Compact and Tightly Selective-Opening Secure Public-key Encryption Schemes

Jiaxin Pan and Runzhi Zeng

Department of Mathematical Sciences,
NTNU – Norwegian University of Science and Technology, Trondheim, Norway
jiaxin.pan@ntnu.no, runzhi.zeng@ntnu.no

Abstract. We propose four public-key encryption schemes with tight simulation-based selective-opening security against chosen-ciphertext attacks (SIM-SO-CCA) in the random oracle model. Our schemes only consist of small constant amounts of group elements in the ciphertext, ignoring smaller contributions from symmetric-key encryption, namely, they have compact ciphertexts. Furthermore, three of our schemes have compact public keys as well.

Known (almost) tightly SIM-SO-CCA secure PKE schemes are due to the work of Lyu et al. (PKC 2018) and Libert et al. (Crypto 2017). They have either linear-size ciphertexts or linear-size public keys. Moreover, they only achieve almost tightness, namely, with security loss depending on the security parameters.

Different to them, our schemes are the *first* ones achieving both tight SIM-SO-CCA security and compactness. Our schemes can be divided into two families:

Direct Constructions. Our first three schemes are constructed directly based on the Strong Diffie-Hellman (StDH), Computational DH (CDH), and Decisional DH assumptions. Both their ciphertexts and public keys are compact. Their security loss is a small constant. Interestingly, our CDH-based construction is the first scheme achieving all these advantages based on a weak, search assumption.

A Generic Construction. Our last scheme is the well-known Fujisaki-Okamoto transformation. We show that it can turn a lossy encryption scheme into a tightly SIM-SO-CCA secure PKE. This transformation preserves both tightness and compactness of the underlying lossy encryption, which is in contrast to the non-tight proof of Heuer et al. (PKC 2015).

Keywords. Selective-opening security, public-key encryption, tight security, random oracle model.

1 Introduction

Selective-opening (SO) security is a stronger security notion for encryption schemes. It considers encryption security in the multi-challenge setting. More precisely, an adversary is given multiple challenge ciphertexts and it is allowed to corrupt some of them to get the corresponding randomness. SO security guarantees that even with this additional capability an adversary still cannot learn any information about the remaining 'unopened' messages.

The motivation of constructing SO secure encryption is that removing cryptographic information is hard and expensive in practice and adversaries can hack into a user's computer and reveal the randomness used in generating a ciphertext. In some scenario, it is even a requirement to reveal the randomness to publicly verify a user's computation.

Definitions of Selective-Opening Security. There are two types of definitions for SO security, the indistinguishability-based (IND-based) ones (weak-IND-SO and full-IND-SO) [3,8] and the simulation-based (SIM-based) one (SIM-SO) [3]. They are not polynomial-time equivalent to each other. For SIM-SO security, it requires that for every SO adversary its output can be efficiently simulated by a simulator that sees only the opened messages. SIM-SO notion is the most common one to study [28,20,25,22,29], since it does not require the message distribution to be efficiently conditionally resamplable (cf. [3]). Moreover, previous work showed that SIM-SO-CCA and full-IND-SO-CCA notions are the strongest SO security [8,2,25].

Tight Reductions. When we prove the security of a cryptographic scheme $\mathit{\Pi}$, we often construct a reduction to show that breaking the security of $\mathit{\Pi}$ implies breaking the underlying assumption $\mathit{\Gamma}$. For concrete security, we argue that if an adversary $\mathcal{A}$ has advantage ϵ in breaking $\mathit{\Pi}$ then we have another adversary $\mathcal{B}$ that breaks $\mathit{\Gamma}$ with advantage $\epsilon' = \epsilon/L$, and the factor L is called the security loss.

A cryptographic scheme is called tightly secure if L is a small constant, assuming that the running time of $\mathcal{A}$ is approximately the same as $\mathcal{B}$ (up to a constant factor). A tight reduction can give quantitatively higher guarantees than a loose one. From a more practical perspective, a tight reduction allows shorter key-length recommendations based on the best known attacks against the underlying assumption. This can potentially yield more efficient schemes. Currently, our community aims to reduce the cost for tight security and construct efficient and tightly secure cryptographic schemes (such as the signature scheme in [12]). Hence, it is more desirable to have an efficient and tightly secure scheme, compared to its non-tight counterparts.

Our Goal: Compact PKE with Tight SIM-SO-CCA Security. In this paper, we are interested in efficient and tightly SIM-SO-CCA secure public-key encryption schemes. We aim at schemes with compact ciphertexts and public keys. Here 'compact' means constant-size, and SIM-SO-CCA security provides security against chosen-ciphertext attacks in addition to the SIM-SO security. We discuss the state of the art in approaching this goal as follows:

(Almost) Tight, yet Non-Compact Schemes. While there are compact and tightly IND-CCA secure PKE schemes [16,18], known tightly SIM-SO-CCA PKE schemes [27,29] are still non-compact wrt. either ciphertext size or public key size. Moreover, the security reductions in both schemes are not fully tight, but almost tight (in the terminology of [11]), namely, the security loss depends on the message bit-length that is a polynomial of the security parameter. Although almost tightness is already interesting, our goal is to achieve security loss with small constants, and it was unknown even with random oracles.

To provide more details, the scheme of Lyu et al. [29] is a recent PKE scheme with tight SIM-SO-CCA security, and its ciphertexts consist of $\mathbf{O}(|m|)$ group elements, where $|m|$ is the bit-length of the message. In a nutshell, their construction is a generic construction that tightly turns a IND-CCA secure key encapsulation mechanism (KEM) to a SIM-SO-CCA secure PKE, and their technique is to encrypt the message "bit-by-bit". Hence, their resulting construction does not preserve the compactness of the underlying KEM in terms of ciphertext overhead. Namely, even if we instantiate it with a compact KEM, it cannot give us a compact PKE with tight SIM-SO-CCA. Furthermore, we note that this bit-wise approach is used in many SIM-SO secure schemes [3,14,28].

While the scheme of Libert et al. [27] has compact ciphertexts, its public keys are not compact. Besides the large public key, their encryption algorithm needs to homomorphically evaluate the evaluation circuit of a PRF over GSW [17] ciphertexts that encrypts a PRF key. Hence, their scheme is very impractical.

Compact, yet Non-Tight Schemes. The work of Heuer et al. [20] is an exception to the bit-wise approach. It is the first work that proves SIM-SO-CCA security of practical PKE schemes, such as DHIES [1], OAEP [5], and Fujisaki-Okamoto (FO) [15], in the random oracle model [4]. All these schemes have compact ciphertexts. However, their security reduction is not tight, due to the guessing strategy in their security proofs. For instance, their proof for the FO transformation lose a factor of $\mathbf{O}(\mu \cdot Q_h)$, where μ and Q_h are numbers of challenge ciphertexts and random oracle queries, respectively.

Finally, we stress that, even though there exist compact and (almost) tightly SIM-SO-CPA secure schemes from [3,25], it is not known how to transform them into SIM-SO-CCA by preserving its tightness and compactness. This is the case even in the random oracle model, given the non-tight bounds from the work of Heuer et al. [20].

1.1 Our Contribution

We construct the first compact PKE schemes with tight SIM-SO-CCA security in the random oracle model. More precisely, we propose four PKE schemes following two main ideas. We highlight that our first three schemes achieve tight SIM-SO-CCA security and compact ciphertexts and compact public keys at the same time. Table 1 compares our schemes with other known SO secure PKE schemes under the Diffie-Hellman assumptions.

Three Direct Constructions. Our first construction, $\mathsf{PKE}_{\mathsf{StDH}}$, is a direct construction of tightly SIM-SO-CCA secure PKE based on the strong Diffie-Hellman (StDH) assumption [1]. We then use the twinning technique from [10] to remove the decision oracle in the StDH assumption and construct our second tight scheme (called $\mathsf{PKE}_{\mathsf{TDH}}$) based on the twin DH (TDH) assumption. The TDH assumption is tightly implied by the standard computational DH (CDH)

Scheme	**Security**	**Ass.**	**Loss**	$\vert\mathsf{pk}\vert$	$\vert\mathsf{m}\vert$	$\vert\mathsf{c}\vert - \vert\mathsf{m}\vert$	**RO?**
BHY [3]	IND-SO-CPA	DDH	1	$2\vert\mathbb{G}\vert$	$\vert\mathbb{G}\vert$	$\vert\mathbb{G}\vert$	No
HJR [25]	SIM-SO-CPA	DDH	$\mathbf{O}(\ell)$	$(\ell+1)^2\vert\mathbb{G}\vert$	ℓ	$\vert\mathbb{G}\vert$	No
LLHG [29]	SIM-SO-CCA	DDH	$\mathbf{O}(\ell)$	$6\vert\mathbb{G}\vert$	ℓ	$3\ell\vert\mathbb{G}\vert$	No
DHIES proved in [20]	SIM-SO-CCA	StDH	$\mathbf{O}(\mu)$	$\vert\mathbb{G}\vert$	ℓ	$\vert\mathbb{G}\vert$	Yes
FO proved in [21]	SIM-SO-CCA	DDH	$\mathbf{O}(\mu Q_h)$	$\vert\mathbb{G}\vert$	ℓ	$\vert\mathbb{G}\vert$	Yes
$\mathsf{PKE}_{\mathsf{StDH}}$ (Figure 4)	SIM-SO-CCA	StDH	8	$\vert\mathbb{G}\vert$	ℓ	$2\vert\mathbb{G}\vert$	Yes
$\mathsf{PKE}_{\mathsf{TDH}}$ (Figure 10)	SIM-SO-CCA	CDH	8	$2\vert\mathbb{G}\vert$	ℓ	$2\vert\mathbb{G}\vert$	Yes
$\mathsf{PKE}_{\mathsf{DDH}}$ (Figure 11)	SIM-SO-CCA	DDH	10	$\vert\mathbb{G}\vert$	ℓ	$4\vert\mathbb{G}\vert$	Yes
FO_1 (in full version [7])	IND-SO-CCA	DDH	2	$2\vert\mathbb{G}\vert$	ℓ	$\vert\mathbb{G}\vert$	Yes
FO_2 (Figure 16)	SIM-SO-CCA	DDH	$\mathbf{O}(\ell)$	$(\ell+1)^2\vert\mathbb{G}\vert$	ℓ	$\vert\mathbb{G}\vert$	Yes

Table 1. Comparison of our constructions with other SO secure PKE schemes. We ignore schemes that are non-tight and significantly less efficient than ours. $|\mathbb{G}|$ is the bit-length of group $\mathbb{G}$. ℓ is the message bit-length, which is independent of the group size, and it can be any polynomial in the security parameter λ. μ and Q_h are numbers of challenge ciphertexts and random oracle queries, respectively. The SO security losses of DHIES and FO can be found in [20, Theorem 6] and [21, Theorem 6].

assumption. Hence, this yields the first tightly SIM-SO-CCA secure PKE based on such a standard search assumption.

Both schemes have very short ciphertexts and public keys. Concretely, there are 2 group elements in the ciphertext overhead for $\mathsf{PKE}_{\mathsf{StDH}}$ and $\mathsf{PKE}_{\mathsf{TDH}}$, and 1 element for $\mathsf{PKE}_{\mathsf{StDH}}$'s public key and 2 for $\mathsf{PKE}_{\mathsf{TDH}}$.

We also show that the decision oracle in the proof of $\mathsf{PKE}_{\mathsf{StDH}}$ can be removed using the decisional DH assumption. However, the resulting scheme $\mathsf{PKE}_{\mathsf{DDH}}$ has longer ciphertexts than the previous two, although it is still compact. All these schemes have small-constant security loss and compact ciphertexts and compact public keys.

Fourth Construction: Fujisaki-Okamoto, Revisited. Our last contribution is to prove that a lossy encryption [3] can be transformed to a PKE with tight SO security via the well-known Fujisaki-Okamoto (FO) transformation [15]. The transformation preserves the tightness (up to a small constant) and compactness of the underlying lossy encryption.

Roughly speaking, a lossy encryption scheme has normal and lossy keys. Under normal keys, the scheme behave as a normal PKE. But under lossy keys, there exists an opener that can explain a ciphertext to any message by outputting the suitable randomness. An opener is not necessarily efficient. Especially, if the lossy encryption does not have an efficient opener (e.g., the BHY scheme [3]), then we can only show tight IND-SO-CCA security of the FO transformation. However, if the lossy encryption has an efficient opener (e.g., the HJR scheme [25]), then it yields tight SIM-SO-CCA security of the FO transformation.

Our result implies that tight IND-SO-CCA and SIM-SO-CCA security can be achieved from any assumption that has suitable lossy encryption. For a fair comparison, we implement our generic construction with DDH-based lossy en-

Scheme	Security	Ass.	Bit Security	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{m}\rvert$	$\lvert\mathsf{c}\rvert - \lvert\mathsf{m}\rvert$
BHY [3]	IND-SO-CPA	DDH	128	64	32	32
HJR [25]	SIM-SO-CPA	DDH	120	2113568	32	32
LLHG [29]	SIM-SO-CCA	DDH	120	192	32	24576
DHIES proved in [20]	SIM-SO-CCA	StDH	96	32	32	64
FO proved in [21]	SIM-SO-CCA	CDH	64	32	32	32
$\mathsf{PKE}_{\mathsf{StDH}}$ (Figure 4)	SIM-SO-CCA	StDH	125	32	32	96
$\mathsf{PKE}_{\mathsf{TDH}}$ (Figure 10)	SIM-SO-CCA	CDH	125	64	32	96
$\mathsf{PKE}_{\mathsf{DDH}}$ (Figure 11)	SIM-SO-CCA	DDH	124	32	32	160
$\mathtt{FO}_1$ (in full version [7])	IND-SO-CCA	DDH	127	64	32	32
$\mathtt{FO}_2$ (Figure 16)	SIM-SO-CCA	DDH	120	2113568	32	32

Table 2. Concrete security and efficiency comparison. All schemes are instantiated with P256, and we consider $\mu = 2^{32}$, $q_H = 2^{32}$, $\lvert\mathsf{m}\rvert = 32$ bytes, and the output length of hash is 32 bytes. We consider the concrete security loss in the "Bit Security".

cryption schemes from [3,25]. They both have only 1 group element in the ciphertext (cf. Table 1). Our proof for the FO transformation is compactness- and tightness-preserving. Hence, for SIM-SO-CCA security, since the HJR scheme has non-compact public keys, it is also the case for our scheme. Similarly, the HJR scheme has only almost tightness, so has ours. We suppose that the size of ciphertexts is more critical than that of public keys, since ciphertexts have to be sent frequently over the internet for each communication, while public keys are stored in a server and can be used for a very long time.

Efficiency Comparison. In Table 2 we estimate our concrete efficiency and compare it with other known SO secure schemes. We focus on schemes based the Diffie-Hellman assumptions and ignore those non-tight and significantly less efficient than ours (e.g., [23]). We estimate the efficiency of all schemes using the same NIST P256 curve. According to the corresponding security proofs, we consider the security level achieve by those schemes.

Our schemes significantly reduce the cost for tight SIM-SO-CCA, compared to LLHG. Moreover, our schemes are comparable to the practical PKE schemes, such as FO and DHIES. For instance, our $\mathtt{FO}_2$ has the same ciphertext size, but it achieves a higher level of security, thanks to the tight security proof. Both $\mathsf{PKE}_{\mathsf{StDH}}$ and $\mathsf{PKE}_{\mathsf{TDH}}$ are comparable to DHIES.

Practical Relevance. When a RO-based scheme is implemented in practice, one would instantiate the RO with a hash function, such as SHA-3. For SIM-SO-CCA PKE schemes in the ROM (including the previous work of Heuer et al. [20] and ours), we should be more careful and pay extra attention to the impossibility result of Bellare et al. [2]. More precisely, it shows that if a PKE scheme is binding then it cannot be SIM-SO secure. In a nutshell, it uses the binding property to construct an adversary such that there is no simulator can conclude the SIM-SO security. Hence, in the programmable ROM, the work of Heuer et al. and our schemes can all bypass it, since they are not binding according to the definition in [2]. The programmability is crucial for our proofs.

However, if one simply replaces the RO with, for instance, SHA-3, the situation becomes rather complex. For our fourth construction, it is not binding and the security results remain, since it uses lossy encryption and it allows us to generate encryption collisions. This is also the reason why [2] does not apply to lossy encryption schemes. For the scheme of Heuer et al. and our first three direct constructions, they will become binding in this case. Hence, the impossibility result of Bellare et al. applies, and they cannot have SIM-SO-CCA security. But the attack in [2] does not imply an adversary breaking IND-SO security, which means the scheme of Heuer et al. and our first three direct constructions can have IND-SO-CCA security, since SIM-SO-CCA implies IND-SO-CCA. An alternative solution could be finding a suitable programmable hash function in the standard model to instantiate our first three direction constructions. We leave constructing compact and tight SIM-SO-CCA secure PKE in the standard model as an interesting open problem.

