$ distinguishes game $\mathsf{Ideal}_{\ell,\mathsf{PKE},\mathcal{A}}(n)$ and $\mathsf{Real}^{\mathsf{fRandom}}_{\ell,\mathsf{PKE},\mathcal{A}}(n)$ by leveraging the warden $\mathcal{W}_1$ and $\mathcal{W}_2$ in CHA security game. $\mathcal{A}$ simulates the game for $\mathcal{W}_1$ and $\mathcal{W}_2$ as follows.

$\mathcal{A}$ receives $\mathtt{pp}$, ℓ forced key pairs $\{(\mathtt{fpk}_i,\mathtt{fsk}_i)\}_{i\in[\ell]}$ and $\mathtt{dpk}$, and can make encryption queries on $(\mathtt{FM},\mathtt{dm})$. $\mathcal{A}$ forwards $(\mathtt{pp},\mathtt{pk})$ to $\mathcal{W}_1$ and simulates the oracle $M(\mathtt{h})$ as follows.

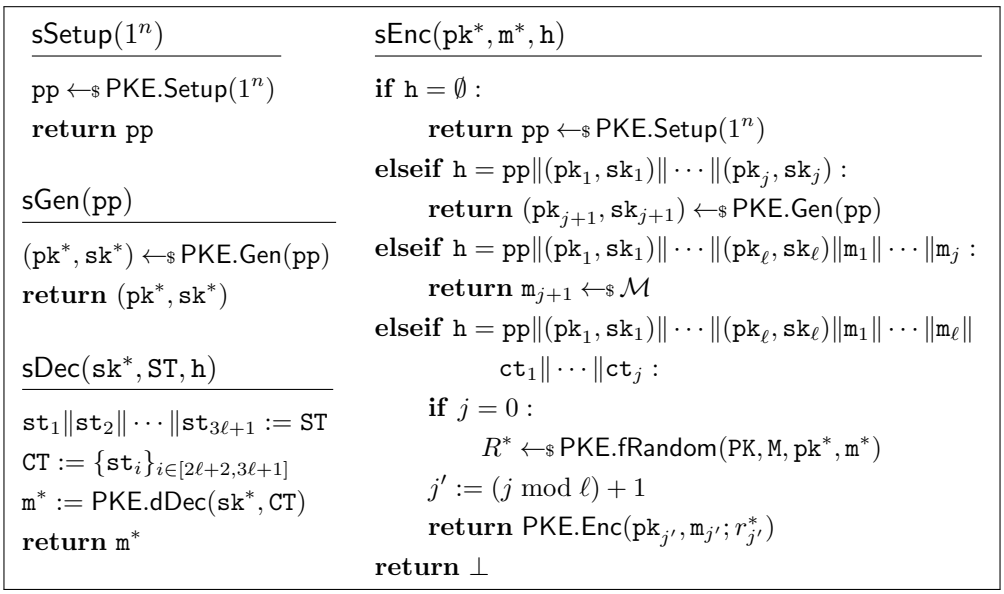

Figure 12: Generic public-key stegosystem built on ℓ-sender-AME.

- If $\mathtt{h} = \emptyset$, $\mathcal{A}$ returns $\mathtt{pp}$ to $\mathcal{W}_1$;
- If $\mathtt{h} = \mathtt{pp}\|(\mathtt{pk}_1, \mathtt{sk}_1)\|\cdots\|(\mathtt{pk}_r, \mathtt{sk}_r)$ with $r \in [0, \ell-1]$, $\mathcal{A}$ returns the $(r+1)$-th forced key pair $(\mathtt{fpk}_{r+1}, \mathtt{fsk}_{r+1})$ to $\mathcal{W}_1$;
- If $\mathtt{h} = \mathtt{pp}\|(\mathtt{pk}_1, \mathtt{sk}_1)\|\cdots\|(\mathtt{pk}_\ell, \mathtt{sk}_\ell)\|\mathtt{m}_1\|\cdots\|\mathtt{m}_r$ with $r \in [0, \ell-1]$, $\mathcal{A}$ samples $\mathtt{m}_{r+1}$ uniformly from $\mathcal{M}$, and outputs $\mathtt{m}_{r+1}$;
- If $\mathtt{h} = \mathtt{pp}\|(\mathtt{pk}_1, \mathtt{sk}_1)\|\cdots\|(\mathtt{pk}_\ell, \mathtt{sk}_\ell)\|\mathtt{m}_1\|\cdots\|\mathtt{m}_\ell\|\mathtt{ct}_1\|\cdots\|\mathtt{ct}_r$ with $r \geq 0$, $\mathcal{A}$ computes $\mathtt{ct}_{r+1} \leftarrow\!\!\$\ \mathsf{PKE.Enc}(\mathtt{pk}_{r'}, \mathtt{m}_{r'})$, where $r' = (r \bmod \ell) + 1$, and outputs $\mathtt{ct}_{r+1}$.

One can note that $\mathcal{A}$ simulates the channel $\mathcal{C}_{\mathsf{PKE},n}(\ell)$ perfectly. After receiving $(\mathtt{m}^*, \mathtt{h}^*)$ from $\mathcal{W}_1$, $\mathcal{A}$ simulates the challenge stegotext $\mathtt{ST}_b$ as follows.

- If $\mathtt{h}^* = \emptyset$, $\mathcal{A}$ outputs $\mathtt{pp}$;
- If $\mathtt{h}^* = \mathtt{pp}\|(\mathtt{pk}_1, \mathtt{sk}_1)\|\cdots\|(\mathtt{pk}_r, \mathtt{sk}_r)$ with $r \in [0, \ell-1]$, $\mathcal{A}$ outputs the $(r+1)$-th forced key pair $(\mathtt{fpk}_{r+1}, \mathtt{fsk}_{r+1})$;
- If $\mathtt{h}^* = \mathtt{pp}\|(\mathtt{pk}_1, \mathtt{sk}_1)\|\cdots\|(\mathtt{pk}_\ell, \mathtt{sk}_\ell)\|\mathtt{m}_1\|\cdots\|\mathtt{m}_r$ with $r \in [0, \ell-1]$, $\mathcal{A}$ samples $\mathtt{m}_{r+1}$ uniformly from $\mathcal{M}$, and outputs $\mathtt{m}_{r+1}$;
- If $\mathtt{h}^* = \mathtt{pp}\|(\mathtt{pk}_1, \mathtt{sk}_1)\|\cdots\|(\mathtt{pk}_\ell, \mathtt{sk}_\ell)\|\mathtt{m}_1\|\cdots\|\mathtt{m}_\ell\|\mathtt{ct}_1\|\cdots\|\mathtt{ct}_r$ with $r \geq 0$: If $(r \bmod \ell) = 0$, $\mathcal{A}$ queries the encryption oracle on $\mathtt{M} := \{\mathtt{m}_i\}_{i\in[\ell]}$ and $\mathtt{m}^*$, saves the ciphertext set $\mathtt{CT}$ and outputs the first ciphertext $\mathtt{ct}_1$. Otherwise, $\mathcal{A}$ only needs to output the r'-th ciphertext $\mathtt{ct}_{r'}$ where $r' = (r \bmod \ell) + 1$.

If $\mathcal{A}$ is in game $\mathsf{Ideal}_{\ell,\mathsf{PKE},\mathcal{A}}(n)$, then $\mathtt{ST}_b$ is sampled from $\mathcal{C}_{\mathsf{PKE},n,\mathtt{h}^*}(\ell)$. If $\mathcal{A}$ is in game $\mathsf{Real}^{\mathsf{fRandom}}_{\ell,\mathsf{PKE},\mathcal{A}}(n)$, then $\mathtt{ST}_b$ is the output of $\mathsf{sEnc}(\mathtt{pk}, \mathtt{m}^*, \mathtt{h}^*)$. □

We remark that not every public-key stegosystem implies an ℓ-sender-AME. In particular, the chosen-stegotext secure construction in [32] also uses the secret key of sender (i.e., Alice) to generate the stegotexts, while the input of algorithm $\mathsf{fRandom}$ in ℓ-sender-AME only involves public keys and plaintexts.