1.2 Technical Overview

TECHNICAL GOAL: OPENABILITY AND TIGHTNESS. Selective-opening security is usually difficult to achieve. This is because the simulator $\mathcal{S}$ has to be able to 'open' any challenge ciphertext by producing the corresponding message and randomness. An adversary can verify whether a ciphertext has been correctly opened using the public encryption algorithm. It is not entirely trivial how to provide this openability efficiently. During the security proof, the simulator needs to embed a problem instance into the unopened ciphertexts, since usually it cannot open a ciphertext with a problem instance. Even worse, achieving tightness introduce an additional layer of complexity to the problem, namely, this opening procedure should be done in a tight fashion.

The work of Heuer et al. provides efficient openability by reprogramming the random oracle (RO) and guessing one unopened ciphertext. This unopened ciphertext will be embedded a problem challenge. We recall Heuer et al.'s strategy [20] of proving DHIES as an example to illustrate the aforementioned challenges in achieving tight SIM-SO-CCA security. The work of Heuer et al. is also the starting point of our work.

We consider the DHIES scheme with one-time pad as the symmetric encryption. Let $\mathbb{G} := \langle g \rangle$ be a group with order p, and $\mathsf{pk} := g^x$ be a public key. A ciphertext C of DHIES has the form

$$C := (R := g^r, \mathsf{d} := K \oplus \mathsf{m}, \mathsf{MAC}_k(R, \mathsf{d}))\,,$$

where $(K, k) := H(R, \mathsf{pk}^r)$ and H is modeled as a RO. MAC_k produces a MAC tag using k.

To prove its SIM-SO-CCA security, we use the strong Diffie-Hellman (StDH) assumption which states that given a StDH instance $(X = g^x, Y)$ and oracle access to $\textsc{dhp}_X$, it is hard to compute Y^x. Here, $\textsc{dhp}_X(\hat{Y}, \hat{Z})$ outputs the Boolean value of $\hat{Z} = \hat{Y}^x$. The reduction for SIM-SO-CCA security of DHIES firstly define $\mathsf{pk} := X$ and guesses the i^*-th ciphertext will not be opened ($i^* \xleftarrow{\$} [\mu]$). Then Y is embeded into C_{i^*} by $R_{i^*} := Y$. By using the DHP_X oracle and the RO patching technique [20], the reduction simulates the whole security game without knowing the secret x. We can prove that the adversary cannot get any information about $(K_{i^*}, k_{i^*}) = H(Y, Y^x)$ unless it computes Y^x, which breaks the StDH assumption. Thus, d_{i^*} is uniformly random and independent of R_{i^*}.

Unfortunately, since the above strategy needs to guess i^*, it requires a loss of μ, and the resulting security is non-tight and depends on the number of challenge ciphertext. One may consider using the random self-reducibility of StDH and embedding a randomized instance into challenge ciphertext C_i as $R_i := Y \cdot g^{s_i}$ where $s_i \xleftarrow{\$} \mathbb{Z}_p$ (for all $i \in [\mu]$). However, after doing so, one cannot open any ciphertext, since the discrete logarithm of Y is unknown. This is why the guessing approach is required.

Our Solution I: DHIES with Double Randomness. Our first solution is a direct improvement on the DHIES scheme by doubling the randomness R in the ciphertext. We only give some rough idea here and refer Section 3 for more details.

More precisely, we modify the generation of ciphertexts in DHIES: Instead of sampling a single r, we firstly choose a random bit $b \xleftarrow{\$} \{0,1\}$, and then we choose $r_b \xleftarrow{\$} \mathbb{Z}_p$ and $R_{1-b} \xleftarrow{\$} \mathbb{G}$ (without knowing R_{1-b}'s discrete logarithm). Our modified DHIES scheme has ciphertexts with form:

$$C = (R_0, R_1, \mathsf{d} = K \oplus \mathsf{m}, h(k, R_0, R_1, \mathsf{d}))\,,$$

where $(K, k) := H(b, R_0, R_1, \mathsf{pk}^{r_b})$, H is a RO, and h is a collision-resistant hash function. We note that sampling a random group element without knowing its discrete logarithm can be done in many widely-used groups like a subgroup of $\mathbb{Z}_q^*$ where q is a safe prime and prime-order elliptic curves.

After the modification, a ciphertext can have two valid randomness, namely, (b, r_b, R_{1-b}) and $(1-b, r_{1-b}, R_b)$, in the view of an adversary, by carefully programming the RO H. Based on this, our simulator can embed the StDH instances to all challenge ciphertexts and open any ciphertext.

Our Solution II: Lossy Encryption. The idea of having multiple valid randomness can be implemented by a lossy encryption, since under its lossy keys a ciphertext can be explained to different messages. Based on this, we use the lossy encryption as a tool to revise the security proof for the Fujisaki-Okamoto transformation and give a tight proof for its SIM-SO-CCA security. Another view of our second solution is that we transform the lossy-encryption-based SIM-SO-CPA secure PKE to a SIM-SO-CCA secure one, tightly.

Open Problems. We leave constructing (almost) tightly SIM-SO-CCA secure PKE with compact ciphertexts and compact public keys in the standard model as an interesting open problem. Moreover, our direction constructions are based on the Diffie-Hellman assumptions. We will study how to extend them in the post-quantum setting (for instance, with lattices).

2 Preliminaries

Let n be an integer. $[n]$ denotes the set $\{1, ..., n\}$. Let $\mathcal{A}$ be an algorithm. If $\mathcal{A}$ is probabilistic, then $y \xleftarrow{\$} \mathcal{A}(x)$ means that the variable y is assigned to the output of $\mathcal{A}$ on input x. If $\mathcal{A}$ is deterministic, then we write $y := \mathcal{A}(x)$. We write $\mathcal{A}^{\mathcal{O}}$ to indicate that $\mathcal{A}$ has classical access to oracle $\mathcal{O}$. $\mathcal{A} \Rightarrow \mathsf{out}$ denotes the event that $\mathcal{A}$ outputs out. Unless we state it explicitly, all our algorithm are probabilistic polynomial-time (PPT). Throughout this paper, λ is the security parameter. The terms such as 'PPT' and 'negligible' are defined wrt λ.

Games. We use the code-based games [6] to define and prove security. We implicitly assume that Boolean flags are initialized to false, numerical types are initialized to 0, sets are initialized to $\emptyset$, while strings are initialized to the empty string ϵ. $\Pr[\mathsf{G}^{\mathcal{A}} \Rightarrow 1]$ denotes the probability that the final output $\mathsf{G}^{\mathcal{A}}$ of game G running an adversary $\mathcal{A}$ is 1. Let $\mathtt{Ev}$ be an (classical and well-defined) event. We write $\Pr[\mathtt{Ev} : \mathsf{G}]$ to denote the probability that $\mathtt{Ev}$ occurs during the game G.

Random Oracle. We use lazy sampling to simulate random oracles in this paper. Let $\mathcal{X}$ and $\mathcal{Y}$ be two finite sets and $H : \mathcal{X} \to \mathcal{Y}$ be a random oracle in a security game G. During the simulation of G, we use a list H to record all query-respond pairs of H. On query x, the game simulator samples $y \xleftarrow{\$} \mathcal{Y}$, sets $\mathsf{H}[x] := y$ (which means that now $H(x) = y$), and then returns y as the respond. We say x has been queried, or simply $x \in \mathsf{H}$, if and only if $\mathsf{H}[x] = y$ for some $y \in \mathcal{Y}$. For $x \notin \mathsf{H}$, we always have $\mathsf{H}[x] = \perp \notin \mathcal{Y}$.

2.1 Cryptographic Assumptions

Let $\mathbb{G}$ be an cyclic group with a generator g and prime order p. Let $X = g^x$ and $Y = g^y$ for some $x, y \in \mathbb{Z}_p$. The CDH value of X and Y is written as $\mathsf{cdh}(X, Y) = g^{xy}$. Here we suppose that $(\mathbb{G}, g, p)$ is a public parameter.

Definition 1 (Multi-Instance DDH (mDDH)). *We say the* mDDH *problem is hard on* $\mathbb{G}$ *if for any* $\mathcal{A}$*, the* mDDH *advantage of* $\mathcal{A}$ *against* $\mathbb{G}$

$$\mathsf{Adv}_{\mathbb{G}}^{\mathsf{mDDH}}(\mathcal{A}) := \Big| \Pr\left[\mathcal{A}(g_1, (g_0^{r_i}, g_1^{r_i})_{i \in [\mu]}) \Rightarrow 1\right] \\ - \Pr\left[\mathcal{A}(g_1, (g_0^{r_i}, g_1^{r'_i})_{i \in [\mu]}) \Rightarrow 1\right] \Big|$$

is negligible, where μ *is the number of challenges,* $g_0 := g$*,* $g_1 := g_0^{\omega}$ *for some* $\omega \xleftarrow{\$} \mathbb{Z}_p$*, and* $r_i, r'_i \xleftarrow{\$} \mathbb{Z}_p$ *for some* $i \in [\mu]$*.*

By the random self-reducibility of DDH [13], mDDH assumption is tightly equivalent to DDH assumption (i.e., single-instance version of mDDH).

Definition 2 (Strong Diffie-Hellman (StDH) Problem [1]). *For a fixed* $X \in \mathbb{G}$*, let* $\textsc{dhp}_X$ *be the gap oracle that given* $(Y', Z') \in \mathbb{G}^2$ *outputs whether* $\mathsf{cdh}(X, Y') = Z'$ *or not. We say the* StDH *problem is hard on* $\mathbb{G}$ *if for any* $\mathcal{A}$*, the* StDH *advantage of* $\mathcal{A}$ *against* $\mathbb{G}$*,* $\mathsf{Adv}_{\mathbb{G}}^{\mathsf{StDH}}(\mathcal{A})$*, is negligible.*

$$\mathsf{Adv}_{\mathbb{G}}^{\mathsf{StDH}}(\mathcal{A}) := \Pr\left[(X, Y) \xleftarrow{\$} \mathbb{G}^2, \mathcal{A}^{\textsc{dhp}_X(\cdot,\cdot)}(X, Y) \Rightarrow \mathsf{cdh}(X, Y)\right]$$

Definition 3 (Twin Diffie-Hellman (TDH) Problem [10]). *For fixed $X_0, X_1 \in \mathbb{G}$, let $2\textsc{dhp}_{X_0,X_1}$ be an oracle that on input $(Y', Z_0', Z_1') \in \mathbb{G}^3$, determines whether $\mathsf{cdh}(X_0, Y') = Z_0'$ and $\mathsf{cdh}(X_1, Y') = Z_1'$. We say the TDH problem is hard on $\mathbb{G}$ if for any $\mathcal{A}$, the TDH advantage of $\mathcal{A}$ against $\mathbb{G}$*

$$\mathsf{Adv}_{\mathbb{G}}^{\mathsf{TDH}}(\mathcal{A}) := \Pr\left[\mathcal{A}^{2\textsc{dhp}_{X_0,X_1}(\cdot,\cdot,\cdot)}(X_0, X_1, Y) \Rightarrow (\mathsf{cdh}(X_0, Y), \mathsf{cdh}(X_1, Y))\right]$$

is negligible, where $X_0, X_1, Y \xleftarrow{\$} \mathbb{G}$.

Definition 4 (Multi-Instance StDH (mStDH)). *Let μ be the number of instance. We say the mStDH problem is hard on $\mathbb{G}$ if for any $\mathcal{A}$, given $X, Y_1, ..., Y_\mu \xleftarrow{\$} \mathbb{G}$, the mStDH advantage of $\mathcal{A}$ against $\mathbb{G}$, $\mathsf{Adv}_{\mathbb{G}}^{\mathsf{mStDH}}(\mathcal{A})$, is negligible.*

$$\mathsf{Adv}_{\mathbb{G}}^{\mathsf{mStDH}}(\mathcal{A}) := \Pr\left[\mathcal{A}^{\textsc{dhp}_X(\cdot,\cdot)}(X, (Y_i)_{i\in[\mu]}) \Rightarrow \mathsf{cdh}(X, Y_i) \textit{ for some } i \in [\mu]\right]$$

Definition 5 (Multi-Instance TDH (mTDH)). *Let μ be the number of instance. We say the mTDH problem is hard on $\mathbb{G}$ if for any $\mathcal{A}$, given $X_0, X_1, Y_1, ...,$ $Y_\mu \xleftarrow{\$} \mathbb{G}$, the mTDH advantage of $\mathcal{A}$ against $\mathbb{G}$, $\mathsf{Adv}_{\mathbb{G}}^{\mathsf{mStDH}}(\mathcal{A})$, is negligible*

$$\begin{aligned}\mathsf{Adv}_{\mathbb{G}}^{\mathsf{mStDH}}(\mathcal{A}) := \Pr\Big[&\mathcal{A}^{2\textsc{dhp}_{X_0,X_1}(\cdot,\cdot,\cdot)}(X_0, X_1, (Y_i)_{i\in[\mu]}) \\ &\Rightarrow (\mathsf{cdh}(X_0, Y_i), \mathsf{cdh}(X_1, Y_i)) \textit{ for some } i \in [\mu]\Big]\end{aligned}$$

The mStDH and mTDH assumptions are tightly implied by the StDH and TDH assumption, respectively. This can be showed naturally by the random self-reducibility of the Diffie-Hellman assumption. We state the lemmas here and leave the proof in our full version paper [7].

Lemma 1 ($\mathsf{StDH} \xrightarrow{\textbf{tight}} \mathsf{mStDH}$). *For any mStDH adversary $\mathcal{A}$, there exists an StDH adversary $\mathcal{B}$ such that $\mathsf{Adv}_{\mathbb{G}}^{\mathsf{mStDH}}(\mathcal{A}) \le \mathsf{Adv}_{\mathbb{G}}^{\mathsf{StDH}}(\mathcal{B})$.*

Lemma 2 ($\mathsf{TDH} \xrightarrow{\textbf{tight}} \mathsf{mTDH}$). *For any mTDH adversary $\mathcal{A}$, there exists an TDH adversary $\mathcal{B}$ such that $\mathsf{Adv}_{\mathbb{G}}^{\mathsf{mTDH}}(\mathcal{A}) \le \mathsf{Adv}_{\mathbb{G}}^{\mathsf{TDH}}(\mathcal{B})$.*

Definition 6 (Collision Resistance). *A hash function h has collision resistance if for all adversary $\mathcal{A}$, the CR advantage of $\mathcal{A}$ against h*

$$\mathsf{Adv}_h^{\mathsf{CR}}(\mathcal{A}) := \Pr\left[x \ne x' \wedge h(x) \ne h(x') | (x, x') \xleftarrow{\$} \mathcal{A}(h)\right]$$

is negligible. A hash function family $\mathcal{H}$ is collision-resistant if for all $h \xleftarrow{\$} \mathcal{H}$, $\mathsf{Adv}_h^{\mathsf{CR}}(\mathcal{A})$ is negligible.

2.2 Public-Key Encryption Scheme

Definition 7 (PKE). *A Public-Key Encryption (PKE) scheme PKE consists of three polynomial-time algorithms $(\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ and a message space $\mathcal{M}$, a randomness space $\mathcal{R}$, and a ciphertext space $\mathcal{C}$. KG outputs a public and secret key pair $(\mathsf{pk}, \mathsf{sk})$. The encryption algorithm Enc, on input pk and a message $\mathsf{m} \in \mathcal{M}$, outputs a ciphertext $c \in \mathcal{C}$. We also write $c := \mathsf{Enc}(\mathsf{pk}, \mathsf{m}; r)$ to indicate the randomness $r \in \mathcal{R}$ explicitly. The decryption algorithm Dec, on input sk and a ciphertext c, outputs a message $m' \in \mathcal{M}$ or a rejection symbol $\perp \notin \mathcal{M}$.*

```
**GAME** $\mathsf{COR}^{\mathcal{A}}_{\mathsf{PKE}}$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}$
$\mathsf{m} \xleftarrow{\$} \mathcal{A}^{\mathcal{O}}(\mathsf{pk}, \mathsf{sk})$
$c = \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$
**if** $\mathsf{Dec}(\mathsf{sk}, c) = \mathsf{m}$ : **return** 1
**return** 0
```

Fig. 1. The COR game for a PKE scheme PKE and $\mathcal{A}$. $\mathcal{A}$ might have access to some oracle $\mathcal{O}$ (e.g., random oracles, decryption oracles). It depends on the specific reduction.

Correctness of PKE. Some of our PKE schemes do not have perfect correctness, and the correctness bound of PKE might depend on some computational bound, e.g., the collision bound of hash function. Following [24], we use a game COR to define PKE correctness.

Definition 8 (PKE Correctness). *Let* $\mathsf{PKE} := (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *be a PKE scheme with message space* $\mathcal{M}$ *and* $\mathcal{A}$ *be an adversary against* PKE*. The* COR *advantage of* $\mathcal{A}$ *is defined as*

$$\mathsf{Adv}^{\mathsf{COR}}_{\mathsf{PKE}}(\mathcal{A}) := \Pr\left[\mathsf{COR}^{\mathcal{A}}_{\mathsf{PKE}} \Rightarrow 1\right],$$

where the COR *game is defined in Figure 1. If there exists a constant* δ *such that for all adversary* $\mathcal{A}$*,* $\mathsf{Adv}^{\mathsf{COR}}_{\mathsf{PKE}}(\mathcal{A}) \leq \delta$*, then we say* PKE *is* $(1-\delta)$*-correct.*

Selective Opening Security. Selective Opening (SO) security preserves confidentiality even if an adversary opens the randomnesses of some ciphertexts. We use simulation-based approach to define SO security as in [20]. We consider two types of SO security definition: Simulation-based SO security against Chosen-Ciphertext Attacks (SIM-SO-CCA, Definition 9) and Indistinguishability-based SO security against Chosen-Ciphertext Attacks (IND-SO-CCA, Definition 10).

Definition 9 (SIM-SO-CCA security). *Let* PKE *be a PKE scheme with message space* $\mathcal{M}$ *and randomness space* $\mathcal{R}$ *and* $\mathcal{A} := (\mathcal{A}_0, \mathcal{A}_1)$ *be an adversary against* PKE*. Let* μ *be the number of challenge ciphertexts Let* Rel *be a relation. We consider two games defined in Figure 2, where* $\mathcal{A}$ *is run in* $\mathsf{REAL\text{-}SO\text{-}CCA}_{\mathsf{PKE}}$ *and a SO simulator* $\mathcal{S} := (\mathcal{S}_0, \mathcal{S}_1)$ *in* $\mathsf{IDEAL\text{-}SO\text{-}CCA}_{\mathsf{PKE}}$*.* $\mathcal{M}_a$ *is a distribution over* $\mathcal{M}$ *chosen by* $\mathcal{A}_0$*. We define the SIM-SO-CCA advantage function*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{SIM\text{-}SO\text{-}CCA}}_{\mathsf{PKE}}(\mathcal{A}, \mathcal{S}, \mu, \mathsf{Rel}) := \Big| &\Pr\left[\mathsf{REAL\text{-}SO\text{-}CCA}^{\mathcal{A}}_{\mathsf{PKE}} \Rightarrow 1\right] \\ &- \Pr\left[\mathsf{IDEAL\text{-}SO\text{-}CCA}^{\mathcal{S}}_{\mathsf{PKE}} \Rightarrow 1\right]\Big|,\end{aligned}$$

PKE *is SIM-SO-CCA secure if, for every adversary* $\mathcal{A}$ *and every relation* Rel*, there exists a simulator* $\mathcal{S}$ *such that* $\mathsf{Adv}^{\mathsf{SIM\text{-}SO\text{-}CCA}}_{\mathsf{PKE}}(\mathcal{A}, \mathcal{S}, \mu, \mathsf{Rel})$ *is negligible.*

Definition 10 (IND-SO-CCA security). *Let* PKE *be a PKE scheme with message space* $\mathcal{M}$ *and randomness space* $\mathcal{R}$ *and* $\mathcal{A} := (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ *be an adversary against* PKE*. Let* μ *be the number of challenge ciphertext.*

```
GAME $\mathsf{REAL\text{-}SO\text{-}CCA}^{\mathcal{A}}_{\mathsf{PKE}}$
$(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KG}$
$(\mathcal{M}_a, st) \xleftarrow{\$} \mathcal{A}_0^{\textsc{Dec}}(\mathsf{pk})$
for $i \in [\mu]$ :
    $\mathbf{m}[i] := m_i \xleftarrow{\$} \mathcal{M}_a$
    $r_i \xleftarrow{\$} \mathcal{R}$
    $\mathbf{c}[i] := \mathsf{Enc}(pk, m_i; r_i)$
$out \xleftarrow{\$} \mathcal{A}_1^{\textsc{Open},\textsc{Dec}}(st, \mathbf{c})$
return $\mathsf{Rel}(\mathcal{M}_a, \mathbf{m}, I, out)$

$\textsc{Dec}(c)$ // for $c \notin \mathbf{c}$
$m := \mathsf{Dec}(\mathsf{sk}, c)$
return $m$
```

```
GAME $\mathsf{IDEAL\text{-}SO\text{-}CCA}^{\mathcal{S}}_{\mathsf{PKE}}$
$(\mathcal{M}_a, st) \xleftarrow{\$} \mathcal{S}_0$
for $i \in [\mu]$ :
    $\mathbf{m}[i] := m_i \xleftarrow{\$} \mathcal{M}_a$
    $\mathbf{m}''[i] := |m_i|$
$out \xleftarrow{\$} \mathcal{S}_1^{\textsc{Open}}(st, \mathbf{m}'')$
return $\mathsf{Rel}(\mathcal{M}_a, \mathbf{m}, I, out)$

$\textsc{Open}(i)$ // $i \in [\mu]$
$I := I \cup \{i\}$
return $(m_i, r_i)$    // $\mathsf{REAL\text{-}SO\text{-}CCA}_{\mathsf{PKE}}$
return $m_i$           // $\mathsf{IDEAL\text{-}SO\text{-}CCA}_{\mathsf{PKE}}$
```

Fig. 2. The SO security games for PKE schemes. $\mathcal{S}_1$ only learn the lengths of challenge messages m_i instead of the challenge ciphertexts.

We consider the game defined in Figure 3. Samp *and* ReSamp *are efficient algorithms output by* $\mathcal{A}_0$*, where* Samp *outputs* μ *messages according to some distribution (determined by* $\mathcal{A}_0$*) over* $\mathcal{M}$*, and* $\mathsf{ReSamp}(I, \mathbf{m}_0)$ *resamples* $\mathbf{m}_0[i]$ *for* $i \notin I$ *according to the same distribution of* Samp *and then outputs* $\mathbf{m}_1$*. For* $i \in I$*,* $\mathbf{m}_0[i] = \mathbf{m}_1[i]$*. We define the IND-SO-CCA advantage function*

$$\mathsf{Adv}^{\mathsf{IND\text{-}SO\text{-}CCA}}_{\mathsf{PKE}}(\mathcal{A}, \mu) := \Big| \Pr\Big[\mathsf{IND\text{-}SO\text{-}CCA}^{\mathcal{A}}_{\mathsf{PKE},0} \Rightarrow 1\Big] - \Pr\Big[\mathsf{IND\text{-}SO\text{-}CCA}^{\mathcal{A}}_{\mathsf{PKE},1} \Rightarrow 1\Big]\Big|.$$

PKE *is IND-SO-CCA secure if* $\mathsf{Adv}^{\mathsf{IND\text{-}SO\text{-}CCA}}_{\mathsf{PKE}}(\mathcal{A}, \mu)$ *is negligible for any* $\mathcal{A}$*.*

3 Direct Constructions

We construct a compact and tightly SIM-SO-CCA PKE, $\mathsf{PKE}_{\mathsf{StDH}}$, from the strong Diffie-Hellman assumption. We also weaken this assumption using the twinning technique from [10], and the resulting scheme is only based on the Computational Diffie-Hellman assumption at the cost of being less efficient.

3.1 Construction from the Strong Diffie-Hellman Assumption

Let $\mathbb{G}$ be a group with order p. Let $H : \{0,1\} \times \mathbb{G}^3 \to \mathcal{M} \times \{0,1\}^l, h : \{0,1\}^l \times \mathbb{G}^2 \to \{0,1\}^\ell$ be hash functions. We construct a compact and tightly SIM-SO-CCA PKE scheme $\mathsf{PKE}_{\mathsf{StDH}} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ as in Figure 4. The randomness space of $\mathsf{PKE}_{\mathsf{StDH}}$ is the set $\{0,1\} \times \mathbb{Z}_p \times \mathbb{G}$.

<u>Correctness.</u> The correctness of $\mathsf{PKE}_{\mathsf{StDH}}$ depends on the hash function h. If h is not collision resistant, then there is a decryption error. For instance, a ciphertext

```
GAME $\mathsf{IND\text{-}SO\text{-}CCA}^{\mathcal{A}}_{\mathsf{PKE},b}$
$(\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{KG}$
$(\mathsf{Samp},\mathsf{ReSamp},\mathsf{st}_0) \xleftarrow{\$} \mathcal{A}_0(\mathsf{pk})$
$\mathbf{m}_0 \xleftarrow{\$} \mathsf{Samp}$
for $i \in [\mu]$ :
    $r_i \xleftarrow{\$} \mathcal{R}$
    $\mathbf{c}[i] := \mathsf{Enc}(pk,\mathbf{m}_0[i];r_i)$
$\mathsf{st}_1 \xleftarrow{\$} \mathcal{A}_1^{\textsc{Open},\textsc{Dec}}(\mathbf{c},\mathsf{st}_0)$
for $i \in [\mu]\backslash I$ :
    $\mathbf{m}_1[i] := \mathsf{ReSamp}(I,\mathbf{m}_0)$
$b' \xleftarrow{\$} \mathcal{A}_1^{\textsc{Dec}}(\mathsf{st}_1,\mathbf{m}_b)$
return $b'$

$\textsc{Dec}(c)$ // for $c \notin \mathbf{c}$
$m := \mathsf{Dec}(\mathsf{sk},c)$
return $m$

$\textsc{Open}(i)$ // $i \in [\mu]$
$I := I \cup \{i\}$
return $(m_i,r_i)$
```

Fig. 3. The SO security games for PKE schemes. $\mathcal{S}_1$ only learn the lengths of challenge messages m_i instead of the challenge ciphertexts. For $i \in I, \mathbf{m}_0[i] = \mathbf{m}_1[i]$, and for $i \in [\mu]\backslash I$, $\mathbf{m}_0[i]$ has the same distribution with $\mathbf{m}_1[i]$ but not necessary to be the same.

```
$\mathsf{KG}$
$x \xleftarrow{\$} \mathbb{Z}_p$
$X := g^x$
$\mathsf{pk} := X$
$\mathsf{sk} := x$
return $(\mathsf{pk},\mathsf{sk})$

$\mathsf{Enc}(\mathsf{pk},\mathsf{m} \in \mathcal{M})$
$b \xleftarrow{\$} \{0,1\}$
$r_b \xleftarrow{\$} \mathbb{Z}_p$
$R_b := g^{r_b}$
$R_{1-b} \xleftarrow{\$} \mathbb{G}$
$Z_b := \mathsf{pk}^{r_b}$
$(K,k) := H(b,R_0,R_1,Z_b)$
$\mathsf{d} := K \oplus \mathsf{m}$
$\mathcal{T} := h(k,R_0,R_1,\mathsf{d})$
return $(R_0,R_1,\mathsf{d},\mathcal{T})$

$\mathsf{Dec}(\mathsf{sk},(R_0,R_1,\mathsf{d},\mathcal{T}))$
$m := \perp$
$Z_0 := R_0^x, Z_1 := R_1^x$
$(K_0,k_0) := H(0,R_0,R_1,Z_0)$
$\mathcal{T}_0 := h(k_0,R_0,R_1,\mathsf{d})$
$(K_1,k_1) := H(1,R_0,R_1,Z_1)$
$\mathcal{T}_1 := h(k_1,R_0,R_1,\mathsf{d})$
if $\mathcal{T}_0 = \mathcal{T}$ : $\mathsf{m} := \mathsf{d} \oplus K_0$
if $\mathcal{T}_1 = \mathcal{T}$ : $\mathsf{m} := \mathsf{d} \oplus K_1$
return $m$
```

Fig. 4. Our Direct Construction of SIM-SO-CCA secure PKE schemes from the mStDH assumption, $\mathsf{PKE}_{\mathsf{StDH}} = (\mathsf{KG},\mathsf{Enc},\mathsf{Dec})$

c of m is generated using $b = 1$, which means it uses $\tau_1 = h(k_1,R_0,R_1,d)$ with $(K_1,k_1) := H(1,R_0,R_1,Z_1)$. If there is a collision as $h(k_1,R_0,R_1,d) = h(k_0,R_0,R_1,d)$ and $(K_1,k_1) \neq (K_0,k_0)$, then c will be decrypted incorrectly as $m' := d \oplus K_0 \neq m = d \oplus K_1$. Hence, the correctness error $\mathsf{Adv}^{\mathsf{COR}}_{\mathsf{PKE}_{\mathsf{StDH}}}(\mathcal{A})$ is bounded by the collision probability of h. If h is modeled as a random oracle, then $\mathsf{Adv}^{\mathsf{COR}}_{\mathsf{PKE}_{\mathsf{StDH}}}(\mathcal{A}) \leq \frac{q_h}{2^\ell}$. In our tight proof, we require collision resistance of a standard hash function, and thus we use the similar requirement here, namely, $\mathsf{Adv}^{\mathsf{COR}}_{\mathsf{PKE}_{\mathsf{StDH}}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{CR}}_{h}(\mathcal{A})$.

On Sampling of a Group Element. We require that a group element of $\mathbb{G}$ can be sampled without knowing the corresponding exponent. A concrete example is as follow: Let p be a prime s.t. $q = rp + 1$ is also a prime for some r. Let $\mathbb{G}$ be a subgroup of $\mathbb{Z}_q$ and with order p. Canetti et al. [9, Section 4.3.2] showed how to sample a group element from such $\mathbb{G}$ without knowing exponent. Other examples are some widely-used standard elliptic-curve groups, such as NIST P256, NIST P384, and Curve25519. To generate a random point without knowing the exponent, we can pick a random x-coordinate, compute the point, and then use the cofactor to check whether the point is in its prime subgroup.

Theorem 1. $\mathsf{PKE}_{\mathsf{StDH}}$ *in Figure 4 is SIM-SO-CCA secure (Definition 9) if the* mStDH *problem is hard on* $\mathbb{G}$ *and* H *and* h *are modeled as random oracles. For any SIM-SO-CCA adversary* $\mathcal{A}$ *and relation* Rel*, there exists a simulator* $\mathcal{S}$ *and an adversary* $\mathcal{B}$ *such that:*

$$\mathsf{Adv}^{\mathsf{SIM\text{-}SO\text{-}CCA}}_{\mathsf{PKE}_{\mathsf{StDH}}}(\mathcal{A}, \mathcal{S}, \mu, \mathsf{Rel}) \le 8\mathsf{Adv}^{\mathsf{mStDH}}_{\mathbb{G}}(\mathcal{B}) + \frac{2n_H^2}{|\mathcal{M}|} + \frac{2(n_H^2 + n_h^2)}{2^l}$$

where q_H *and* n_{DEC} *are the numbers of* $\mathcal{A}$*'s queries to* H *and* DEC*, respectively, and* μ *is the number of challenge ciphertexts.* $n_H = \mu + q_H + 2n_{\text{DEC}}$ *and* $n_h = \mu + q_h + 2n_{\text{DEC}}$ *are the total numbers of queries to* H *and* h*, respectively.*

By Lemma 1, $\mathsf{PKE}_{\mathsf{StDH}}$ in Figure 4 is SIM-SO-CCA secure under the StDH assumption, and the security reduction is tight.