8 Relation between AME and Generalized ASA on PKE

8.1 ASA Model for PKE

In this section, we extend the generalized ASA model in [8] for PKE. Unlike symmetric encryption, PKE uses *public key* to encrypt messages, and it is impossible to extract the corresponding secret key from the ciphertext. Hence, one natural goal of ASA against PKE is recovering underlying plaintext or other subliminal messages from the ciphertext.

Definition 8.1 (Generalized ASA on PKE)**.** Let $\mathsf{PKE} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE. For $\mathtt{pp} \leftarrow\!\!\$\, \mathsf{PKE.Setup}(1^n)$ and $(\mathtt{pk}, \mathtt{sk}) \leftarrow\!\!\$\, \mathsf{PKE.Gen}(\mathtt{pp})$, an ASA on PKE is $\mathsf{ASA} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Ext})$.

- $\mathsf{ASA.Gen}(\mathtt{pp})$ returns a subversion key $\mathtt{skey}$.
- $\mathsf{ASA.Enc}(\mathtt{skey}, \mathtt{sm}, \mathtt{pk}, \mathtt{m}, \tau)$ takes as input $\mathtt{skey}$, a subliminal message $\mathtt{sm} \in \overline{\mathcal{M}}$, a public key $\mathtt{pk}$, an encryption message $\mathtt{m} \in \mathcal{M}$ and a (possible) state τ, returns a ciphertext $\mathtt{ct}$ and updates the state τ.
- $\mathsf{ASA.Ext}(\mathtt{skey}, \{\mathtt{ct}_i\}_{i\in[\ell]})$ takes as input $\mathtt{skey}$ and a ciphertext set $\{\mathtt{ct}_i\}_{i\in[\ell]}$ with $\ell = \mathsf{poly}(n)$, and returns the subliminal message $\mathtt{sm}$.

We remark that when message $\mathtt{sm}$ is set as plaintext $\mathtt{m}$, the goal of ASA is recovering the plaintext from ciphertext. In some cases, it is unlikely to encode all the bits of message $\mathtt{sm}$ into one ciphertext $\mathtt{ct}$. Instead, adversary takes advantage of the fact that the user would invoke algorithm $\mathsf{ASA.Enc}$ multiple times to embed the message $\mathtt{sm}$ into a bunch of ciphertexts $\{\mathtt{ct}_i\}_{i\in[\ell]}$, and the subverted encryption algorithm can be denoted as below.

$\underline{\mathsf{ASA.Enc}^{\ell}(\mathtt{skey}, \mathtt{sm}, \mathtt{PK} := \{\mathtt{pk}_i\}_{i\in[\ell]}, \mathtt{M} := \{\mathtt{m}_i\}_{i\in[\ell]})}$

$1:\ \tau := \varepsilon$

$2:\ \textbf{for}\ i \in [\ell]\ \textbf{do}:$

$3:\qquad \mathtt{ct}_i \leftarrow\!\!\$\, \mathsf{ASA.Enc}(\mathtt{skey}, \mathtt{sm}, \mathtt{pk}_i, \mathtt{m}_i, \tau)$

$4:\ \textbf{return}\ \mathtt{CT} := \{\mathtt{ct}_i\}_{i\in[\ell]}$

We say ASA is *asymmetric* if 1) subversion key $\mathtt{skey}$ can be parsed as a subversion key pair $(\mathtt{psk}, \mathtt{ssk})$, 2) $\mathsf{ASA.Enc}$ takes public subversion key $\mathtt{psk}$ as input instead of $\mathtt{skey}$ and 3) $\mathsf{ASA.Ext}$ takes secret subversion key $\mathtt{ssk}$ as input instead of $\mathtt{skey}$. The generalized ASA in Def. 8.1 can be regarded as *symmetric*. Clearly, asymmetric ASA is a special case of symmetric ASA. The definitions of properties for these two types of ASAs are slightly different.

To capture the recovery of subliminal message, we present the definition of recoverability as follows. In particular, for asymmetric ASA, only public subversion key $\mathtt{psk}$ is hardwired into the subverted encryption algorithm $\mathsf{ASA.Enc}$, and extracting subliminal message requires secret subversion key $\mathtt{ssk}$.

Definition 8.2 (Recoverability)**.** Let $\mathsf{ASA} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Ext})$ be an ASA on $\mathsf{PKE} = (\mathsf{Setup}, \mathsf{Gen},$

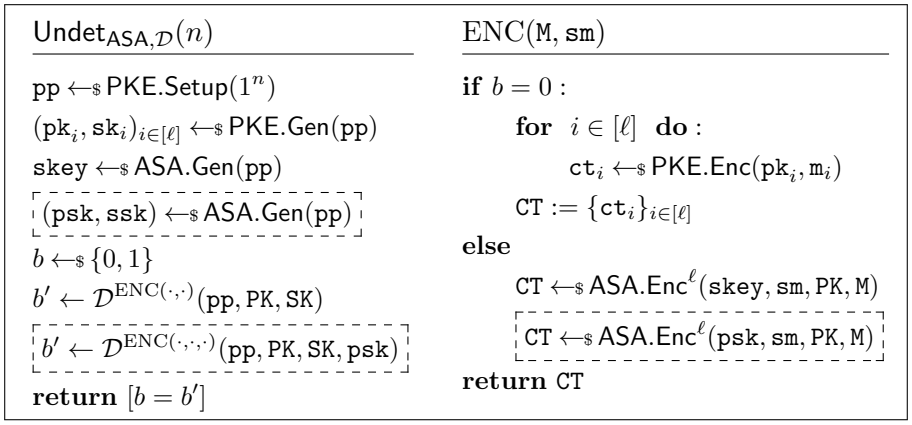

Figure 13: Definition of game $\mathsf{Undet}_{\mathsf{ASA},\mathcal{D}}(n)$.

$\mathsf{Enc}, \mathsf{Dec}$). We say ASA satisfies recoverability if for any $\mathtt{M} \in \mathcal{M}^{\ell}$ and any $\mathtt{sm} \in \overline{\mathcal{M}}$,

$$\Pr\left[\begin{array}{l} \mathsf{ASA.Ext}(\mathtt{skey}, \mathtt{CT}) \neq \mathtt{sm} \\ \boxed{\mathsf{ASA.Ext}(\mathtt{ssk}, \mathtt{CT}) \neq \mathtt{sm}} \end{array} : \begin{array}{l} \mathtt{pp} \leftarrow\!\!\$\ \mathsf{PKE.Setup}(1^n) \\ \mathtt{skey} \leftarrow\!\!\$\ \mathsf{ASA.Gen}(\mathtt{pp}) \\ \boxed{(\mathtt{psk}, \mathtt{ssk}) \leftarrow\!\!\$\ \mathsf{ASA.Gen}(\mathtt{pp})} \\ (\mathtt{pk}_i, \mathtt{sk}_i)_{i\in[\ell]} \leftarrow\!\!\$\ \mathsf{PKE.Gen}(\mathtt{pp}) \\ \mathtt{CT} \leftarrow\!\!\$\ \mathsf{ASA.Enc}^{\ell}(\mathtt{skey}, \mathtt{sm}, \mathtt{PK}, \mathtt{M}) \\ \boxed{\mathtt{CT} \leftarrow\!\!\$\ \mathsf{ASA.Enc}^{\ell}(\mathtt{psk}, \mathtt{sm}, \mathtt{PK}, \mathtt{M})} \end{array}\right] \leq \mathsf{negl}(n)\,.$$

The differences of recoverability for asymmetric ASA are marked with dashed boxes.

The stealthy feature of ASA on PKE requires that the ordinary users cannot distinguish the honest or subverted implementation of encryption algorithm with overwhelming advantage.

Definition 8.3 (Undetectability)**.** Let $\mathsf{ASA} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Ext})$ be an ASA on $\mathsf{PKE} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. We say ASA satisfies *secret undetectability* if for any PPT detector $\mathcal{D}$, the advantage of $\mathcal{D}$ in game $\mathsf{Undet}_{\mathsf{ASA},\mathcal{D}}(n)$ is negligible.

If detector $\mathcal{D}$ in game $\mathsf{Undet}_{\mathsf{ASA},\mathcal{D}}(n)$ is not provided with $\mathtt{SK} = \{\mathtt{sk}_i\}_{i\in[\ell]}$, we say ASA satisfies *public undetectability*.