Corollary 1. $\mathsf{PKE}_{\mathsf{StDH}}$ *in Figure 4 is SIM-SO-CCA secure (Definition 9) if the* StDH *problem is hard on* $\mathbb{G}$ *and* H *and* h *are modeled as random oracles. For any SIM-SO-CCA adversary* $\mathcal{A}$ *and relation* Rel*, there exists a simulator* $\mathcal{S}$ *and an adversary* $\mathcal{B}$ *such that:*

$$\mathsf{Adv}^{\mathsf{SIM\text{-}SO\text{-}CCA}}_{\mathsf{PKE}_{\mathsf{StDH}}}(\mathcal{A}, \mathcal{S}, \mu, \mathsf{Rel}) \le 8\mathsf{Adv}^{\mathsf{StDH}}_{\mathbb{G}}(\mathcal{B}) + \frac{2n_H^2}{|\mathcal{M}|} + \frac{2(n_H^2 + n_h^2)}{2^l}$$

where q_H *and* n_{DEC} *are the numbers of* $\mathcal{A}$*'s queries to* H *and* DEC*, respectively, and* μ *is the number of challenge ciphertexts.* $n_H = \mu + q_H + 2n_{\text{DEC}}$ *and* $n_h = \mu + q_h + 2n_{\text{DEC}}$ *are the total numbers of queries to* H *and* h*, respectively.*

Proof (Theorem 1). The theorem is proved by the game sequence in Figures 5 and 6. In G_0, we use lazy sampling to simulate Random oracles H and h. We assume that from G_0 to G_8, there is no collision among the outputs of random oracle h, the first parts of outputs of H (i.e., K), and the second parts of outputs of H (i.e., k). Let n_H and n_h be the total numbers of queries (including the queries from the game simulator) to H and h, respectively. By collision bounds,

$$\left| \Pr\left[\mathsf{REAL\text{-}SO\text{-}CCA}^{\mathcal{A}}_{\mathsf{PKE}_{\mathsf{StDH}}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_0^{\mathcal{A}} \Rightarrow 1\right] \right| \le \frac{n_H^2}{|\mathcal{M}|} + \frac{n_H^2 + n_h^2}{2^l}$$

GAME G_1: We generate $R_{i,1-b_i} := g^{r_{i,1-b_i}}$ by choosing $r_{i,1-b_i} \xleftarrow{\$} \mathbb{Z}_p$, and compute $Z_{i,1-b_i} := X^{r_{i,1-b_i}}$. This modification does not change $\mathcal{A}$'s view since $R_{i,1-b_i}$ is still distributed uniformly at random. Therefore, we have

$$\Pr\left[\mathsf{G}_0^{\mathcal{A}} \Rightarrow 1\right] = \Pr\left[\mathsf{G}_1^{\mathcal{A}} \Rightarrow 1\right]$$

GAME G_2: We modify DEC oracle. When $\mathcal{A}$ queries DEC on $\mathsf{c} := (R_0, R_1, \mathsf{d}, \mathcal{T})$, if $\mathcal{T}$ is the tag of one of the challenge ciphertexts (i.e., $\mathcal{T} = \mathcal{T}_i$ for some

```
Games $\mathsf{G}_0$-$\mathsf{G}_2$
$(X, x) \xleftarrow{\$} \mathsf{KG}$
$(\mathcal{M}_a, \mathsf{st}) \xleftarrow{\$} \mathcal{A}_0^{\textsc{Dec},H,h}(X)$
for $i \in [\mu]$
    $\mathbf{m}[i] := \mathsf{m}_i \xleftarrow{\$} \mathcal{M}_a$
    $b_i \xleftarrow{\$} \{0,1\}$
    $r_{i,b_i} \xleftarrow{\$} \mathbb{Z}_p, R_{i,b_i} := g^{r_{i,b_i}}$
    $Z_{i,b_i} := X^{r_{i,b_i}}$
    $R_{i,1-b_i} \xleftarrow{\$} \mathbb{G}$                // $\mathsf{G}_0$
    $r_{i,1-b_i} \xleftarrow{\$} \mathbb{Z}_p$              // $\mathsf{G}_1$-$\mathsf{G}_2$
    $R_{i,1-b_i} := g^{r_{i,1-b_i}}$           // $\mathsf{G}_1$-$\mathsf{G}_2$
    $Z_{i,1-b_i} := X^{r_{i,1-b_i}}$           // $\mathsf{G}_1$-$\mathsf{G}_2$
    $(K_i, k_i) := H(b_i, R_{i,0}, R_{i,1}, Z_{i,b_i})$
    $\mathsf{d}_i := \mathsf{m}_i \oplus K_i$
    $\mathcal{T}_i := h(k_i, R_{i,0}, R_{i,1}, \mathsf{d}_i)$
    $\mathbf{c}[i] := (R_{i,0}, R_{i,1}, \mathsf{d}_i, \mathcal{T}_i)$
$out \xleftarrow{\$} \mathcal{A}_1^{\textsc{Open},\textsc{Dec},H,h}(\mathsf{st}, \mathbf{c})$
return $\mathsf{Rel}(\mathcal{M}_a, \mathbf{m}, I, out)$

$\textsc{Open}(i)$
$I := I \cup \{i\}$
$\mathsf{rand} := (b_i, r_{i,b_i}, R_{i,1-b_i})$
return $(\mathsf{m}_i, \mathsf{rand})$

$H(b, R_0, R_1, Z)$
if $\mathsf{H}[b, R_0, R_1, Z] = \perp$:
    $(K, k) \xleftarrow{\$} \mathcal{M} \times \{0,1\}^l$
    $\mathsf{H}[b, R_0, R_1, Z] := (K, k)$
return $\mathsf{H}[b, R_0, R_1, Z]$

$\textsc{Dec}(c)$ // $c \notin \mathbf{c}$
parse $(R_0, R_1, \mathsf{d}, \mathcal{T}) := c$
if $\exists i \in [\mu]$ s.t. $\mathcal{T} = \mathcal{T}_i$     // $\mathsf{G}_2$
    return $\perp$                      // $\mathsf{G}_2$
$\mathsf{m} := \perp$
$Z_0 := R_0^x, Z_1 := R_1^x$
$(K_0, k_0) := H(0, R_0, R_1, Z_0)$
$(K_1, k_1) := H(1, R_0, R_1, Z_1)$
$\mathcal{T}_0 := h(k_0, R_0, R_1, \mathsf{d})$
$\mathcal{T}_1 := h(k_1, R_0, R_1, \mathsf{d})$
if $\mathcal{T}_0 = \mathcal{T}$ : $\mathsf{m} = \mathsf{d} \oplus K_0$
if $\mathcal{T}_1 = \mathcal{T}$ : $\mathsf{m} = \mathsf{d} \oplus K_1$
return $\mathsf{m}$
```

Fig. 5. Games G_0-G_2 for proving Theorem 1. Random oracle h is simulated as usual (i.e., similar to the simulation of H in G_0).

$i \in [\mu]$), then Dec returns $\perp$. By the definition of SIM-SO-CCA security, we have $(R_0, R_1, \mathsf{d}, \mathcal{T}) \notin \mathbf{c}$. Thus, if $\mathcal{T} = \mathcal{T}_i$, we have $(R_0, R_1, \mathsf{d}) \neq (R_{i,0}, R_{i,1}, \mathsf{d}_i)$. From this, we can find a collision for h, since $\mathcal{T}$ must equal to $h(k_0, R_0, R_1, \mathsf{d})$ or $h(k_1, R_0, R_1, \mathsf{d})$. We have assumed there is no collision among the output of h, so we have

$$\Pr\left[\mathsf{G}_1^{\mathcal{A}} \Rightarrow 1\right] = \Pr\left[\mathsf{G}_2^{\mathcal{A}} \Rightarrow 1\right]$$

Game G_3: In this game, we simulate Dec by searching for the corresponding keys from the random oracle queries, instead of computing Z_0, Z_1 as in G_2. Intuitively, this does not change the view of $\mathcal{A}$, since a ciphertext is valid if $\mathcal{A}$ has asked the corresponding random oracle queries before. Otherwise, the ciphertext is invalid and the decryption will only output $\perp$.

Concretely, G_3 use the following three lists $\mathsf{H}_{\mathsf{val}}, \mathsf{H}_{\mathsf{inv}}$, and $\mathsf{H}_{\mathsf{dec}}$ to keep track of the oracle queries to H, and each of them stores a particular type of oracle queries, namely:

- $(b, R_0, R_1, Z) \in \mathsf{H}_{\mathsf{val}}$ if $\mathcal{A}$ has queried H on (b, R_0, R_1, Z) and $Z = R_b^x$. We call this type of hash queries valid.
- $(b, R_0, R_1, Z) \in \mathsf{H}_{\mathsf{inv}}$ if $\mathcal{A}$ has queried H on (b, R_0, R_1, Z) and $Z \neq R_b^x$. We call this type of hash queries invalid.
- $(b, R_0, R_1) \in \mathsf{H}_{\mathsf{dec}}$ if $\mathcal{A}$ has queried Dec with (R_0, R_1) as parts of a ciphertext. It is clear that $\mathsf{H}_{\mathsf{val}} \cap \mathsf{H}_{\mathsf{inv}} = \emptyset$.

```
Games $\mathsf{G}_3$-$\mathsf{G}_9$
01 $(X, x) \xleftarrow{\$} \mathsf{KG}$
02 $(\mathcal{M}_a, \mathsf{st}) \xleftarrow{\$} \mathcal{A}_0^{\textsc{Dec},H,h}(X)$
03 for $i \in [\mu]$
04    $\mathbf{m}[i] := \mathsf{m}_i \xleftarrow{\$} \mathcal{M}_a$
05    $b_i \xleftarrow{\$} \{0,1\}$
06    $r_{i,b_i} \xleftarrow{\$} \mathbb{Z}_p, R_{i,b_i} := g^{r_{i,b_i}}$
07    $Z_{i,b_i} := X^{r_{i,b_i}}$
08    $r_{i,1-b_i} \xleftarrow{\$} \mathbb{Z}_p$
09    $R_{i,1-b_i} := g^{r_{i,1-b_i}}$
10    $Z_{i,1-b_i} := X^{r_{i,1-b_i}}$
11    $(K_i, k_i)$
          $:= H(b_i, R_{i,0}, R_{i,1}, Z_{i,b_i})$            // $\mathsf{G}_3$-$\mathsf{G}_5$
12    $(K_i, k_i) \xleftarrow{\$} \mathcal{M} \times \{0,1\}^l$            // $\mathsf{G}_6$-$\mathsf{G}_9$
13    $\mathsf{d}_i := \mathsf{m}_i \oplus K_i$
14    $\mathcal{T}_i := h(k_i, R_{i,0}, R_{i,1}, \mathsf{d}_i)$
15    $\mathbf{c}[i] := (R_{i,0}, R_{i,1}, \mathsf{d}_i, \mathcal{T}_i)$
16 $out \xleftarrow{\$} \mathcal{A}_1^{\textsc{Open},\textsc{Dec},H,h}(\mathsf{st}, \mathbf{c})$
17 return $\mathsf{Rel}(\mathcal{M}_a, \mathbf{m}, I, out)$

$\textsc{Open}(i)$
18 $I := I \cup \{i\}$
19 $\mathsf{H}_{\mathsf{val}}[b_i, R_{i,0}, R_{i,1}, Z_{i,b_i}]$
          $:= (K_i, k_i)$                      // $\mathsf{G}_6$,$\mathsf{G}_8$-$\mathsf{G}_9$
20 $\mathsf{rand} := (b_i, r_{i,b_i}, R_{i,1-b_i})$  // $\mathsf{G}_3$-$\mathsf{G}_6$,$\mathsf{G}_8$-$\mathsf{G}_9$
21 $\mathsf{H}_{\mathsf{val}}[1-b_i, R_{i,0}, R_{i,1}, Z_{i,1-b_i}]$
          $:= (K_i, k_i)$                      // $\mathsf{G}_7$
22 $\mathsf{rand} := (1-b_i, r_{i,1-b_i}, R_{i,b_i})$           // $\mathsf{G}_7$
23 return $(\mathsf{m}_i, \mathsf{rand})$

$H(b, R_0, R_1, Z)$
24 if $\exists i \in [\mu] \backslash I$ s.t.
       $(b, R_0, R_1, Z) = (1-b_i, R_{i,0}, R_{i,1}, R_{i,1-b_i}^x)$
       abort                                   // $\mathsf{G}_4$-$\mathsf{G}_7$
25 if $\exists i \in [\mu] \backslash I$ s.t.
       $(b, R_0, R_1, Z) = (b_i, R_{i,0}, R_{i,1}, R_{i,b_i}^x)$
       abort                                   // $\mathsf{G}_5$-$\mathsf{G}_7$
26 if $\exists (K, k)$ s.t. $\mathsf{H}_{\mathsf{dec}}[b, R_0, R_1] = (K, k)$
       and $Z = R_b^x$
27    $\mathsf{H}_{\mathsf{val}}[b, R_0, R_1, Z] := (K, k)$
28    $\mathsf{H}_{\mathsf{dec}}[b, R_0, R_1] := \perp$
29 if $\exists (K, k)$ s.t. $\mathsf{H}_{\mathsf{val}}[b, R_0, R_1, Z] = (K, k)$
       or $\mathsf{H}_{\mathsf{inv}}[b, R_0, R_1, Z] = (K, k)$
30    return $(K, k)$
31 else
32    $(K, k) \xleftarrow{\$} \mathcal{M} \times \{0,1\}^l$
33    if $Z = R_b^x$ : $\mathsf{H}_{\mathsf{val}}[b, R_0, R_1, Z] := (K, k)$
34    else $\mathsf{H}_{\mathsf{inv}}[b, R_0, R_1, Z] := (K, k)$
35    return $(K, k)$

$\textsc{Dec}(c)$ // $c \notin \mathbf{c}$
36 parse $(R_0, R_1, \mathsf{d}, \mathcal{T}) =: c$
37 if $\exists i \in [\mu]$ s.t. $\mathcal{T} = \mathcal{T}_i$: return $\perp$        // $\mathsf{G}_3$-$\mathsf{G}_8$
38 $\mathsf{m} := \perp$
39 for $b \in \{0,1\}$:
40    if $\exists (Z, K, k)$ s.t. $\mathsf{H}_{\mathsf{val}}[b, R_0, R_1, Z] = (K, k)$
         or $\exists (K, k)$ s.t. $\mathsf{H}_{\mathsf{dec}}[b, R_0, R_1] = (K, k)$
41       $(K_b, k_b) := (K, k)$
42    else
43       $(K_b, k_b) \xleftarrow{\$} \mathcal{M} \times \{0,1\}^l$
44       $\mathsf{H}_{\mathsf{dec}}[b, R_0, R_1] := (K_b, k_b)$
45    $\mathcal{T}_b := h(k_b, R_0, R_1, \mathsf{d})$
46    if $\mathcal{T}_b = \mathcal{T}$ : $\mathsf{m} = \mathsf{d} \oplus K_b$
47 return $m$
```

Fig. 6. Games G_3-G_9 for proving Theorem 1.

Oracles H and Dec in G_3 are simulated in the following ways:

- Dec oracle: On input $(R_0, R_1, \mathsf{d}, \mathcal{T})$, the simulator tries to search (K_b, k_b) $(b \in \{0,1\})$ from $\mathsf{H}_{\mathsf{val}}$ (see Items 40 and 41). If it fails, the simulator samples a random key pair (K_b, k_b) and store (b, K_b, k_b) in $\mathsf{H}_{\mathsf{dec}}$. Then the simulator decrypts $(R_0, R_1, \mathsf{d}, \mathcal{T})$ as usual.
- H oracle: On input (b, R_0, R_1, Z), the simulator firstly checks if $(b, R_0, R_1) \in \mathsf{H}_{\mathsf{dec}}$. If $(b, R_0, R_1) \in \mathsf{H}_{\mathsf{dec}}$ and $Z = R_b^x$, then the simulator sets $\mathsf{H}_{\mathsf{val}}[b, R_0, R_1, Z] = (K_b, k_b)$ and removes (b, R_0, R_1) from $\mathsf{H}_{\mathsf{dec}}$. Then the simulator checks whether (b, R_0, R_1, Z) has been queried, and if so returns the recorded response (see Items 29 and 30). Otherwise, it determines (b, R_0, R_1, Z) should be added to $\mathsf{H}_{\mathsf{val}}$ or to $\mathsf{H}_{\mathsf{inv}}$ by checking $Z = R_b^x$ (see Items 33 and 34), and samples a fresh (K, k) and records it in $\mathsf{H}_{\mathsf{val}}$ or $\mathsf{H}_{\mathsf{inv}}$. The output distribution of H in this game is still uniformly random.

Now consider the case that $\mathcal{A}$ queries Dec on $(R_0, R_1, \mathsf{d}, \mathcal{T})$ but $\mathcal{A}$ has not queried H on the corresponding H-query of $(R_0, R_1, \mathsf{d}, \mathcal{T})$. In this case, the simulator cannot extract (K_0, k_0) and (K_1, k_1) from $\mathsf{H}_{\mathsf{val}}$. Instead of using x to compute Z_0 and Z_1 as in G_2, the game simulator of G_3 samples fresh key pairs (K_0, k_0) and (K_1, k_1) and adds $(0, R_0, R_1)$ and $(1, R_0, R_1)$ into $\mathsf{H}_{\mathsf{dec}}$. Lately, when $\mathcal{A}$ queries H on (b, R_0, R_1, Z) where $Z = R_b^x$, the game simulator "patches" (b, R_0, R_1, Z) into $\mathsf{H}_{\mathsf{val}}$, i.e., sets $\mathsf{H}_{\mathsf{val}}[b, R_0, R_1, Z] = (K_b, k_b)$, and removes (b, R_0, R_1) from $\mathsf{H}_{\mathsf{dec}}$ (see Items 26 to 28).

We note that the use of these three lists is internal but the outputs of H and Dec are the same as in G_2. Thus,

$$\Pr\left[\mathsf{G}_2^{\mathcal{A}} \Rightarrow 1\right] = \Pr\left[\mathsf{G}_3^{\mathcal{A}} \Rightarrow 1\right]$$

Game G_4: G_4 aborts if $\mathcal{A}$ queries H on $(1-b_i, R_{i,0}, R_{i,1}, Z_{i,1-b_i})$ with $Z_{i,1-b_i} = R_{i,1-b_i}^x$ and $\mathbf{c}[i]$ is not opened for some $1 \leq i \leq \mu$. We note that this abort condition lead to the CDH value of X and $R_{i,1-b_i}$. Hence, we can bound the probability of this abort event with the multi-challenge strong Diffie-Hellman (mStDH) assumption.

```
$\mathcal{B}_1^{\text{DHP}_X}(X, Y_1, ..., Y_\mu)$
01 $Z^* := \perp$
02 $(\mathcal{M}_a, \mathsf{st}) \xleftarrow{\$} \mathcal{A}_0^{\text{Dec},H,h}(X)$
03 for $i \in [\mu]$
04     $\mathbf{m}[i] := \mathsf{m}_i \xleftarrow{\$} \mathcal{M}_a$
05     $b_i \xleftarrow{\$} \{0,1\}$
06     $r_{i,b_i} \xleftarrow{\$} \mathbb{Z}_p, R_{i,b_i} := g^{r_{i,b_i}}$
07     $Z_{i,b_i} := X^{r_{i,b_i}}$
08     $(K_i, k_i) := H(b_i, R_{i,0}, R_{i,1}, Z_{i,b_i})$
09     $R_{i,1-b_i} := Y_i$
10     $\mathsf{d}_i := \mathsf{m}_i \oplus K_i$
11     $\mathcal{T}_i := h(k_i, R_{i,0}, R_{i,1}, \mathsf{d}_i)$
12     $\mathbf{c}[i] := (R_{i,0}, R_{i,1}, \mathsf{d}_i, \mathcal{T}_i)$
13 $out \xleftarrow{\$} \mathcal{A}_1^{\text{Open},\text{Dec},H,h}(\mathsf{st}, \mathbf{c})$
14 if $Z^* = \perp : Z^* \xleftarrow{\$} \mathbb{G}$
15 return $Z^*$

$H(b, R_0, R_1, Z)$
16 if $\exists i \in [\mu] \backslash I$ s.t.
       $(b, R_0, R_1) = (1 - b_i, R_{i,0}, R_{i,1})$
       and $\text{DHP}_X(R_{i,1-b_i}, Z) = 1$
17     $Z^* := Z$          // records the solution
18     Aborts the simulation and returns $Z^*$
19 if $\exists (K, k)$ s.t. $\mathsf{H}_{\mathsf{dec}}[b, R_0, R_1] = (K, k)$
       and $\text{DHP}_X(R_b, Z) = 1$
20     $\mathsf{H}_{\mathsf{val}}[b, R_0, R_1, Z] := (K, k)$
21     $\mathsf{H}_{\mathsf{dec}}[b, R_0, R_1] := \perp$
22 if $\exists (K, k)$ s.t. $\mathsf{H}_{\mathsf{val}}[b, R_0, R_1, Z] = (K, k)$
       or $\mathsf{H}_{\mathsf{inv}}[b, R_0, R_1, Z] = (K, k)$
23     return $(K, k)$
24 else
25     $(K, k) \xleftarrow{\$} \mathcal{M} \times \{0,1\}^l$
26     if $\text{DHP}_X(R_b, Z) = 1$
27         $\mathsf{H}_{\mathsf{val}}[b, R_0, R_1, Z] := (K, k)$
28     else $\mathsf{H}_{\mathsf{inv}}[b, R_0, R_1, Z] := (K, k)$
29     return $(K, k)$
```

Fig. 7. mStDH adversary $\mathcal{B}_1$ in bounding the difference between G_3 and G_4. The simulation of Dec and h are the same as in G_4 in Figure 6.

The reduction $\mathcal{B}_1$ against the mStDH assumption is constructed in Figure 7. On input $(X, Y_1, ..., Y_\mu)$, $\mathcal{B}_1$ sets $R_{i,1-b_i} := Y_i$. It can simulate G_4 without x, since it can use its DHP_X oracle to check whether $Z = \mathsf{cdh}(X, R_{i,1-b_i})$. Therefore,

$$\left|\Pr\left[\mathsf{G}_3^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_4^{\mathcal{A}} \Rightarrow 1\right]\right| \leq \mathsf{Adv}_{\mathbb{G}}^{\mathsf{mStDH}}(\mathcal{B}_1)$$

Game G_5: We introduce the abort rule in the H oracle: If $\mathcal{A}$ queries $H(b_i, R_{i,0}, R_{i,1}, R^x_{i,b_i})$ for some $i \in [\mu]$, then G_5 aborts. Let Bad be this querying event and Bad_j be the event that Bad happens in G_j. The adversary cannot detect this modification unless it triggers Bad_5. We have

$$\left|\Pr\left[\mathsf{G}_4^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_5^{\mathcal{A}} \Rightarrow 1\right]\right| \leq \Pr\left[\text{Bad}_5\right]$$

Here we cannot bound $\Pr\left[\text{Bad}_5\right]$ using mStDH yet, since if the adversary queries $\text{Open}(i)$, then the simulator has to returns r_{i,b_i}, where is unknown when constructing reduction from mStDH . We will bound it later. Our strategy is to decouple $\mathbf{c}[i]$ with $H(b_i, R_{i,0}, R_{i,1}, R^x_{i,b_i})$ and then use the randomness $(1 - b_i, r_{i,1-b_i}, R_{i,b_i})$ to explain $\mathbf{c}[i]$ (and thus we do not need r_{i,b_i} and can construct reduction from mStDH).

Game G_6: The difference to G_5 is that when generating $\mathbf{c}[i]$, we choose random key pair (K_i, k_i) independent of $H(b_i, R_{i,0}, R_{i,1}, R^x_{i,b_i})$, and when $\mathcal{A}$ opens $\mathbf{c}[i]$, then we define $H(b_i, R_{i,0}, R_{i,1}, R^x_{i,b_i})$ as (K_i, k_i).

By abort condition in H, $H(b_i, R_{i,0}, R_{i,1}, R^x_{i,b_i})$ will not be defined before $\mathbf{c}[i]$ is opened, so this modification does not change $\mathcal{A}$'s view, we have

$$\left|\Pr\left[\mathsf{G}_5^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_6^{\mathcal{A}} \Rightarrow 1\right]\right| \leq \Pr\left[\text{Bad}_6\right], \Pr\left[\text{Bad}_5\right] = \Pr\left[\text{Bad}_6\right]$$

Game G_7: We modify the simulation of Open: When $\mathcal{A}$ opens $\mathbf{c}[i]$, we set $H(1 - b_i, R_{i,0}, R_{i,1}, R^x_{i,1-b_i}) := (K_i, k_i)$, but not $H(b_i, R_{i,0}, R_{i,1}, R^x_{i,b_i})$. Moreover, instead of returning $(b_i, r_{i,b_i}, R_{i,1-b_i})$, we return its complement, $(1 - b_i, r_{i,1-b_i}, R_{i,b_i})$.

We argue that if Bad_7 does not occur, then the view of $\mathcal{A}$ in G_7 is the same as in G_6. This is because G_7 does not abort means that $\mathcal{A}$ has queried neither $H(b_i, R_{i,0}, R_{i,1}, R^x_{i,b_i})$ for some $i \in [\mu]\backslash I$ nor $H(1-b_i, R_{i,0}, R_{i,1}, R^x_{i,1-b_i})$ for some $i \in [\mu]\backslash I$. Hence, $\mathcal{A}$ has no information about these two values, and, as a result, $\mathcal{A}$ cannot tell the change in Open. We have

$$\left|\Pr\left[\mathsf{G}_6^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_7^{\mathcal{A}} \Rightarrow 1\right]\right| \leq \Pr\left[\text{Bad}_7\right], \Pr\left[\text{Bad}_6\right] = \Pr\left[\text{Bad}_7\right]$$

To conclude our argument, we construct a reduction $\mathcal{B}_2$ against the mStDH assumption to bound $\Pr\left[\text{Bad}_7\right]$. $\mathcal{B}_2$ has a similar structure with $\mathcal{B}_1$ in Figure 7, except that now $\mathcal{B}_2$ embeds Y_i into R_{i,b_i} (by setting $R_{i,b_i} := Y_i$ for all $i \in [\mu]$). The construction of $\mathcal{B}_2$ is shown in Figure 8.

In $\mathcal{B}_2$'s construction, it does not have r_{i,b_i} and cannot compute $Z_{i,b_i} = R^x_{i,b_i}$. But it is not a problem, since $\mathcal{B}_3$ can program the random oracle H. More precisely, it leaves Z_{i,b_i} as unknown and choose a random pair (K_i, k_i) (cf. Item 10). Now if Bad_7 does not happen then the response of $H(b_i, R_{i,0}, R_{i,1}, Z_{i,b_i})$ is anyway random to $\mathcal{A}$ and it does not change its view. If Bad_7 happens, then $\mathcal{B}_2$ can find out $Z_{i,b_i} = g^{r_{i,b_i} \cdot x}$ by its DHP oracle and extract the solution to the mStDH problem. Thus, we have

$$\Pr\left[\text{Bad}_5\right] = \Pr\left[\text{Bad}_6\right] = \Pr\left[\text{Bad}_7\right] \leq \mathsf{Adv}_{\mathbb{G}}^{\mathsf{mStDH}}(\mathcal{B}_2)$$

```
$\mathcal{B}_2^{\text{DHP}_X}(X, Y_1, ..., Y_\mu)$
$Z^* := \perp$
$(\mathcal{M}_a, \mathsf{st}) \xleftarrow{\$} \mathcal{A}_0^{\text{Dec},H,h}(X)$
for $i \in [\mu]$
    $\mathbf{m}[i] := \mathsf{m}_i \xleftarrow{\$} \mathcal{M}_a$
    $b_i \xleftarrow{\$} \{0,1\}$
    $r_{i,1-b_i} \xleftarrow{\$} \mathbb{Z}_p$
    $R_{i,1-b_i} := g^{r_{i,1-b_i}}$
    $Z_{i,1-b_i} := X^{r_{i,1-b_i}}$
    $R_{i,b_i} := Y_i$
    $(K_i, k_i) \xleftarrow{\$} K \times \{0,1\}^l$
    $\mathsf{d}_i := \mathsf{m}_i \oplus K_i$
    $\mathcal{T}_i := h(k_i, R_{i,0}, R_{i,1}, \mathsf{d}_i)$
    $\mathbf{c}[i] := (R_{i,0}, R_{i,1}, \mathsf{d}_i, \mathcal{T}_i)$
$out \xleftarrow{\$} \mathcal{A}_1^{\text{Open},\text{Dec},H,h}(\mathsf{st}, \mathbf{c})$
if $Z^* = \perp$ : $Z^* \xleftarrow{\$} \mathbb{G}$
return $Z^*$

$\text{Open}(i)$
$I := I \cup \{i\}$
$\mathsf{H}_{\mathsf{val}}[1 - b_i, R_{i,0}, R_{i,1}, Z_{i,1-b_i}]$
    $:= (K_i, k_i)$
return $(\mathsf{m}_i, (1 - b_i, r_{i,1-b_i}, R_{i,b_i}))$
```

```
$H(b, R_0, R_1, Z)$
if $\exists i \in [\mu] \backslash I$ s.t.
    $(b, R_0, R_1, Z) = (1 - b_i, R_{i,0}, R_{i,1}, Z_{i,1-b_i})$
    $Z^* \xleftarrow{\$} \mathbb{G}$
    Aborts the simulation and returns $Z^*$
if $\exists i \in [\mu] \backslash I$ s.t.
    $(b, R_0, R_1) = (b_i, R_{i,0}, R_{i,1})$
    and $\text{DHP}_X(R_{i,b_i}, Z) = 1$
    $Z^* := Z$                 // records the solution
    Aborts the simulation and returns $Z^*$
if $\exists (K, k)$ s.t. $\mathsf{H}_{\mathsf{dec}}[b, R_0, R_1] = (K, k)$
    and $\text{DHP}_X(R_b, Z) = 1$
    $\mathsf{H}_{\mathsf{val}}[b, R_0, R_1, Z] := (K, k)$
    $\mathsf{H}_{\mathsf{dec}}[b, R_0, R_1] := \perp$
if $\exists (K, k)$ s.t. $\mathsf{H}_{\mathsf{val}}[b, R_0, R_1, Z] = (K, k)$
    or $\mathsf{H}_{\mathsf{inv}}[b, R_0, R_1, Z] = (K, k)$
    return $(K, k)$
else
    $(K, k) \xleftarrow{\$} \mathcal{M} \times \{0,1\}^l$
    if $\text{DHP}_X(R_b, Z) = 1$
        $\mathsf{H}_{\mathsf{val}}[b, R_0, R_1, Z] := (K, k)$
    else $\mathsf{H}_{\mathsf{inv}}[b, R_0, R_1, Z] := (K, k)$
    return $(K, k)$
```

Fig. 8. mStDH adversary $\mathcal{B}_2$ in bounding Bad_7. It simulates G_7 for $\mathcal{A}$. The simulation of Dec and h are the same as in Figure 6. If $\mathcal{A}$ queries H on $b_i, R_{i,0}, R_{i,1}, R_{i,1-b_i}$ for some $i \in [\mu] \backslash I$, $\mathcal{B}_2$ aborts the simulation and return a random solution.

Now all challenge ciphertexts are encrypted by random key (K_i, k_i). From G_8 we conclude the proof by undoing the other changes in a reverse order.

Game G_8: We undo the abort rules in the H oracle, and explain the randomness of $\mathbf{c}[i]$ using $(b_i, r_{i,b_i}, R_{i,1-b_i})$. That is, we withdraw the modifications made in G_7, G_5 and G_4. Since now the computation of (K_i, k_i) is independent of b_i and $1 - b_i$, we can construct reduction from mStDH as we did in G_4 and G_7. Roughly, if we want to embed the challenge into R_{i,b_i}, then we can specify the random bit of $\mathbf{c}[i]$ as $1 - b_i$ and explain the randomness of $\mathbf{c}[i]$ by reprogramming H, and so we do not need the exponent of R_{i,b_i}. We have

$$\left| \Pr[\mathsf{G}_7 \Rightarrow 1] - \Pr[\mathsf{G}_8 \Rightarrow 1] \right| \leq 4\mathsf{Adv}_{\mathbb{G}}^{\mathsf{mStDH}}(\mathcal{B})$$

Game G_9: We undo the modification made in G_2. We have

$$\Pr[\mathsf{G}_8 \Rightarrow 1] = \Pr[\mathsf{G}_9 \Rightarrow 1]$$

Now we can construct a SIM-SO-CCA simulator $\mathcal{S}$ that simulates G_9 for $\mathcal{A}$ and interacts with the IDEAL-SO-CCA game to conclude the proof. The construction of simulator is shown in Figure 9.

$\mathcal{S}$ samples d_i uniformly from $\mathcal{M}$ and computes K_i as $\mathsf{d}_i \oplus \mathsf{m}_i$ (when $\mathcal{A}$ opens $\mathbf{c}[i]$), which is equivalent to sampling K_i firstly and then computing $\mathsf{d}_i := K_i \oplus \mathsf{m}_i$.

```
$\mathcal{S}^{\text{Open}}$
$(X, x) \xleftarrow{\$} \mathsf{KG}$
$(\mathcal{M}_a, st) \xleftarrow{\$} \mathcal{A}_0^{\text{Dec},H,h}(X)$
Outputs $\mathcal{M}_a$ and receives $\mathbf{m}''$ $/\!\!/\ \mathcal{S}_0$
for $i \in [\mu]$
    $b_i \xleftarrow{\$} \{0,1\}$
    $r_{i,b_i} \xleftarrow{\$} \mathbb{Z}_p, R_{i,b_i} := g^{r_{i,b_i}}$
    $Z_{i,b_i} := X^{r_{i,b_i}}$
    $r_{i,1-b_i} \xleftarrow{\$} \mathbb{Z}_p, R_{i,1-b_i} := g^{r_{i,1-b_i}}$
    $Z_{i,1-b_i} := X^{r_{i,1-b_i}}$
    $(\mathsf{d}_i, k_i) \xleftarrow{\$} \mathcal{M} \times \{0,1\}^l$
    $\mathcal{T}_i := h(k_i, R_{i,0}, R_{i,1}, \mathsf{d}_i)$
    $\mathbf{c}[i] := (R_{i,0}, R_{i,1}, \mathsf{d}_i, \mathcal{T}_i)$
$out \xleftarrow{\$} \mathcal{A}_1^{\text{Open},\text{Dec},H,h}(\mathsf{st}, \mathbf{c})$
return $out$    $/\!\!/\ \mathcal{S}_1$

$\text{Open}(i)$
Queries its Open on $i$
Receives $\mathsf{m}_i$ and records
$\mathsf{H}[b_i, R_{i,0}, R_{i,1}, Z_{i,b_i}] := (\mathsf{m}_i \oplus \mathsf{d}_i, k_i)$
$\mathsf{rand} := (b_i, r_{i,b_i}, R_{i,1-b_i})$
return $(\mathsf{m}_i, \mathsf{rand})$
```

Fig. 9. SIM-SO-CCA simulator $\mathcal{S}$ that simulates G_9 to conclude the proof of Theorem 1. We ignore the simulation of H, h, and Dec which are the same as in G_9 in Figure 6.