8.2 Symmetric ASA on PKE $\Rightarrow$ ℓ-Receiver-AME

Theorem 8.1. *Let* $\mathsf{ASA} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Ext})$ *be an ASA on CPA-secure* $\mathsf{PKE} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with subliminal message space* $\overline{\mathcal{M}}$*. Then,* PKE *is a secure* ℓ*-receiver-AME with extension in Fig. 14 and anamorphic plaintext space* $\overline{\mathcal{M}}$*.*

Proof. By the recoverability of ASA, PKE with extension in Fig. 14 is an ℓ-receiver-AME with anamorphic plaintext space $\overline{\mathcal{M}}$.

One can note that for every $\widehat{M} \in \mathcal{M}^{\ell}$, $\mathsf{fAME}_{\widehat{M}}$ is a symmetric encryption scheme.

Next we prove that if ASA satisfies indistinguishability, then PKE with algorithms in Fig. 14 satisfies the third condition of secure ℓ-receiver-AME. In particular, we show how to break the indistinguishability of ASA using the adversary $\mathcal{D}$ in game $\mathsf{nGame}_{\mathsf{PKE},\mathcal{D}}(n)$ or $\mathsf{faGame}_{\mathsf{PKE},\mathcal{D}}(n)$.

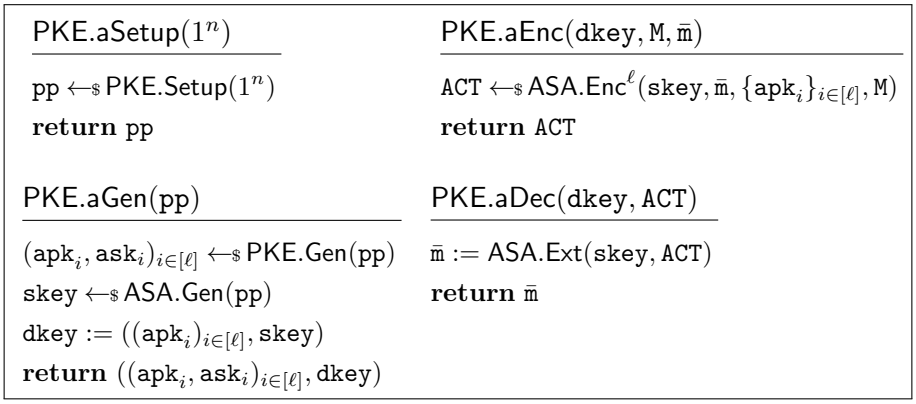

Figure 14: Generic ℓ-receiver-AME built on ASA against PKE.

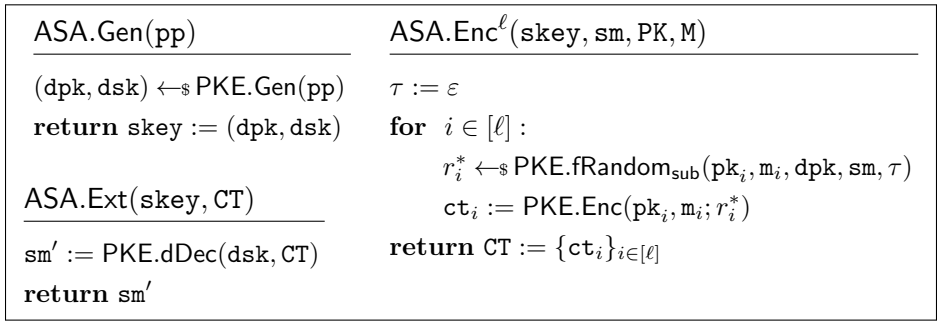

Figure 15: Generalized ASA on PKE from ℓ-sender-AME.

Simulator $\mathcal{B}$ plays the role of detector in game $\mathsf{Undet}_{\mathsf{ASA},\mathcal{B}}(n)$ and simulates the game $\mathsf{nGame}_{\mathsf{PKE},\mathcal{D}}(n)$ or $\mathsf{faGame}_{\mathsf{PKE},\mathcal{D}}(n)$ for $\mathcal{D}$ as follows. $\mathcal{B}$ receives $(\mathtt{pp}, \mathtt{PK}, \mathtt{SK})$ and forwards them to $\mathcal{D}$. To answer the encryption query $(\mathtt{M}, \bar{\mathtt{m}})$ from $\mathcal{D}$, $\mathcal{B}$ queries its own encryption oracle on $(\mathtt{M}, \bar{\mathtt{m}})$ and returns the results to $\mathcal{D}$.

The simulation above is perfect. If $b = 0$ in game $\mathsf{Undet}_{\mathsf{ASA},\mathcal{B}}(n)$, $\mathcal{B}$ simulates $\mathsf{nGame}_{\mathsf{PKE},\mathcal{D}}(n)$ for $\mathcal{D}$; Otherwise, it simulates $\mathsf{faGame}_{\mathsf{PKE},\mathcal{D}}(n)$. □

We remark that not every ℓ-receiver-AME implies an ASA on PKE. In particular, the subversion key in ASA is independent of public and secret keys, while the double key in ℓ-receiver-AME might include the secret key. Thus, it might be impossible to build algorithm Gen for ASA with algorithm aGen in ℓ-receiver-AME. See Appendix C.1 for concrete example.

8.3 ℓ-Sender-AME ⇒ Asymmetric ASA on PKE

Note that the algorithm $\mathsf{ASA.Enc}^{\ell}$ runs the subverted encryption ASA.Enc for ℓ times sequentially. To achieve this, it is required that the algorithm PKE.fRandom could be rewrote as the iteration of running (possibly stateful) sub-algorithm $\mathsf{PKE.fRandom}_{\mathsf{sub}}$ that take as input a forced public key, a forced plaintext, a duplicate public key, a duplicate plaintext and a state (if exists), and outputs a randomness. In this case, the subverted encryption ASA.Enc runs $\mathsf{PKE.fRandom}_{\mathsf{sub}}$ to generate randomness r_i^* and PKE.Enc to encrypt the forced plaintext $\mathtt{m}_i$ with forced public key $\mathtt{pk}_i$ and randomness r_i^*.

Theorem 8.2. *Let* PKE *be a strongly secure ℓ-sender-AME associated with algorithm* fRandom

and dDec. *Then, there exists an asymmetric ASA on* PKE*, as shown in Fig. 15, satisfying both recoverability and secret undetectability.*

Proof. By the definition of ℓ-sender-AME, for any $\mathtt{FM} \in \mathcal{M}^\ell$, any $\mathtt{dm} \in \overline{\mathcal{M}}$,

$$\Pr\left[\begin{array}{l|l} \mathsf{PKE.dDec}(\mathtt{dsk},\mathtt{CT}) \neq \mathtt{dm} & \begin{array}{l} \mathtt{pp} \leftarrow\!\!\$\ \mathsf{PKE.Setup}(1^n) \\ (\mathtt{fpk}_i,\mathtt{fsk}_i)_{i\in[\ell]} \leftarrow\!\!\$\ \mathsf{PKE.Gen}(\mathtt{pp}) \\ (\mathtt{dpk},\mathtt{dsk}) \leftarrow\!\!\$\ \mathsf{PKE.Gen}(\mathtt{pp}) \\ R^* \leftarrow\!\!\$\ \mathsf{PKE.fRandom}(\mathtt{FPK},\mathtt{FM},\mathtt{dpk},\mathtt{dm}) \\ \mathtt{CT} := \{\mathsf{PKE.Enc}(\mathtt{fpk}_i,\mathtt{fm}_i;r_i^*)\}_{i\in[\ell]} \end{array} \end{array}\right] \leq \mathsf{negl}(n)$$

According to the description of ASA in Fig. 15, one can note that the following inequality also holds for any $\mathtt{M} \in \mathcal{M}^\ell$ and any $\mathtt{sm} \in \overline{\mathcal{M}}$.

$$\Pr\left[\mathsf{ASA.Ext}(\mathtt{skey},\mathtt{CT}) \neq \mathtt{sm} : \begin{array}{l} \mathtt{pp} \leftarrow\!\!\$\ \mathsf{PKE.Setup}(1^n) \\ (\mathtt{pk}_i,\mathtt{sk}_i)_{i\in[\ell]} \leftarrow\!\!\$\ \mathsf{PKE.Gen}(\mathtt{pp}) \\ \mathtt{skey} \leftarrow\!\!\$\ \mathsf{ASA.Gen}(\mathtt{pp}) \\ \mathtt{CT} \leftarrow\!\!\$\ \mathsf{ASA.Enc}^\ell(\mathtt{skey},\mathtt{sm},\mathtt{PK},\mathtt{M}) \end{array}\right] \leq \mathsf{negl}(n)\,.$$

To prove the secret undetectability of ASA, we demonstrate how adversary $\mathcal{A}$ distinguishes game $\mathsf{Ideal}_{\ell,\mathsf{PKE},\mathcal{A}}(n)$ and $\mathsf{Real}^{\mathsf{fRandom}}_{\ell,\mathsf{PKE},\mathcal{A}}(n)$ in Fig. 2 by leveraging the detector $\mathcal{D}$ in $\mathsf{Undet}_{\mathsf{ASA},\mathcal{D}}(n)$. Adversary $\mathcal{A}$ plays the role of challenger for $\mathcal{D}$ as follows.