Therefore, $\mathcal{S}$ perfectly simulates G_9. Note that at the start of the proof we assume that from G_0 to G_8, there is no collision among the outputs of random oracle h, the first parts of outputs of H (i.e., K), and the second parts of outputs of H (i.e., k). Here we need to add back this collision bound. That is,

$$\left|\Pr\left[\mathsf{G}_9^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{IDEAL\text{-}SO\text{-}CCA}_{\mathsf{PKE}_{\mathsf{StDH}}}^{\mathcal{S}} \Rightarrow 1\right]\right| \leq \frac{n_H^2}{|\mathcal{M}|} + \frac{n_H^2 + n_h^2}{2^l}$$

By combining all the probability bounds, we have

$$\left|\Pr\left[\mathsf{REAL\text{-}SO\text{-}CCA}_{\mathsf{PKE}_{\mathsf{StDH}}}^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{IDEAL\text{-}SO\text{-}CCA}_{\mathsf{PKE}_{\mathsf{StDH}}}^{\mathcal{S}} \Rightarrow 1\right]\right|$$
$$\leq 8\mathsf{Adv}_{\mathbb{G}}^{\mathsf{mStDH}}(\mathcal{B}) + \frac{2n_H^2}{|\mathcal{M}|} + \frac{2(n_H^2 + n_h^2)}{2^l},$$

as stated in Theorem 1.

3.2 Construction from the Twin Diffie-Hellman Assumption

Using the twinning technique from [10], we can remove the use of StDH assumption in $\mathsf{PKE}_{\mathsf{StDH}}$ and have a scheme based on the standard CDH assumption. Let $\mathbb{G}$ be a group with prime order p and generator g. Let $H : \{0,1\} \times \mathbb{G}^3 \to \mathcal{M} \times \{0,1\}^l, h : \mathbb{G}^2 \times \{0,1\}^l \to \{0,1\}^\ell$ be hash functions. We propose a PKE scheme $\mathsf{PKE}_{\mathsf{TDH}} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ (shown in Figure 10) based on TDH. The randomness space of $\mathsf{PKE}_{\mathsf{TDH}}$ is $\{0,1\} \times \mathbb{Z}_p \times \mathbb{G}$. By [10], the TDH problem is tightly equivalent to the CDH problem. We give the security theorem of $\mathsf{PKE}_{\mathsf{TDH}}$ in the full version [7]. The probability bounds are identical to Theorem 1.

```
KG
$x_0, x_1 \xleftarrow{\$} \mathbb{Z}_p$
$X_0 := g^{x_0}, X_1 := g^{x_1}$
$\mathsf{pk} := (X_0, X_1), \mathsf{sk} := (x_0, x_1)$
return $(\mathsf{pk}, \mathsf{sk})$

Enc(pk, m ∈ $\mathcal{M}$)
parse $(X_0, X_1) := \mathsf{pk}$
$b \xleftarrow{\$} \{0,1\}, r_b \xleftarrow{\$} \mathbb{Z}_p$
$R_b := g^{r_b}, R_{1-b} \xleftarrow{\$} \mathbb{G}$
$Z_{b,0} := X_0^{r_b}, Z_{b,1} := X_1^{r_b}$
$(K, k) := H(b, R_0, R_1, Z_{b,0}, Z_{b,1})$
$\mathsf{d} := K \oplus \mathsf{m}$
$\mathcal{T} := h(k, R_0, R_1, \mathsf{d})$
return $(R_0, R_1, \mathsf{d}, \mathcal{T})$

Dec(sk, $(R_0, R_1, \mathsf{d}, \mathcal{T})$)
parse $(x_0, x_1) := \mathsf{sk}$
$\mathsf{m} := \perp$
$Z_{0,0} := R_0^{x_0}, Z_{0,1} := R_0^{x_1}$
$Z_{1,0} := R_1^{x_0}, Z_{1,1} := R_1^{x_1}$
$(K_0, k_0) := H(0, R_0, R_1, Z_{0,0}, Z_{0,1})$
$(K_1, k_1) := H(1, R_0, R_1, Z_{1,0}, Z_{1,1})$
$\mathcal{T}_0 := h(k_0, R_0, R_1, \mathsf{d})$
$\mathcal{T}_1 := h(k_1, R_0, R_1, \mathsf{d})$
if $\mathcal{T}_0 = \mathcal{T}$ : $\mathsf{m} = \mathsf{d} \oplus K_0$
if $\mathcal{T}_1 = \mathcal{T}$ : $\mathsf{m} = \mathsf{d} \oplus K_1$
return m
```

Fig. 10. Our Direction Construction of SIM-SO-CCA secure PKE schemes from the TDH assumption, $\mathsf{PKE}_{\mathsf{TDH}} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$

3.3 Direct Construction from the Decisional Diffie-Hellman Assumption

Our third direct construction is based on THE DDH assumption. Let $\mathbb{G}$ be a group with prime order p and two generators g_0 and g_1. Let $H : \{0,1\} \times \mathbb{G}^3 \to \mathcal{M} \times \{0,1\}^l, h : \{0,1\}^l \times \mathbb{G}^2 \to \{0,1\}^\ell$ be hash functions. The PKE scheme $\mathsf{PKE}_{\mathsf{DDH}} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ is shown in Figure 11. The randomness space of $\mathsf{PKE}_{\mathsf{DDH}}$ is the set $\{0,1\} \times \mathbb{Z}_p \times \mathbb{G}^2$.

CORRECTNESS. Similar to $\mathsf{PKE}_{\mathsf{StDH}}$, the correctness of $\mathsf{PKE}_{\mathsf{DDH}}$ depends on the hash function h. The correctness error $\mathsf{Adv}^{\mathsf{COR}}_{\mathsf{PKE}_{\mathsf{StDH}}}(\mathcal{A})$ is bounded by the collision probability of h, namely, $\mathsf{Adv}^{\mathsf{COR}}_{\mathsf{PKE}_{\mathsf{DDH}}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{CR}}_{h}(\mathcal{A})$.

Theorem 2. $\mathsf{PKE}_{\mathsf{DDH}}$ *in Figure 11 is SIM-SO-CCA secure (Definition 9) if the* mDDH *problem is hard on* $\mathbb{G}$ *and* H *and* h *are modeled as random oracles. Concretely, for any SIM-SO-CCA adversary* $\mathcal{A}$ *and relation* Rel*, there exists a simulator* $\mathcal{S}$ *and a adversary* $\mathcal{B}$ *such that:*

$$\mathsf{Adv}^{\mathsf{SIM\text{-}SO\text{-}CCA}}_{\mathsf{PKE}_{\mathsf{DDH}}}(\mathcal{A}, \mathcal{S}, \mu, \mathsf{Rel}) \leq 10\mathsf{Adv}^{\mathsf{mDDH}}_{\mathsf{GGen}}(\mathcal{B}) + \frac{6\mu q_H}{p} + \frac{2n_H^2}{|\mathcal{M}|} + \frac{2(n_H^2 + n_h^2)}{2^l}$$

where q_H *and* n_{DEC} *are the numbers of* $\mathcal{A}$*'s queries to* H *and* DEC*, respectively, and* μ *is the number of challenge ciphertexts.* $n_H = \mu + q_H + 2n_{\mathrm{DEC}}$ *and* $n_h = \mu + q_H + 2n_{\mathrm{DEC}}$ *are the total numbers of queries to* H *and* h*, respectively.*

$\mathsf{PKE}_{\mathsf{DDH}}$ is based on the DDH-based non-committing KEM in [26], plus the double-randomness technique. The proof of Theorem 2 is similar to Theorem 1. In the reduction, we can embed the DDH challenge into one of $(R_{b,0}, R_{b,1})$ and $(R_{1-b,0}, R_{1-b,1})$, and then claim the ciphertext to another one. Since we always have the secret key (x_0, x_1) in reduction, the decryption oracle can be simulated in a straightforward way. We leave the proof in our full version paper [7].

```
KG
$(x_0, x_1) \xleftarrow{\$} \mathbb{Z}_p^2$
$\mathsf{pk} := g_0^{x_0} g_1^{x_1}$
$\mathsf{sk} := (x_0, x_1)$
return $(\mathsf{pk}, \mathsf{sk})$

Enc(pk, m ∈ M)
parse $(X_0, X_1) := \mathsf{pk}$
$b \xleftarrow{\$} \{0,1\}, r_b \xleftarrow{\$} \mathbb{Z}_p$
$R_{b,0} := g_0^{r_b}, R_{b,1} := g_1^{r_b}$
$R_{1-b,0} \xleftarrow{\$} \mathbb{G}, R_{1-b,1} \xleftarrow{\$} \mathbb{G}$
$Z_b := \mathsf{pk}^{r_b}$
$(K, k) := H(b, R_{0,0}, ..., R_{1,1}, Z_b)$
$\mathsf{d} := K \oplus \mathsf{m}$
$\mathcal{T} := h(k, R_{0,0}, ..., R_{1,1}, \mathsf{d})$
return $(R_{0,0}, ..., R_{1,1}, \mathsf{d}, \mathcal{T})$

Dec(sk, (R_{0,0}, R_{0,1}, R_{1,0}, R_{1,1}, d, T))
parse $(x_0, x_1) := \mathsf{sk}$
$\mathsf{m} := \perp$
$Z_0 := R_{0,0}^{x_0} R_{0,1}^{x_1}$
$Z_1 := R_{1,0}^{x_0} R_{1,1}^{x_1}$
$(K_0, k_0) := H(0, R_{0,0}, ..., R_{1,1}, Z_0)$
$(K_1, k_1) := H(1, R_{0,0}, ..., R_{1,1}, Z_1)$
$\mathcal{T}_0 := h(k_0, R_{0,0}, ..., R_{1,1}, \mathsf{d})$
$\mathcal{T}_1 := h(k_1, R_{0,0}, ..., R_{1,1}, \mathsf{d})$
if $\mathcal{T}_0 = \mathcal{T}$ : $\mathsf{m} = \mathsf{d} \oplus K_0$
if $\mathcal{T}_1 = \mathcal{T}$ : $\mathsf{m} = \mathsf{d} \oplus K_1$
return m
```

Fig. 11. SIM-SO-CCA secure PKE scheme $\mathtt{PKE}_{\mathsf{DDH}} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$

4 Generic Construction: From Lossy Encryption to SO-CCA PKE

In this section, we prove the tight SO security of Fujisaki-Okamoto's (FO) transformation [15] assuming that the underlying PKE is a lossy encryption [3]. More precisely, if the lossy encryption scheme has efficient opener (e.g., the one from [25]), then FO is SIM-SO-CCA-secure. If the lossy encryption does not have efficient opener (e.g., the one from hash proof systems [19,3]), then FO is IND-SO-CCA secure.

We recall the notion of lossy encryption and the FO transformation. Then we prove the tight SO security of FO's transformation in the random oracle model.

Definition 11 (Lossy Encryption [3]). *Let* $\mathsf{wPKE} := (\mathsf{wKG}, \mathsf{wEnc}, \mathsf{wDec})$ *be a PKE scheme with message space* $\mathcal{M}$ *and randomness space* $\mathcal{R}$. wPKE *is lossy if it has the following properties:*

- wPKE *is correct according to Definition 8.*
- *Key indistinguishability: We say* wPKE *has key indistinguishability if there is an algorithm* LKG *such that, for any adversary* $\mathcal{B}$*, the advantage function*

$$\mathsf{Adv}^{\mathsf{key\text{-}ind}}_{\mathsf{wPKE}}(\mathcal{B}) := |\Pr[\mathcal{B}(\mathsf{pk}) \Rightarrow 1] - \Pr[\mathcal{B}(\mathsf{pk}') \Rightarrow 1]|$$

  is negligible, where $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{wKG}$ *and* $(\mathsf{pk}', \mathsf{td}) \xleftarrow{\$} \mathsf{LKG}$.
- *Lossiness: Let* $(\mathsf{pk}', \mathsf{td}) \xleftarrow{\$} \mathsf{LKG}$ *and* m, m' *be arbitrary messages in* $\mathcal{M}'$*, the statistical distance between* $\mathsf{wEnc}(\mathsf{pk}', \mathsf{m})$ *and* $\mathsf{wEnc}(\mathsf{pk}', \mathsf{m}')$ *is negligible.*
- *Openability: Let* $(\mathsf{pk}', \mathsf{td}) \xleftarrow{\$} \mathsf{LKG}$, m *and* m' *be arbitrary messages, and* r *be arbitrary randomness. For ciphertext* $c := \mathsf{wEnc}(\mathsf{pk}', \mathsf{m}; r)$*, there exists an algorithm* open *such that* $\mathsf{open}(\mathsf{td}, \mathsf{pk}', c, r, \mathsf{m}')$ *outputs* r' *where* $c = \mathsf{wEnc}(\mathsf{pk}', \mathsf{m}'; r')$*. Here* open *can be inefficient.*

We extend the above lossiness definition to a multi-challenge setting. The multi-challenge lossiness is implied by the single-challenge one using hybrid argument. Since it is only a statistical property, the hybrid argument will not affect tightness of the computational advantage.

Definition 12 (Multi-Challenge Lossiness). *Let* $(\mathsf{pk}', \mathsf{td}) \xleftarrow{\$} \mathsf{LKG}$*,* μ *be the number of challenge, and* $r_1, r'_1, ...r_\mu, r'_\mu$ *be arbitrary messages in* $\mathcal{M}'$*. Multi-challenge Lossiness requires that statistical distance between* $\{\mathsf{wEnc}(\mathsf{pk}', r_i)\}_{i\in[\mu]}$ *and* $\{\mathsf{wEnc}(\mathsf{pk}', r'_i)\}_{i\in[\mu]}$ *is negligible. We write the distance as* $\varepsilon^{\mathsf{m\text{-}enc\text{-}los}}_{\mathsf{wPKE}}$*.*

We require γ-spreadness for our construction.

Definition 13 (γ-Spreadness). *Let* $\mathsf{wPKE} := (\mathsf{wKG}, \mathsf{wEnc}, \mathsf{wDec})$ *be a PKE scheme with message space* $\mathcal{M}$*, randomness space* $\mathcal{R}$*, and ciphertext space* $\mathcal{C}$*. We say* wPKE *is* γ*-spread if for every key pair* $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{wKG}$*, and every message* $\mathsf{m} \in \mathcal{M}$*,*

$$\max_{\mathsf{c}\in\mathcal{C}} \Pr_{r \xleftarrow{\$} \mathcal{R}} [\mathsf{c} = \mathsf{wEnc}(\mathsf{pk}, \mathsf{m}; r)] \leq 2^{-\gamma}.$$

4.1 Construction

Let $\mathsf{wPKE} := (\mathsf{wKG}, \mathsf{wEnc}, \mathsf{wDec})$ be a lossy encryption scheme with message space $\mathcal{M}'$ and randomness space $\mathcal{R}'$. Let $H : \mathcal{M}' \to \mathcal{M}$ and $G : \mathcal{M}' \times \mathcal{M} \to \mathcal{R}'$ be two hash functions. The FO transformation $\mathtt{FO} := (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ is defined in Figure 12. Here we use the one-time pad as the symmetric part to encrypt the message. The randomness space of $\mathtt{FO}$ is $\mathcal{R}'$.

```
KG
  $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{wKG}$
  return $(\mathsf{pk}, \mathsf{sk})$

Enc(pk, m)
  $r \leftarrow \mathcal{M}'$
  $K := H(r)$
  $\mathsf{d} := K \oplus \mathsf{m}$
  $R := G(r, \mathsf{d})$
  $e := \mathsf{wEnc}(\mathsf{pk}, r; R)$
  return $(e, \mathsf{d})$

Dec(sk, (e, d))
  $\mathsf{m}' := \perp$
  $r' := \mathsf{wDec}(\mathsf{sk}, e)$
  $R' := G(r', \mathsf{d}), K' := H(r')$
  if $e = \mathsf{wEnc}(\mathsf{pk}, r'; R')$
    $\mathsf{m}' := \mathsf{d} \oplus K'$
  return $\mathsf{m}'$
```

Fig. 12. Fujisaki-Okamoto's transformation FO with lossy encryption wPKE.

As shown in [24], if wPKE is $(1 - \delta)$-correct and G is modeled as a random oracle, then $\mathtt{FO}$ is $(1 - q_G\delta)$-correct where q_G is the number of queries to G.

Theorems 3 and 4 show the tight SIM-SO-CCA and IND-SO-CCA security of $\mathtt{FO}$, respectively. We only prove Theorem 3 in the main body and leave that of Theorem 4 in our full version paper [7], since both proofs are similar and the SIM-SO-CCA security is more common.

Theorem 3. $\mathtt{FO}$ *in Figure 12 is SIM-SO-CCA secure if* G *and* H *are modeled as random oracles, and* wPKE *is a lossy encryption with efficient openability and*

γ-spreadness. Concretely, for any SIM-SO-CCA adversary $\mathcal{A}$ and relation Rel*, there exists a simulator $\mathcal{S}$ and $\mathcal{B}$ such that:*

$$\mathsf{Adv}_{\mathtt{FO}}^{\mathsf{SIM\text{-}SO\text{-}CCA}}(\mathcal{A}, \mathcal{S}, \mu, \mathsf{Rel}) \leq \mathsf{Adv}_{\mathsf{wPKE}}^{\mathsf{key\text{-}ind}}(\mathcal{B}) + 2\varepsilon_{\mathsf{wPKE}}^{\mathsf{m\text{-}enc\text{-}los}} + \frac{\mu n_{\mathrm{DEC}}}{2^{\gamma}} + \frac{2n_H^2}{|\mathcal{M}|} + \frac{2n_G^2}{|\mathcal{R}'|} + \frac{4\mu^2 + 5\mu(q_G + q_H)}{|\mathcal{M}'|},$$

where q_H, q_G, and n_{DEC} are the numbers of $\mathcal{A}$'s queries to G, H, and Dec*, respectively, μ is the number of challenge ciphertexts, and $n_G = \mu + n_{\mathrm{DEC}} + q_H$ and $n_H = \mu + n_{\mathrm{DEC}} + q_G$ are the number of queries (including the simulator) to G and H, respectively.*

Theorem 4. $\mathtt{FO}$ *in Figure 12 is IND-SO-CCA secure (Definition 10) if G and H are modeled as random oracles, and* wPKE *is a lossy encryption and γ-spreadness. Concretely, for any IND-SO-CCA adversary $\mathcal{A}$, there exists $\mathcal{B}$ such that:*

$$\mathsf{Adv}_{\mathtt{FO}}^{\mathsf{IND\text{-}SO\text{-}CCA}}(\mathcal{A}, \mu) \leq 2(\mathsf{Adv}_{\mathsf{wPKE}}^{\mathsf{key\text{-}ind}}(\mathcal{B}) + 3\varepsilon_{\mathsf{wPKE}}^{\mathsf{m\text{-}enc\text{-}los}} + \frac{\mu n_{\mathrm{DEC}}}{2^{\gamma}}) + \frac{2n_H^2}{|\mathcal{M}|} + \frac{2n_G^2}{|\mathcal{R}'|} + \frac{6\mu^2 + 5\mu(q_G + q_H)}{|\mathcal{M}'|},$$

where q_H, q_G, and n_{DEC} are the numbers of $\mathcal{A}$'s queries to G, H, and Dec*, respectively, μ is the number of challenge ciphertexts, and $n_G = \mu + n_{\mathrm{DEC}} + q_H$ and $n_H = \mu + n_{\mathrm{DEC}} + q_G$ are the number of queries (including the simulator) to G and H, respectively.*

4.2 Proof of Theorem 3

We prove it by the game sequence as in Figure 13. G_0 is the original game except that we use lazy sampling to simulate ROs G and H. We assume that, from G_0 to G_9, there is no collision among r_i's and the outputs of H and G. Let n_G and n_H be the number of queries to G and H, respectively. By the security game in Figure 13, $n_G = \mu + n_{\mathrm{DEC}} + q_G$ and $n_H = \mu + n_{\mathrm{DEC}} + q_H$. We have

$$\left|\Pr\left[\mathsf{REAL\text{-}SO\text{-}CCA}_{\mathtt{FO}}^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_0^{\mathcal{A}} \Rightarrow 1\right]\right| \leq \frac{n_H^2}{|\mathcal{M}|} + \frac{\mu^2}{|\mathcal{M}'|} + \frac{n_G^2}{|\mathcal{R}'|}$$

Game G_1: We modify Dec. Instead of using sk to simulate Dec, we use the randomness recorded in G to decrypt given ciphertexts (see Items 40 to 42). This simulation method is exact the same as the one in the original FO transformation [15]. By the argument in [15], if wPKE is γ-spread, then we have

$$\left|\Pr\left[\mathsf{G}_0^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_1^{\mathcal{A}} \Rightarrow 1\right]\right| \leq \frac{\mu \cdot n_{\mathrm{DEC}}}{2^{\gamma}}$$

Game G_2: We switch the public key to lossy mode by $(\mathsf{pk}', \mathsf{td}) \xleftarrow{\$} \mathsf{LKG}$. Since in this game the decryption oracle are simulated without using sk, we can simulate G_2 with pk'. By the key indistinguishability of the lossy encryption,

$$\left|\Pr\left[\mathsf{G}_1^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_2^{\mathcal{A}} \Rightarrow 1\right]\right| \leq \mathsf{Adv}_{\mathsf{wPKE}}^{\mathsf{key\text{-}ind}}(\mathcal{B}_0)$$

```
Games $G_0$-$G_7$
$(pk, sk) \xleftarrow{\$} wKG$                                  // $G_0$-$G_1$
$(pk', td) \xleftarrow{\$} LKG$                                 // $G_2$-$G_7$
$(pk, sk) := (pk', td)$                                         // $G_2$-$G_7$
$(\mathcal{M}_a, st) \xleftarrow{\$} \mathcal{A}_0^{H,G}(pk)$
for $i \in [\mu]$
    $\mathbf{m}[i] := m_i \xleftarrow{\$} \mathcal{M}_a$
    $r_i \xleftarrow{\$} \mathcal{M}'$
    $r'_i \xleftarrow{\$} \mathcal{M}'$                          // $G_3$-$G_7$
    $K_i := H(r_i)$
    $K_i \xleftarrow{\$} \mathcal{M}$                            // $G_5$
    $d_i := m_i \oplus K_i$
    $d_i \xleftarrow{\$} \mathcal{M}$                            // $G_6$-$G_7$
    $K_i := d_i \oplus m_i$                                     // $G_6$-$G_7$
    $R_i := G(r_i, d_i)$
    $R_i \xleftarrow{\$} \mathcal{R}'$                           // $G_5$-$G_7$
    $e_i := wEnc(pk, r_i; R_i)$
    $\mathbf{c}[i] := (e_i, d_i)$
$out \xleftarrow{\$} \mathcal{A}_1^{\text{OPEN},H,G}(st, \mathbf{c})$
return $Rel(\mathcal{M}_a, \mathbf{m}, I, out)$

$H(r)$
if $\exists i \in [\mu] \backslash I$ s.t. $r = r'_i$           // $G_3$-$G_7$
    abort                                                       // $G_3$-$G_7$
if $\exists i \in [\mu] \backslash I$ s.t. $r = r_i$            // $G_4$-$G_7$
    abort                                                       // $G_4$-$G_7$
if $H[r] = \perp$
    $H[r] := K \xleftarrow{\$} \mathcal{M}$
return $H[r]$

OPEN($i$)
$G[r_i, d_i] := R_i$                                            // $G_5$-$G_7$
$H[r_i] := K_i$                                                 // $G_5$-$G_7$
$R'_i :=$ open$(sk, pk, e_i, R_i, r'_i)$                        // $G_7$
$G[r'_i, d_i] := R'_i$                                          // $G_7$
$H[r'_i] := K_i$                                                // $G_7$
$I := I \cup \{i\}$
return $(m_i, r_i)$

DEC($c$) // $c \notin \mathbf{c}$
parse $(e, d) := c$
$m' := \perp$
$r' := wDec(sk, e)$                                             // $G_0$
$R' := G(r', d), K' := H(r')$                                   // $G_0$
if $e = wEnc(pk, r'; R')$                                       // $G_0$
    $m' := d \oplus K'$                                         // $G_0$
if $\exists (r', R')$ s.t. $G[r', d] = R'$
    and $e = wPKE(pk, r'; R')$                                  // $G_1$-$G_7$
    $K' := H(r')$                                               // $G_1$-$G_7$
    $m' := d \oplus K'$                                         // $G_1$-$G_7$
return $m'$

$G(r, d)$
if $\exists i \in [\mu] \backslash I$ s.t. $r = r'_i$           // $G_3$-$G_7$
    abort                                                       // $G_3$-$G_7$
if $\exists i \in [\mu] \backslash I$ s.t. $r = r_i$            // $G_4$-$G_7$
    abort                                                       // $G_4$-$G_7$
if $G[r, d] = \perp$
    $G[r, d] := R \xleftarrow{\$} \mathcal{R}'$
return $G[r, d]$
```

Fig. 13. Games G_0-G_7 for proving Theorem 3.

Game G_3: This is a preparation step. We choose some internal randomness r'_i for the opening queries in the next games. We abort G_3 if $\mathcal{A}$ queries either H or G with r'_i before opening $\mathbf{c}[i]$. Since r'_i (for $i \in [\mu]$) are internal and never revealed to $\mathcal{A}$, the probability that $\mathcal{A}$ queries r'_i for some i is $\frac{q_H + q_G}{|\mathcal{M}'|}$. We also require all r'_i's are different. By the union bound and collision bound, we have

$$\left| \Pr\left[\mathsf{G}_2^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_3^{\mathcal{A}} \Rightarrow 1\right] \right| \leq \frac{\mu \cdot (q_H + q_G)}{|\mathcal{M}'|} + \frac{\mu^2}{|\mathcal{M}'|}$$

Game G_4: We further modify the abort rules in H and G. If $\mathcal{A}$ queries H or G with r_i and $\mathbf{c}[i]$ is unopened, then G_4 aborts. Let $\mathsf{QueryBad}_j$ be the event that such abort event occurs in G_j, i.e., $\mathcal{A}$ queries H (resp., G) on r_i (resp, (r_i, d_i)) where $\mathbf{c}[i]$ is unopened. Then we have

$$\left| \Pr\left[\mathsf{G}_3^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_4^{\mathcal{A}} \Rightarrow 1\right] \right| \leq \Pr\left[\mathsf{QueryBad}_4\right]$$

Here we cannot bound $\Pr[\mathsf{QueryBad}_4]$ directly yet, since all e_i are correlated to $H(r_i)$ and $G(r_i, \mathsf{d}_i)$. We will bound $\Pr[\mathsf{QueryBad}_4]$ later. Our strategy for that is to decouple e_i with $G(r_i, \mathsf{d}_i)$ and $H(r_i)$. In the end, $\mathcal{A}$ can query r_i for $i \in [\mu] \backslash I$ (i.e., $\mathbf{c}[i]$ is unopened) with negligible probability.

Game G_5: We modify the generation of R_i and K_i. In this game, R_i and K_i are chosen uniformly, instead of using H and G. Moreover, upon Open(i), we set $H(r_i) := K_i$ and $G(r_i, \mathsf{d}_i) := R_i$. By the abort rules in G and H, $\mathcal{A}$ can learn neither $H(r_i)$ nor $G(r_i, \mathsf{d}_i)$ before opening $\mathbf{c}[i]$. Thus, we have

$$\Pr\left[\mathsf{G}_4^{\mathcal{A}} \Rightarrow 1\right] = \Pr\left[\mathsf{G}_5^{\mathcal{A}} \Rightarrow 1\right], \ \Pr[\mathsf{QueryBad}_4] = \Pr[\mathsf{QueryBad}_5]$$

Game G_6: We further modify the computation of d_i and K_i. In this game, d_i are chosen uniformly at random, and K_i are computed as $K_i := \mathsf{d}_i \oplus \mathsf{m}_i$. In G_5 K_i is distributed uniformly at random. Hence, this modification is conceptual.

$$\Pr\left[\mathsf{G}_5^{\mathcal{A}} \Rightarrow 1\right] = \Pr\left[\mathsf{G}_6^{\mathcal{A}} \Rightarrow 1\right], \ \Pr[\mathsf{QueryBad}_5] = \Pr[\mathsf{QueryBad}_6]$$

Game G_7: Upon Open(i), we compute the opened randomness R'_i with respect to r'_i and e_i using the open algorithm (see Item 29), and then set $G(r'_i, \mathsf{d}_i) := R'_i$ and $H(r'_i) := K_i$. Looking ahead, this modification is necessary for the later modification that $\mathbf{c}[i] = (e_i, \mathsf{d}_i)$ can be claimed to r'_i. $\mathcal{A}$ detects this modification if it queries $H(r'_i)$ or $G(r'_i, \mathsf{d}_i)$. This modification does not affect the occurring probability of $\mathsf{QueryBad}_7$, since r'_i is perfectly hidden. Therefore,

$$\left|\Pr\left[\mathsf{G}_6^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_7^{\mathcal{A}} \Rightarrow 1\right]\right| \leq \frac{\mu(q_G + q_H)}{|\mathcal{M}'|}, \ \Pr[\mathsf{QueryBad}_6] = \Pr[\mathsf{QueryBad}_7]$$

In G_7, we have the following observation: Before $\mathcal{A}$ opens i, R_i are independent of r_i, r'_i, K_i, and d_i, so e_i can be viewed as a ciphertext that $e_i := \mathsf{wPKE}(\mathsf{pk}', r_i; R_i)$ where the randomness R_i is sampled independently and uniformly. Therefore, by the *lossiness* of pk', we can replace $\mathsf{wPKE}(\mathsf{pk}', r_i; R_i)$ as another ciphertext $\mathsf{wPKE}(\mathsf{pk}', r''_i; R''_i)$ where r''_i and R''_i are sampled independently and uniformly, and $\mathcal{A}$ cannot distinguish such replacement except with $\varepsilon_{\mathsf{wPKE}}^{\mathsf{m\text{-}enc\text{-}los}}$. We move the description of G_7-G_9 to Figure 14.

Game G_8: We modify the generation of ciphertext e_i and simulation of Open. In this game, e_i is an encryption of a randomly chosen r''_i with randomness R''_i (see Item 14) which are independent of $r_i, r'_i, R_i, \mathsf{d}_i$. When $\mathcal{A}$ opens $\mathbf{c}[i] = (e_i, \mathsf{d}_i)$, the game simulator reprograms H and G so that $\mathbf{c}[i]$ can be "explained" by message m_i and randomness r'_i (i.e., $\mathsf{Enc}(\mathsf{pk}, \mathsf{m}_i; r'_i) = \mathbf{c}[i]$), and returns (m_i, r'_i). By the lossiness of wPKE, the statistical distance between $\{\mathsf{wPKE}(\mathsf{pk}', r_i)\}_{i\in[\mu]}$ with $\{\mathsf{wPKE}(\mathsf{pk}', r''_i)\}_{i\in[\mu]}$ is $\varepsilon_{\mathsf{wPKE}}^{\mathsf{m\text{-}enc\text{-}los}}$. Hence, we have