$\mathcal{A}$ receives $\mathtt{pp}$, ℓ forced public/secret key pairs $\{(\mathtt{fpk}_i,\mathtt{fsk}_i)\}_{i\in[\ell]}$ and duplicate public key $\mathtt{dpk}$, and can make encryption query for normal plaintext set $\mathtt{M} = \{\mathtt{m}_i\}_{i\in[\ell]}$ and anamorphic plaintext $\bar{\mathtt{m}}$. $\mathcal{A}$ forwards $(\mathtt{pp},\mathtt{FPK} := \{\mathtt{fpk}_i\}_{i\in[\ell]},\mathtt{FSK} := \{\mathtt{fsk}_i\}_{i\in[\ell]},\mathtt{dpk})$ to $\mathcal{D}$ and answers the encryption query from $\mathcal{D}$ with the ciphertexts generated by the oracle. If $\mathcal{A}$ is in game $\mathsf{Ideal}_{\ell,\mathsf{PKE},\mathcal{A}}(n)$, then $\mathcal{A}$ simulates the game $\mathsf{Undet}_{\mathsf{ASA},\mathcal{D}}(n)$ with $b=0$. Otherwise, $\mathcal{A}$ is in game $\mathsf{Real}^{\mathsf{fRandom}}_{\ell,\mathsf{PKE},\mathcal{A}}(n)$ and simulates the game $\mathsf{Undet}_{\mathsf{ASA},\mathcal{D}}(n)$ with $b=1$. □

Finally, we remark that not every asymmetric ASA on PKE directly implies that the underlying PKE is an ℓ-sender-AME. In particular, when the generation of subversion key pair in asymmetric ASA is different from the key generation of PKE, we might be unable to construct coin-toss faking algorithm fRandom using the idea of this ASA. See Appendix C.2 for example.

Acknowledgements. We would like to thank all anonymous reviewers for their valuable comments. This work is supported in part by the National Natural Science Foundation of China (Grant No.62122092, No.62202485, No.62032005).

References

[1] Abdalla, M., Bellare, M., Neven, G.: Robust encryption. In: Theory of Cryptography: 7th Theory of Cryptography Conference, TCC 2010. pp. 480–497. Springer (2010)

[2] Ateniese, G., Magri, B., Venturi, D.: Subversion-resilient signature schemes. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. pp. 364–375 (2015)

[3] Backes, M., Cachin, C.: Public-key steganography with active attacks. In: Theory of Cryptography: Second Theory of Cryptography Conference, TCC 2005. pp. 210–226. Springer (2005)

[4] Banfi, F., Gegier, K., Hirt, M., Maurer, U.: Anamorphic encryption, revisited. Cryptology ePrint Archive, Paper 2023/249 (2023)

[5] Bellare, M., Jaeger, J., Kane, D.: Mass-surveillance without the state: Strongly undetectable algorithm-substitution attacks. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. pp. 1431–1440 (2015)

[6] Bellare, M., Paterson, K.G., Rogaway, P.: Security of symmetric encryption against mass surveillance. In: Advances in Cryptology–CRYPTO 2014: 34th Annual Cryptology Conference, Proceedings, Part I 34. pp. 1–19. Springer (2014)

[7] Bellare, M., Rogaway, P.: Optimal asymmetric encryption. In: Advances in Cryptology—EUROCRYPT'94: Workshop on the Theory and Application of Cryptographic Techniques, Proceedings 13. pp. 92–111. Springer (1995)

[8] Berndt, S., Liśkiewicz, M.: Algorithm substitution attacks from a steganographic perspective. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. pp. 1649–1660 (2017)

[9] Bernstein, D.J., Hamburg, M., Krasnova, A., Lange, T.: Elligator: elliptic-curve points indistinguishable from uniform random strings. In: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. pp. 967–980 (2013)

[10] Chen, R., Huang, X., Yung, M.: Subvert kem to break dem: practical algorithm-substitution attacks on public-key encryption. In: Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Proceedings, Part II 26. pp. 98–128. Springer (2020)

[11] Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Advances in Cryptology—EUROCRYPT 2002: International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings 21. pp. 45–64. Springer (2002)

[12] Degabriele, J.P., Farshim, P., Poettering, B.: A more cautious approach to security against mass surveillance. In: Fast Software Encryption: 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers 22. pp. 579–598. Springer (2015)

[13] ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE transactions on information theory **31**(4), 469–472 (1985)

[14] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the fortieth annual ACM symposium on Theory of computing. pp. 197–206 (2008)

[15] Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of computer and system sciences **28**(2), 270–299 (1984)

[16] Golle, P., Jakobsson, M., Juels, A., Syverson, P.: Universal re-encryption for mixnets. In: Topics in Cryptology–CT-RSA 2004: The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings. pp. 163–178. Springer (2004)

[17] Hofheinz, D., Kiltz, E.: Secure hybrid encryption from weakened key encapsulation. In: Advances in Cryptology-CRYPTO 2007: 27th Annual International Cryptology Conference, Proceedings 27. pp. 553–571. Springer (2007)

[18] Hopper, N.J., Langford, J., Von Ahn, L.: Provably secure steganography. In: Advances in Cryptology—CRYPTO 2002: 22nd Annual International Cryptology Conference, Proceedings 22. pp. 77–92. Springer (2002)

[19] Horel, T., Park, S., Richelson, S., Vaikuntanathan, V.: How to subvert backdoored encryption: Security against adversaries that decrypt all ciphertexts. In: 10th Innovations in Theoretical Computer Science Conference (ITCS 2019). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2018)

[20] Kurosawa, K., Desmedt, Y.: A new paradigm of hybrid encryption scheme. In: Advances in Cryptology–CRYPTO 2004: 24th Annual International Cryptology Conference, Proceedings 24. pp. 426–442. Springer (2004)

[21] Kutylowski, M., Persiano, G., Phan, D.H., Yung, M., Zawada, M.: Anamorphic signatures: Secrecy from a dictator who only permits authentication! Cryptology ePrint Archive, Paper 2023/356 (2023)

[22] Kutylowski, M., Persiano, G., Phan, D.H., Yung, M., Zawada, M.: The self-anti-censorship nature of encryption: On the prevalence of anamorphic cryptography. Cryptology ePrint Archive, Paper 2023/434 (2023)

[23] Möller, B.: A public-key encryption scheme with pseudo-random ciphertexts. In: Computer Security–ESORICS 2004: 9th European Symposium on Research in Computer Security, Sophia Antipolis, France, September 13-15, 2004. Proceedings 9. pp. 335–351. Springer (2004)

[24] Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: Proceedings of the twenty-second annual ACM symposium on Theory of computing. pp. 427–437 (1990)

[25] Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Advances in Cryptology—EUROCRYPT'99: International Conference on the Theory and Application of Cryptographic Techniques, Proceedings 18. pp. 223–238. Springer (1999)

[26] Persiano, G., Phan, D.H., Yung, M.: Anamorphic encryption: Private communication against a dictator. In: Advances in Cryptology–EUROCRYPT 2022: 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Part II. pp. 34–63. Springer (2022)

[27] Prabhakaran, M., Rosulek, M.: Rerandomizable rcca encryption. In: Advances in Cryptology-CRYPTO 2007: 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007. Proceedings 27. pp. 517–534. Springer (2007)

[28] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM (JACM) **56**(6), 1–40 (2009)

[29] Russell, A., Tang, Q., Yung, M., Zhou, H.S.: Generic semantic security against a kleptographic adversary. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. pp. 907–922 (2017)

[30] Simmons, G.J.: The prisoners' problem and the subliminal channel. In: Advances in Cryptology: Proceedings of Crypto 83. pp. 51–67. Springer (1984)

[31] Tibouchi, M.: Elligator squared: Uniform points on elliptic curves of prime order as uniform random strings. In: Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers. pp. 139–156. Springer (2014)

[32] Von Ahn, L., Hopper, N.J.: Public-key steganography. In: Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings 23. pp. 323–341. Springer (2004)

[33] Wang, Y., Chen, R., Yang, G., Huang, X., Wang, B., Yung, M.: Receiver-anonymity in rerandomizable rcca-secure cryptosystems resolved. In: Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part IV 41. pp. 270–300. Springer (2021)

[34] Young, A., Yung, M.: Kleptography: Using cryptography against cryptography. In: Advances in Cryptology—EUROCRYPT'97: International Conference on the Theory and Application of Cryptographic Techniques, Proceedings 16. pp. 62–74. Springer (1997)

[35] Young, A., Yung, M.: The prevalence of kleptographic attacks on discrete-log based cryptosystems. In: Advances in Cryptology—CRYPTO'97: 17th Annual International Cryptology Conference, Proceedings 17. pp. 264–276. Springer (1997)

[36] Young, A., Yung, M.: Kleptography from standard assumptions and applications. In: Security and Cryptography for Networks: 7th International Conference, SCN 2010, Amalfi, Italy, September 13-15, 2010. Proceedings 7. pp. 271–290. Springer (2010)

A Omitted Definitions and Proof

A.1 Anamorphic Encryption

Definition A.1 (Receiver-Anamorphic Encryption [26])**.** Let PKE be a public key encryption scheme. We say PKE is receiver-anamorphic if there exists a receiver-anamorphic extension $(\mathsf{aSetup}, \mathsf{aGen}, \mathsf{aEnc}, \mathsf{aDec})$

- $\mathsf{aSetup}(1^n)$ takes as input 1^n, and produces the public parameter $\mathtt{pp}$;
- $\mathsf{aGen}(\mathtt{pp})$ takes as input $\mathtt{pp}$, and produces an anamorphic public/secret key pair $\mathtt{apk}$ and $\mathtt{ask}$, and a double key $\mathtt{dkey}$;
- $\mathsf{aEnc}(\mathtt{dkey}, \mathtt{m}_0, \mathtt{m}_1)$ takes as input $\mathtt{dkey}$, a normal plaintext $\mathtt{m}_0$ and an anamorphic plaintext $\mathtt{m}_1$, and produces an anamorphic ciphertext $\mathtt{act}$;
- $\mathsf{aDec}(\mathtt{dkey}, \mathtt{act})$ takes as input $\mathtt{dkey}$ and $\mathtt{act}$, and returns the anamorphic plaintext $\mathtt{m}_1$,

and for any $\mathtt{m}_0 \in \mathcal{M}$, any $\mathtt{m}_1 \in \overline{\mathcal{M}}$,

$$\Pr\left[\begin{array}{l|ll} & \mathtt{pp} \leftarrow\!\!\$\ \mathsf{aSetup}(1^n) \\ \mathsf{aDec}(\mathtt{dkey}, \mathtt{act}) \neq \mathtt{m}_1 : & (\mathtt{apk}, \mathtt{ask}, \mathtt{dkey}) \leftarrow\!\!\$\ \mathsf{aGen}(\mathtt{pp}) \\ & \mathtt{act} \leftarrow\!\!\$\ \mathsf{aEnc}(\mathtt{dkey}, \mathtt{m}_0, \mathtt{m}_1) \end{array}\right] \leq \mathsf{negl}(n)\,,$$

where $\mathcal{M}$ and $\overline{\mathcal{M}}$ are the space of normal and anamorphic plaintext respectively.

Definition A.2 (Secure Receiver-AME [26])**.** Let PKE be a receiver-AME associated with extension $(\mathsf{aSetup}, \mathsf{aGen}, \mathsf{aEnc}, \mathsf{aDec})$. PKE is a secure receiver-AME if following conditions hold,

- PKE is CPA secure;
- For any plaintext $\hat{m} \in \mathcal{M}$, $\mathsf{fAME}_{\hat{m}} = (\mathsf{aSetup}, \mathsf{aGen}_3, \mathsf{aEnc}_{1,\hat{m}}, \mathsf{aDec})$ is a symmetric encryption scheme, where aGen_i denotes selecting the ith component of the triplet generated by aGen as the output, $\mathsf{aEnc}_{1,\hat{m}}$ denotes running aEnc with the first message $\mathtt{m}_0 = \hat{m}$;
- For any PPT adversary $\mathcal{D}$,

$$|\Pr[\mathsf{nGame}_{\mathsf{PKE},\mathcal{D}}(n) = 1] - \Pr[\mathsf{faGame}_{\mathsf{PKE},\mathcal{D}}(n) = 1]| \leq \mathsf{negl}(n)\,.$$

$\mathsf{nGame}_{\mathsf{PKE},\mathcal{D}}(n)$	$\mathsf{faGame}_{\mathsf{PKE},\mathcal{D}}(n)$
$\mathtt{pp} \leftarrow\!\!\$\ \mathsf{Setup}(1^n)$	$\mathtt{pp} \leftarrow\!\!\$\ \mathsf{aSetup}(1^n)$
$(\mathtt{pk}, \mathtt{sk}) \leftarrow\!\!\$\ \mathsf{Gen}(\mathtt{pp})$	$(\mathtt{apk}, \mathtt{ask}, \mathtt{dkey}) \leftarrow\!\!\$\ \mathsf{aGen}(\mathtt{pp})$
$b \leftarrow \mathcal{D}^{\mathrm{ENC}(\cdot,\cdot)}(\mathtt{pp}, \mathtt{pk}, \mathtt{sk})$	$b \leftarrow \mathcal{D}^{\mathrm{AENC}(\cdot,\cdot)}(\mathtt{pp}, \mathtt{apk}, \mathtt{ask})$
return b	**return** b
$\mathrm{ENC}(\mathtt{m}_0, \mathtt{m}_1)$	$\mathrm{AENC}(\mathtt{m}_0, \mathtt{m}_1)$
return $\mathsf{Enc}(\mathtt{pk}, \mathtt{m}_0)$	**return** $\mathsf{aEnc}(\mathtt{dkey}, \mathtt{m}_0, \mathtt{m}_1)$

Figure 16: Definitions of game $\mathsf{nGame}_{\mathsf{PKE},\mathcal{D}}(n)$ and $\mathsf{faGame}_{\mathsf{PKE},\mathcal{D}}(n)$.

$\mathsf{Ideal}_{\mathsf{PKE},\mathcal{A}}(n)$	$\mathsf{Real}^{\mathsf{fRandom}}_{\mathsf{PKE},\mathcal{A}}(n)$
$\mathtt{pp} \leftarrow\!\!\$\ \mathsf{Setup}(1^n)$	$\mathtt{pp} \leftarrow\!\!\$\ \mathsf{Setup}(1^n)$
$(\mathtt{fpk}, \mathtt{fsk}) \leftarrow\!\!\$\ \mathsf{Gen}(\mathtt{pp})$	$(\mathtt{fpk}, \mathtt{fsk}), (\mathtt{dpk}, \mathtt{dsk}) \leftarrow\!\!\$\ \mathsf{Gen}(\mathtt{pp})$
$b \leftarrow \mathcal{A}^{\mathrm{ENC}(\cdot,\cdot)}(\mathtt{pp}, \mathtt{fpk})$	$b \leftarrow \mathcal{A}^{\mathrm{ENC}'(\cdot,\cdot)}(\mathtt{pp}, \mathtt{fpk})$
return b	**return** b
$\mathrm{ENC}(\mathtt{fm}, \mathtt{dm})$	$\mathrm{ENC}'(\mathtt{fm}, \mathtt{dm})$
$r \leftarrow\!\!\$\ \mathcal{R}$	$R \leftarrow\!\!\$\ \mathsf{fRandom}(\mathtt{fpk}, \mathtt{fm}, \mathtt{dpk}, \mathtt{dm})$
return $\mathsf{Enc}(\mathtt{fpk}, \mathtt{fm}; r)$	**return** $\mathsf{Enc}(\mathtt{fpk}, \mathtt{fm}; r)$

Figure 17: Definition of game $\mathsf{Ideal}_{\mathsf{PKE},\mathcal{A}}(n)$ and $\mathsf{Real}^{\mathsf{fRandom}}_{\mathsf{PKE},\mathcal{A}}(n)$.

Definition A.3 (Sender-Anamorphic Encryption [26])**.** Let PKE be a public key encryption scheme. We say PKE is sender-anamorphic if there exists an algorithm fRandom that takes as input forced public key $\mathtt{fpk}$, forced plaintext $\mathtt{fm}$, duplicate public key $\mathtt{dpk}$ and duplicate plaintext $\mathtt{dm}$, and returns randomness r^* such that for any $\mathtt{fm} \in \mathcal{M}$, any $\mathtt{dm} \in \overline{\mathcal{M}}$, we have

$$\Pr\left[\mathsf{Dec}(\mathtt{dsk}, \mathtt{ct}) \neq \mathtt{dm} : \begin{array}{l} \mathtt{pp} \leftarrow\!\!\$\ \mathsf{Setup}(1^n) \\ (\mathtt{fpk}, \mathtt{fsk}), (\mathtt{dpk}, \mathtt{dsk}) \leftarrow\!\!\$\ \mathsf{Gen}(\mathtt{pp}) \\ r^* \leftarrow\!\!\$\ \mathsf{fRandom}(\mathtt{fpk}, \mathtt{fm}, \mathtt{dpk}, \mathtt{dm}) \\ \mathtt{ct} := \mathsf{Enc}(\mathtt{fpk}, \mathtt{fm}; r^*) \end{array}\right] \leq \mathsf{negl}(n)\,.$$

Definition A.4 (Secure Sender-AME [26])**.** Let PKE be a sender-AME associated with algorithm fRandom. We say PKE is a *secure sender-AME* if 1) PKE is CPA secure, and 2) for any PPT adversary $\mathcal{A}$ in Fig. 17,

$$\left|\Pr[\mathsf{Ideal}_{\mathsf{PKE},\mathcal{A}}(n) = 1] - \Pr[\mathsf{Real}^{\mathsf{fRandom}}_{\mathsf{PKE},\mathcal{A}}(n) = 1]\right| \leq \mathsf{negl}(n)\,.$$

A.2 Symmetric Encryption

A symmetric encryption scheme SE consists of following algorithms:

- $\mathsf{Gen}(1^n)$ takes as input 1^n, and returns a key $\mathtt{k}$ which is an implicit input of encryption and decryption algorithms.
- $\mathsf{Enc}(\mathtt{k}, \mathtt{m})$ takes as input $\mathtt{k}$ and a plaintext $\mathtt{m}$, and returns a ciphertext $\mathtt{ct}$.
- $\mathsf{Dec}(\mathtt{k}, \mathtt{ct})$ take as input $\mathtt{k}$ and $\mathtt{ct}$, and returns $\mathtt{m}$ or an abort symbol $\perp$.

We say SE has indistinguishable encryptions in the presence of an eavesdropper if for any PPT adversary $\mathcal{A}$,

$$\left| \Pr\left[b = b' : \begin{array}{l} (\mathtt{m}_0, \mathtt{m}_1) \leftarrow \mathcal{A}(1^n) \\ \mathtt{k} \leftarrow\!\!\$\ \mathsf{Gen}(1^n) \\ b \leftarrow\!\!\$\ \{0,1\} \\ \mathtt{ct} \leftarrow\!\!\$\ \mathsf{Enc}(\mathtt{k}, \mathtt{m}_b) \\ b' \leftarrow \mathcal{A}(\mathtt{ct}) \end{array} \right] - \frac{1}{2} \right| \leq \mathsf{negl}(n)\,.$$

A.3 On the CPA Security of Hybrid PKE

Theorem A.1. *Let* PKE *be a hybrid PKE in Fig. 5 with* KEM *satisfying key-pseudo-randomness and* DEM *that has indistinguishable encryptions in the presence of an eavesdropper. Then,* PKE *is CPA secure.*

Proof. Let $\mathbf{H}_0$ denote the CPA game for PKE with the challenge ciphertext $\mathtt{ct}^* = (C, \pi, D)$. We consider game $\mathbf{H}_1$ that is the same as $\mathbf{H}_0$ except that the key K used to encrypt $\mathtt{m}_b$ is uniformly sampled from $\mathcal{K}_{\mathsf{KEM}}$.

Lemma A.2. *By the key-pseudo-randomness of* KEM, $\mathbf{H}_0 \approx_c \mathbf{H}_1$.

Proof. We show how to break the key-pseudo-randomness of KEM with the adversary $\mathcal{D}$ distinguishing $\mathbf{H}_0$ and $\mathbf{H}_1$. The adversary $\mathcal{A}$ in the game of key-pseudo-randomness receives $(\mathtt{pp}, \mathtt{ek}, K, C)$, then generates $(\mathtt{tk}, \mathtt{vk}) \leftarrow\!\!\$\ \mathsf{KEM.Tk}(\mathtt{pp})$, sets $\mathtt{pk} = (\mathtt{ek}, \mathtt{tk})$ and forwards $(\mathtt{pp}, \mathtt{pk})$ to $\mathcal{D}$. After receiving a pair of plaintexts $(\mathtt{m}_0, \mathtt{m}_1)$ from $\mathcal{D}$, $\mathcal{A}$ picks a random bit b, computes $D := \mathsf{DEM.Enc}(K, \mathtt{m}_b)$, $\pi := \mathsf{KEM.Vf}(\mathtt{vk}, C)$ and sets (C, π, D) as the challenge ciphertext.

If K is generated by $\mathsf{KEM.Kg}$, then $\mathcal{A}$ simulates $\mathbf{H}_0$ for $\mathcal{D}$. Otherwise, K is uniformly sampled from $\mathcal{K}_{\mathsf{KEM}}$, and $\mathcal{A}$ simulates $\mathbf{H}_1$ for $\mathcal{D}$. □

Lemma A.3. *By the indistinguishable encryptions of* DEM, *the advantage of adversary* $\mathcal{A}$ *in* $\mathbf{H}_1$ *is negligible.*

Proof. We show how to break the indistinguishable encryptions of DEM with adversary $\mathcal{A}$ in $\mathbf{H}_1$. The simulator $\mathcal{B}$ computes $\mathtt{pp} \leftarrow\!\!\$\ \mathsf{Setup}(1^n)$, $(\mathtt{pk}, \mathtt{sk}) \leftarrow\!\!\$\ \mathsf{Gen}(\mathtt{pp})$, and forwards $(\mathtt{pp}, \mathtt{pk})$ to $\mathcal{A}$. After receiving $(\mathtt{m}_0, \mathtt{m}_1)$ from $\mathcal{A}$, $\mathcal{B}$ forwards $(\mathtt{m}_0, \mathtt{m}_1)$ to its challenger, and receives a challenge ciphertext D^*. Then, $\mathcal{B}$ computes (C, π) using $\mathtt{pk}$ as in $\mathsf{PKE.Enc}$, and sets (C, π, D^*) as the challenge ciphertext for $\mathcal{A}$.

The simulation above is perfect. $\mathcal{B}$ outputs the bit returned by $\mathcal{A}$. □

The CPA security of PKE follows from Lemma A.2 and Lemma A.3. □

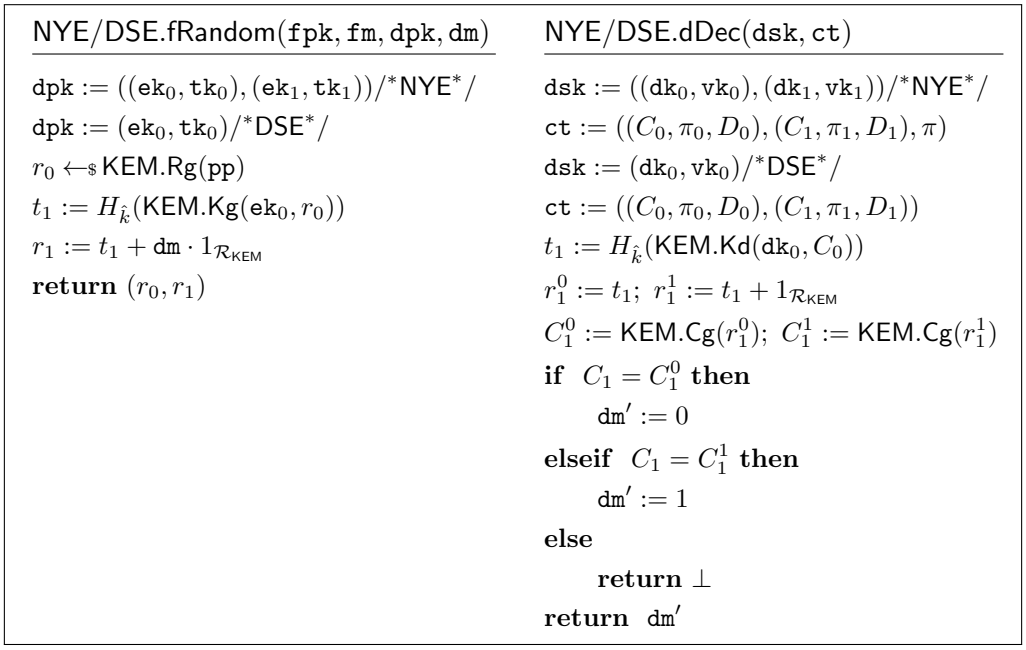

Figure 18: Sender-anamorphic extension for cryptosystem over hybrid PKE.

B Construction III: Cryptosystem over Hybrid PKE

B.1 Cryptosystems over Hybrid PKE

Naor-Yung paradigm. The Naor-Yung paradigm [24] gives a CCA secure PKE from a CPA secure PKE and a simulation sound NIZK.

In a Naor-Yung cryptosystem NYE over PKE in Fig. 6., the key generation algorithm runs PKE.Gen to generate two key pairs $(\mathtt{pk}_0, \mathtt{sk}_0)$ and $(\mathtt{pk}_1, \mathtt{sk}_1)$, and samples a random string Σ. The public key in NYE consists of $\mathtt{pk}_0$, $\mathtt{pk}_1$ and Σ. To encrypt plaintext $\mathtt{m}$, the encryption algorithm samples r_0, r_1 from the randomness space of PKE (i.e., $\mathcal{R}_{\mathsf{KEM}}$), computes $\mathtt{ct}_0 = \mathsf{PKE.Enc}(\mathtt{pk}_0, \mathtt{m}; r_0)$, $\mathtt{ct}_1 = \mathsf{PKE.Enc}(\mathtt{pk}_1, \mathtt{m}; r_1)$, and generates a proof π that $\mathtt{ct}_0$ and $\mathtt{ct}_1$ encrypt same plaintext $\mathtt{m}$. The ciphertext in NYE consists of $\mathtt{ct}_0$, $\mathtt{ct}_1$ and π.

Double-Strand Paradigm. The double-strand paradigm [16, 27, 33] is used to build universal cryptosystem where the ciphertext can be rerandomized without any extra public information.

In a double-strand cryptosystem DSE over PKE in Fig. 6, the key generation algorithm runs PKE.Gen to generate a key pair $(\mathtt{pk}, \mathtt{sk})$. To encrypt plaintext $\mathtt{m}$, the encryption algorithm samples r_0, r_1 from $\mathcal{R}_{\mathsf{KEM}}$, computes $\mathtt{ct}_0 = \mathsf{PKE.Enc}(\mathtt{pk}, \mathtt{m}; r_0)$ and $\mathtt{ct}_1 = \mathsf{PKE.Enc}(\mathtt{pk}, 1; r_1)$, where 1 is the identity element. The ciphertext in DSE consists of $\mathtt{ct}_0$ and $\mathtt{ct}_1$.

B.2 Sender-Anamorphic Extension

Fig. 18 depicts the sender-anamorphic extension for cryptosystem over hybrid PKE. The feature that one ciphertext includes two hybrid PKE ciphertexts enables such cryptosystem to be a 1-sender-AME for 1-bit duplicate message. The strong security and robustness of this 1-sender-AME is obvious, and we omit the proofs here.

C Examples

C.1 ℓ-Receiver-AME Does Not Imply Symmetric ASA

We first recall the well-known ElGamal encryption scheme and then provide its receiver-anamorphic extension with $\ell = 2$.

ElGamal Encryption.

- $\mathsf{Setup}(1^n)$ samples a random generator $g \leftarrow\!\!\$\, \mathbb{G}$ from cyclic group $\mathbb{G}$ of prime order p, and outputs $\mathtt{pp} = (p, g, \mathbb{G})$;
- $\mathsf{Gen}(\mathtt{pp})$ samples $a \leftarrow\!\!\$\, \mathbb{Z}_p$, computes $A = g^a$, and outputs $(\mathtt{pk}, \mathtt{sk}) = (A, a)$;
- $\mathsf{Enc}(\mathtt{pk}, \mathtt{m})$ samples $r \leftarrow\!\!\$\, \mathbb{Z}_p$ and outputs $\mathtt{ct} = (C_1, C_2) = (g^r, A^r \cdot \mathtt{m})$;
- $\mathsf{Dec}(\mathtt{sk}, \mathtt{ct})$ outputs $\mathtt{m}' = C_2/(C_1^a)$.

Receiver-Anamorphic Extension ($\ell = 2$)**.**

- $\mathsf{aSetup}(1^n)$ is the same as $\mathsf{Setup}(1^n)$;
- $\mathsf{aGen}(\mathtt{pp})$ generates $(A, a), (B, b) \leftarrow\!\!\$\, \mathsf{Gen}(\mathtt{pp})$, samples $k \leftarrow\!\!\$\, \mathbb{Z}_p$ and outputs key pairs $\{(A, a), (B, b)\}$ and a double key $\mathtt{dkey} = (A, b, k)$;
- $\mathsf{aEnc}(\mathtt{dkey}, \mathtt{M}, \bar{\mathtt{m}})$ generates $(C_{11}, C_{12}) \leftarrow\!\!\$\, \mathsf{Enc}(A, \mathtt{m}_1)$, computes $C_{21} = C_{11}^k \cdot \bar{\mathtt{m}}$, $C_{22} = C_{21}^b \cdot \mathtt{m}_2$ and outputs $\mathtt{ACT} = \{(C_{11}, C_{12}), (C_{21}, C_{22})\}$;
- $\mathsf{aDec}(\mathtt{dkey}, \mathtt{ACT})$ outputs $\bar{\mathtt{m}}' = C_{21}/C_{11}^k$.

Theorem C.1. *ElGamal encryption scheme* El *is a secure 2-receiver-AME with anamorphic plaintext space* $\mathbb{G}$*.*

Proof. One can easily verify the correctness of 2-receiver-AME for ElGamal scheme and for any plaintext set $\mathtt{M} \in \mathbb{G}^2$, $\mathsf{fAME}_{\mathtt{M}} = (\mathsf{aSetup}, \mathsf{aGen}_5, \mathsf{aEnc}_{1,\mathtt{M}}, \mathsf{aDec})$ is a symmetric encryption scheme. Next, we prove that for any PPT adversary $\mathcal{D}$, the advantage of distinguishing games $\mathsf{nGame}_{2,\mathsf{El},\mathcal{D}}(n)$ and $\mathsf{faGame}_{2,\mathsf{El},\mathcal{D}}(n)$ in Fig. 10 is negligible. The only difference between these two games is the encryption oracle. We claim that the distributions of the second ciphertext (C_{21}, C_{22}) generated by aEnc and normal ciphertext outputted by Enc are computationally indistinguishable. Since k is unknown to $\mathcal{D}$, C_{21} is uniformly distributed over $\mathbb{G}$ in $\mathcal{D}$'s view. Note that C_{22} is computed using secret key b, (C_{21}, C_{22}) is a valid ciphertext of $\mathtt{m}_2$ under public key B. □

We claim this 2-receiver-AME does not imply symmetric ASA on ElGamal scheme. It is unlikely to build the subversion key generation algorithm Gen of ASA with algorithm aGen above. In particular, the double key $\mathtt{dkey}$ outputted by aGen includes the secret key b, while the subversion key $\mathtt{skey}$ in ASA is independent of public and secret keys.

C.2 Asymmetric ASA Does Not Imply ℓ-Sender-AME

We first depict an asymmetric ASA on Cramer-Shoup encryption scheme and then demonstrate it does not imply ℓ-sender-AME.

Cramer-Shoup Encryption. Let $H : \mathbb{G} \times \mathbb{G} \times \mathbb{G} \to \mathbb{Z}_p$ be a collision-resilient hash function.

- $\mathsf{Setup}(1^n)$ samples random generators $g_1, g_2 \leftarrow\!\!\$\, \mathbb{G}$ from cyclic group $\mathbb{G}$ of prime order p, and outputs $\mathtt{pp} = (p, g_1, g_2, \mathbb{G})$;

- $\mathsf{Gen}(\mathtt{pp})$ samples $x_1, x_2, y_1, y_2, z_1, z_2 \leftarrow\!\!\$\, \mathbb{Z}_p$, computes $X = g_1^{x_1} g_2^{x_2}$, $Y = g_1^{y_1} g_2^{y_2}$, $Z = g_1^{z_1} g_2^{z_2}$ and outputs $(\mathtt{pk}, \mathtt{sk}) = ((X, Y, Z), (x_1, x_2, y_1, y_2, z_1, z_2))$;
- $\mathsf{Enc}(\mathtt{pk}, \mathtt{m})$ samples $r \leftarrow\!\!\$\, \mathbb{Z}_p$ and outputs $\mathtt{ct} = (C_1, C_2, C_3, C_4) = (g_1^r, g_2^r, X^r \cdot \mathtt{m}, (YZ^\gamma)^r$ where $\gamma = H(C_1, C_2, C_3)$;
- $\mathsf{Dec}(\mathtt{sk}, \mathtt{ct})$ computes $\gamma' = H(C_1, C_2, C_3), \mathtt{m}' = C_3/(C_1^{x_1} C_2^{x_2})$ and checks $C_4 = C_1^{y_1 + z_1\gamma'} C_2^{y_2 + z_2\gamma'}$. If holds, outputs $\mathtt{m}'$; otherwise, outputs $\perp$.

Asymmetric ASA. Let $\mathcal{F} = \{F_k\}_{k \in \mathcal{K}}$ be a keyed and entropy smoothing hash function family associated with key space $\mathcal{K}$, groups $\mathbb{G}, \mathbb{Z}_p$ and hash function $F_k : \mathbb{G} \to \mathbb{Z}_p$.

- $\mathsf{ASA.Gen}(\mathtt{pp})$ samples $a \leftarrow\!\!\$\, \mathbb{Z}_p$, computes $A = g_1^a$, and outputs $(\mathtt{psk}, \mathtt{ssk}) = (A, a)$;
- $\mathsf{ASA.Enc}^2(\mathtt{psk}, \mathtt{sm} \in \{0,1\}, \mathtt{PK}, \mathtt{M})$ computes $\mathtt{ct}_1 \leftarrow\!\!\$\, \mathsf{Enc}(\mathtt{pk}_1, \mathtt{m}_1)$. Let r_1 denote the randomness of $\mathtt{ct}_1$, it computes $r_2 = F_k(A_1^r) + \mathtt{sm}$ and generates $\mathtt{ct}_2 \leftarrow \mathsf{Enc}(\mathtt{pk}_2, \mathtt{m}_2; r_2)$ with randomness r_2. Finally, it outputs $\{\mathtt{ct}_1, \mathtt{ct}_2\}$;
- $\mathsf{ASA.Ext}(\mathtt{ssk}, \{\mathtt{ct}_1, \mathtt{ct}_2\})$ computes $r_2' = F_k(C_{11}^a)$, $C_{21}' = g_1^{r_2'}$ and checks $C_{21}' = C_{21}$. If holds, it outputs 0. Otherwise, it checks $C_{21}' \cdot g_1 = C_{21}$. If holds, it outputs 1; Otherwise, outputs $\perp$.

One can verify the recoverability of ASA above easily. We show that this ASA satisfies secret undetectability as follows.

Theorem C.2. *The asymmetric ASA on Cramer-Shoup encryption above satisfies secret undetectability.*

Proof. Let $\mathbf{H}_0$ denote the game $\mathsf{Undet}_{\mathsf{ASA},\mathcal{D}}(n)$. For secret undetectability, $\mathcal{D}$ is allowed to access the secret keys $\mathtt{SK}$. We assume that $\mathcal{D}$ makes at most q encryption queries.

Let $\mathbf{H}_1$ be the same as $\mathbf{H}_0$ except that in $\mathsf{ASA.Enc}^2$, randomness $r_2 = F_k(B) + \mathtt{sm}$ where $B \leftarrow\!\!\$\, \mathbb{G}$. By the hardness of DDH assumption over $\mathbb{G}$, we have $\mathbf{H}_0 \approx_c \mathbf{H}_1$. In specific, given a DDH instance (g_1^a, g_1^b, g_1^c), the DDH adversary simulates the game for $\mathcal{D}$ by setting $\mathtt{psk} = g_1^a$ and running $\mathsf{ASA.Enc}^2$ as follows. It samples $r \leftarrow\!\!\$\, \mathbb{Z}_p$, computes $C_{11} = (g_1^b)^r$, $C_{12} = g_2^r$ and derives C_{13}, C_{14} using secret key $\mathtt{sk}_1$. Then, it computes $r_2 = F_k((g_1^c)^r) + \mathtt{sm}$ and generates $\mathtt{ct}_2$ using randomness r_2. If $g_1^c = g_1^{ab}$, DDH adversary simulates $\mathbf{H}_0$. Otherwise, it simulates $\mathbf{H}_1$.

Let $\mathbf{H}_2$ be the same as $\mathbf{H}_1$ except that in $\mathsf{ASA.Enc}^2$, randomness $r_2 \leftarrow\!\!\$\, \mathbb{Z}_p$ is uniformly sampled from $\mathbb{Z}_p$. The advantage of adversary $\mathcal{D}$ in $\mathbf{H}_2$ is 0. Let $\mathbf{H}_{1,i}$ be the same as $\mathbf{H}_{1,i-1}$ except that in $\mathsf{ASA.Enc}^2$, randomness $r_2 \leftarrow\!\!\$\, \mathbb{Z}_p$ is uniformly sampled from $\mathbb{Z}_p$ for the i-th encryption query. We have $\mathbf{H}_1 = \mathbf{H}_{1,0}$ and $\mathbf{H}_2 = \mathbf{H}_{1,q}$ By the entropy smoothness of hash function family, we have $\mathbf{H}_{i-1} \approx_c \mathbf{H}_i$ for $i \in [q]$. In specific, given a entropy smoothness instance (k', δ), the adversary simulates the game for $\mathcal{D}$ by setting $r_2 = F_{k'}(\delta) + \mathtt{sm}$ in $\mathsf{ASA.Enc}^2$ for the i-th encryption query. Clearly, if $\delta \leftarrow\!\!\$\, \mathbb{Z}_p$, it simulates $\mathbf{H}_i$. Otherwise, it simulates $\mathbf{H}_{i-1}$. □

We claim that this asymmetric ASA over Cramer-Shoup encryption does not imply ℓ-sender-AME. In particular, the subversion key pair is not generated using the key generation algorithm of encryption scheme, while the fake coin-tossing algorithm $\mathsf{fRandom}$ only takes normal public keys as input. It is impossible to build algorithm $\mathsf{fRandom}$ with algorithms $\mathsf{ASA.Gen}$ and $\mathsf{ASA.Enc}^2$ above.