$$\left|\Pr\left[\mathsf{G}_7^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_8^{\mathcal{A}} \Rightarrow 1\right]\right| \leq \varepsilon_{\mathsf{wPKE}}^{\mathsf{m\text{-}enc\text{-}los}}$$
$$|\Pr[\mathsf{QueryBad}_7] - \Pr[\mathsf{QueryBad}_8]| \leq \varepsilon_{\mathsf{wPKE}}^{\mathsf{m\text{-}enc\text{-}los}}$$

```
Games $\mathsf{G}_7$-$\mathsf{G}_9$
$(\mathsf{pk}', \mathsf{td}) \xleftarrow{\$} \mathsf{LKG}$
$(\mathsf{pk}, \mathsf{sk}) := (\mathsf{pk}', \mathsf{td})$
$(\mathcal{M}_a, st) \xleftarrow{\$} \mathcal{A}_0^{\textsc{Dec},H,G}(\mathsf{pk})$
for $i \in [\mu]$
    $\mathbf{m}[i] := \mathsf{m}_i \xleftarrow{\$} \mathcal{M}_a$
    $r_i \xleftarrow{\$} \mathcal{M}'$                          // $\mathsf{G}_7$-$\mathsf{G}_8$
    $r_i' \xleftarrow{\$} \mathcal{M}'$
    $\mathsf{d}_i \xleftarrow{\$} \mathcal{M}$
    $K_i := \mathsf{d}_i \oplus \mathsf{m}_i$
    $R_i \xleftarrow{\$} \mathcal{R}'$
    $e_i := \mathsf{wEnc}(\mathsf{pk}, r_i; R_i)$              // $\mathsf{G}_7$
    $r_i'' \xleftarrow{\$} \mathcal{M}'$                        // $\mathsf{G}_8$-$\mathsf{G}_9$
    $R_i'' \xleftarrow{\$} \mathcal{R}'$                        // $\mathsf{G}_8$-$\mathsf{G}_9$
    $e_i \xleftarrow{\$} \mathsf{wEnc}(\mathsf{pk}, r_i''; R_i'')$  // $\mathsf{G}_8$-$\mathsf{G}_9$
    $\mathbf{c}[i] := (e_i, \mathsf{d}_i)$
$out \xleftarrow{\$} \mathcal{A}_1^{\textsc{Open},\textsc{Dec},H,G}(st, \mathbf{c})$
return $\mathsf{Rel}(\mathcal{M}_a, \mathbf{m}, I, out)$

$H(r)$
if $\exists i \in [\mu] \backslash I$ s.t. $r = r_i'$            // $\mathsf{G}_7$-$\mathsf{G}_8$
    abort                                           // $\mathsf{G}_7$-$\mathsf{G}_8$
if $\exists i \in [\mu] \backslash I$ s.t. $r = r_i$             // $\mathsf{G}_7$-$\mathsf{G}_8$
    abort                                           // $\mathsf{G}_7$-$\mathsf{G}_8$
if $\mathsf{H}[r] = \bot$
    $\mathsf{H}[r] := K \xleftarrow{\$} \mathcal{M}$
return $\mathsf{H}[r]$

$\textsc{Open}(i)$
$R_i' := \mathsf{open}(\mathsf{sk}, \mathsf{pk}, e_i, R_i, r_i')$     // $\mathsf{G}_7$
$R_i' := \mathsf{open}(\mathsf{sk}, \mathsf{pk}, e_i, R_i'', r_i')$   // $\mathsf{G}_8$-$\mathsf{G}_9$
$\mathsf{G}[r_i', \mathsf{d}_i] := R_i'$
$\mathsf{H}[r_i'] := K_i$
$\mathsf{H}[r_i] := K_i$                                  // $\mathsf{G}_7$-$\mathsf{G}_8$
$\mathsf{G}[r_i, \mathsf{d}_i] := R_i$                           // $\mathsf{G}_7$-$\mathsf{G}_8$
$I := I \cup \{i\}$
return $(\mathsf{m}_i, r_i)$                             // $\mathsf{G}_7$
return $(\mathsf{m}_i, r_i')$                            // $\mathsf{G}_8$-$\mathsf{G}_9$

$\textsc{Dec}(c)$ // $c \notin \mathbf{c}$
parse $(e, \mathsf{d}) := c$
$\mathsf{m}' := \bot$
if $\exists (r', K')$ s.t. $\mathsf{G}[r', \mathsf{d}] = R'$
    and $e = \mathsf{wPKE}(\mathsf{pk}, r'; R')$
    $K' := H(r')$
    $\mathsf{m}' := \mathsf{d} \oplus K'$
return $\mathsf{m}'$

$G(r, \mathsf{d})$
if $\exists i \in [\mu] \backslash I$ s.t. $r = r_i'$            // $\mathsf{G}_7$-$\mathsf{G}_8$
    abort                                           // $\mathsf{G}_7$-$\mathsf{G}_8$
if $\exists i \in [\mu] \backslash I$ s.t. $r = r_i$             // $\mathsf{G}_7$-$\mathsf{G}_8$
    abort                                           // $\mathsf{G}_7$-$\mathsf{G}_8$
if $\mathsf{G}[r, \mathsf{d}] = \bot$ : $\mathsf{G}[r, \mathsf{d}] := R \xleftarrow{\$} \mathcal{R}'$
return $\mathsf{G}[r, \mathsf{d}]$
```

Fig. 14. Games G_7-G_9 for proving Theorem 3.

Now $\Pr[\mathsf{QueryBad}_8]$ can be bounded. Since r_i and r_i' are chosen uniformly and independent of $\mathbf{c}[i]$ (for $i \in [\mu]$), we have

$$\Pr[\mathsf{QueryBad}_8] \leq \frac{\mu(q_G + q_H)}{|\mathcal{M}'|}, \Pr[\mathsf{QueryBad}_7] \leq \varepsilon_{\mathsf{wPKE}}^{\mathsf{m\text{-}enc\text{-}los}} + \frac{\mu(q_G + q_H)}{|\mathcal{M}'|}$$

Since now r_i' are independent of e_i before opening, and r_i is redundant in the simulation, we withdraw all the abort events defined in H and G, and no longer reprogram $H(r_i)$ and $G(r_i, \mathsf{d}_i)$.

Game G_9: the aborts event defined in H and G are withdraw, and we no longer generate r_i and reprogram $H(r_i)$ and $G(r_i, \mathsf{d}_i)$ when $\mathbf{c}[i]$ is opened. Since in G_9, for $i \in [\mu]$, r_i are independent of $\mathbf{c}[i]$, and r_i' are independent of $\mathbf{c}[i]$ before opening, the probability that $\mathcal{A}$ can detect this modification is $\frac{2\mu(q_G+q_H)}{|\mathcal{M}'|}$. Note that we have assumed that there is no collision among $r_i's$. So, we have

$$\left|\Pr\left[\mathsf{G}_8^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_9^{\mathcal{A}} \Rightarrow 1\right]\right| \leq \frac{2\mu(q_G + q_H)}{|\mathcal{M}'|} + \frac{\mu^2}{|\mathcal{M}'|}$$

```
$\underline{\mathcal{S}^{\textsc{Open}'}}$
$(\mathsf{pk}', \mathsf{td}) \xleftarrow{\$} \mathsf{LKG}$
$(\mathsf{pk}, \mathsf{sk}) := (\mathsf{pk}', \mathsf{td})$
$(\mathcal{M}_a, st) \xleftarrow{\$} \mathcal{A}_0^{\textsc{Dec},H,G}(\mathsf{pk})$
Outputs $\mathcal{M}_a$ and receives $\mathbf{m}''$        $/\!\!/ \mathcal{S}_0$
for $i \in [\mu]$
    $\mathsf{d}_i \xleftarrow{\$} \mathcal{M}$
    $r_i'' \xleftarrow{\$} \mathcal{M}', R_i'' \xleftarrow{\$} \mathcal{R}'$
    $e_i \xleftarrow{\$} \mathsf{wEnc}(\mathsf{pk}, r_i'' l R_i'')$
    $\mathbf{c}[i] := (e_i, \mathsf{d}_i)$
$out \xleftarrow{\$} \mathcal{A}_1^{\textsc{Open},\textsc{Dec},H,G}(st, \mathbf{c})$
return $out$        $/\!\!/ \mathcal{S}_1$

$\underline{\textsc{Open}(i)}$
$r_i' \xleftarrow{\$} \mathcal{M}'$
Queries $\textsc{Open}'(i)$
Receives and records $\mathsf{m}_i$
$K_i := \mathsf{d}_i \oplus \mathsf{m}_i$
$R_i' := \mathsf{open}(\mathsf{sk}, \mathsf{pk}, e_i, R_i'', r_i')$
$\mathsf{G}[r_i', \mathsf{d}_i] := R_i'$
$\mathsf{H}[r_i'] := K_i$
return $(r_i', \mathsf{m}_i)$
```

Fig. 15. SIM-SO-CCA simulator $\mathcal{S}$ that simulates G_9 to conclude the proof of Theorem 3. Here we ignore the details about simulation of H, G, and Dec which are the same as in Figure 14.

Now we can construct a simulator $\mathcal{S}$ that interacts with the IDEAL-SO-CCA game and simulate G_9 for $\mathcal{A}$. The construction of $\mathcal{S}$ is shown in Figure 15. The main difference between G_9 and $\mathcal{S}$ is that r_i' is sampled uniformly and K_i is computed when $\mathcal{A}$ queries $\textsc{Open}(i)$, which is conceptual. We have assumed that all r_i''s and all K's are pair-wise distinct, and the outputs of ROs H and G are different. Hence, we have

$$\left| \Pr\left[\mathsf{G}_9^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{IDEAL\text{-}SO\text{-}CCA}_{\mathtt{FO}}^{\mathcal{S}} \Rightarrow 1\right] \right| \leq \frac{n_H^2}{|\mathcal{M}|} + \frac{\mu^2}{|\mathcal{M}'|} + \frac{n_G^2}{|\mathcal{R}'|}$$

Combining all the above difference, we conclude Theorem 3 as

$$\left| \Pr\left[\mathsf{REAL\text{-}SO\text{-}CCA}_{\mathtt{FO}}^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{IDEAL\text{-}SO\text{-}CCA}_{\mathtt{FO}}^{\mathcal{S}} \Rightarrow 1\right] \right|$$
$$\leq \mathsf{Adv}_{\mathsf{wPKE}}^{\mathsf{key\text{-}ind}}(\mathcal{B}) + 2\varepsilon_{\mathsf{wPKE}}^{\mathsf{m\text{-}enc\text{-}los}} + \frac{\mu n_{\textsc{Dec}}}{2^{\gamma}} + \frac{2n_H^2}{|\mathcal{M}|} + \frac{2n_G^2}{|\mathcal{R}'|} + \frac{4\mu^2 + 5\mu(q_G + q_H)}{|\mathcal{M}'|}$$

4.3 Instantiations from DDH

We instantiate FO using the DDH-based lossy encryption from Bellare et al. [3] and Hofheinz et al. [25]. We describe the one with [25] here, since it leads to an (almost) tightly SIM-SO-CCA secure PKE, which is the main focus of this paper. Due to space limitation, we leave the one with [3] in our full version paper [7].

An Instantiation with Hofheinz et al.'s Lossy Encryption [25]. We use Hofheinz et al.'s DDH-based lossy encryption to instantiate FO. Following the notation in [25], we use the matrix Diffie-Hellman notation [13] to describe this scheme. Let $\mathbb{G}$ be a group with prime order p and generator g. Let $\mathbf{A} := (a_{i,j})_{(i,j)\in[l]\times[k]}$ be a matrix in $\mathbb{Z}_p^{l\times k}$, then the group representation of $\mathbf{A}$, denoted as $[\mathbf{A}]$, is defined as $(g^{a_{i,j}})_{(i,j)\in[l]\times[k]}$. Given $\mathbf{r}$ and $[\mathbf{A}]$, one can efficiently compute $[\mathbf{Ar}]$ (if their sizes match). We refer [13] for more details .

Let N be a positive integer. Let $H : \{0,1\}^N \to \mathcal{M}$ and $G : \{0,1\}^N \times \mathcal{M} \to \mathbb{Z}_p^{N+1}$ be two hash functions. Let $h : \mathbb{G} \to \{0,1\}$ be a universal hash function. The instantiated PKE scheme $\mathtt{FO}_2$ is shown in Figure 16. Hofheinz et al.'s DDH-based lossy encryption has efficient opener, and it is $(\log(p))$-spread, thus by Theorem 3, $\mathtt{FO}_2$ has tight SIM-SO-CCA security.

```
$\mathsf{KG}_2^{\mathsf{fo}}$
$\mathbf{A}_0 \xleftarrow{\$} \mathbb{Z}_p^{1\times(N+1)}$
$\mathbf{T} \xleftarrow{\$} \mathbb{Z}_p^{N\times 1}$
$\mathbf{A}_1 := \mathbf{T}\mathbf{A}_0 \in \mathbb{Z}_p^{N\times(N+1)}$
$\mathsf{pk} := ([\mathbf{A}_0], [\mathbf{A}_1])$
$\mathsf{sk} := \mathbf{T}$
return $(\mathsf{pk}, \mathsf{sk})$
```

```
$\mathsf{Enc}_2^{\mathsf{fo}}(\mathsf{pk}, \mathsf{m})$
$s \leftarrow \{0,1\}^N$
$K := H(s)$
$\mathsf{d} := K \oplus \mathsf{m}$
$\mathbf{r} := G(s, \mathsf{d}) \in \mathbb{Z}_p^{N+1}$
$[\mathbf{R}_0] := [\mathbf{A}_0\mathbf{r}] \in \mathbb{G}$
$[\mathbf{Z}] := [\mathbf{A}_1\mathbf{r}] \in \mathbb{G}^N$
for $i \in [N]$
    $c_i := h([\mathbf{Z}]_i) \oplus s_i$
$c := c_0c_1...c_N$
return $(([\mathbf{R}_0], c), \mathsf{d})$
```

```
$\mathsf{Dec}_2^{\mathsf{fo}}(\mathsf{sk}, ([\mathbf{R}_0], c), \mathsf{d})$
$\mathsf{m}' := \perp$
$[\mathbf{Z}'] := [\mathbf{T}\mathbf{R}_0]$
$c_1c_2...c_N =: c$
for $i \in [N]$
    $s_i' := c_i \oplus h([\mathbf{Z}']_i)$
$s' := s_1's_2'...s_N'$
$K' := H(s'), \mathbf{r}' := G(s', \mathsf{d})$
if $[\mathbf{R}_0] = [\mathbf{A}_0\mathbf{r}']$
    $\mathsf{m}' := \mathsf{d} \oplus K'$
return $\mathsf{m}'$
```

Fig. 16. A DDH-based scheme $\mathtt{FO}_2$ with efficient opener.

Corollary 2. $\mathtt{FO}_2$ *in Figure 16 is SIM-SO-CCA secure (Definition 9) if the* DDH *problem is hard on* $\mathbb{G}$*. Concretely, for any SIM-SO-CCA adversary* $\mathcal{A}$ *and relation* Rel*, there exists a simulator* $\mathcal{S}$ *and* $\mathcal{B}$ *such that:*

$$\mathsf{Adv}_{\mathsf{FO}}^{\mathsf{SIM\text{-}SO\text{-}CCA}}(\mathcal{A}, \mathcal{S}, \mu, \mathsf{Rel}) \leq N \cdot \mathsf{Adv}_{\mathbb{G}}^{\mathsf{DDH}}(\mathcal{B}) + \frac{2\mu}{p} + \frac{\mu n_{\mathrm{DEC}}}{p} + \frac{2n_H^2}{|\mathcal{M}|} + \frac{2n_G^2}{p^{N+1}} + \frac{4\mu^2 + 5\mu(q_G + q_H)}{2^N},$$

where q_H, q_G*, and* n_{DEC} *are the numbers of* $\mathcal{A}$*'s queries to* G, H*, and* $\textsc{Dec}$*, respectively,* μ *is the number of challenge ciphertexts, and* $n_G = \mu + n_{\mathrm{DEC}} + q_H$ *and* $n_H = \mu + n_{\mathrm{DEC}} + q_G$ *are the number of queries (including the simulator) to* G *and* H*, respectively.*

Acknowledgments This work is supported by the Research Council of Norway under Project No. 324235. We thank the anonymous reviewers from Asiacrypt 2022 for referring us to the work of Bellare et al. [2] and encouraging us to discuss its impacts on previous work in the random oracle model and ours. Moreover, we thank Benedikt Wagner (CISPA, Germany) and one of our reviewers for pointing out a mistake in Game 5 of our previous proof for Theorem 1. Wagner provided very constructive suggestions on it, during his visit at NTNU.

References

1. Abdalla, M., Bellare, M., Rogaway, P.: The oracle Diffie-Hellman assumptions and an analysis of DHIES. In: Naccache, D. (ed.) CT-RSA 2001. LNCS, vol. 2020, pp. 143–158. Springer, Heidelberg (Apr 2001)

2. Bellare, M., Dowsley, R., Waters, B., Yilek, S.: Standard security does not imply security against selective-opening. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 645–662. Springer, Heidelberg (Apr 2012)
3. Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (Apr 2009)
4. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Denning, D.E., Pyle, R., Ganesan, R., Sandhu, R.S., Ashby, V. (eds.) ACM CCS 93. pp. 62–73. ACM Press (Nov 1993)
5. Bellare, M., Rogaway, P.: Optimal asymmetric encryption. In: Santis, A.D. (ed.) EUROCRYPT'94. LNCS, vol. 950, pp. 92–111. Springer, Heidelberg (May 1995)
6. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Heidelberg (May / Jun 2006)
7. Bernstein, D.J., Persichetti, E.: Towards kem unification. Cryptology ePrint Archive (2022)
8. Böhl, F., Hofheinz, D., Kraschewski, D.: On definitions of selective opening security. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 522–539. Springer, Heidelberg (May 2012)
9. Canetti, R., Fischlin, M.: Universally composable commitments. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 19–40. Springer, Heidelberg (Aug 2001)
10. Cash, D., Kiltz, E., Shoup, V.: The twin Diffie-Hellman problem and applications. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 127–145. Springer, Heidelberg (Apr 2008)
11. Chen, J., Wee, H.: Fully, (almost) tightly secure IBE and dual system groups. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 435–460. Springer, Heidelberg (Aug 2013)
12. Diemert, D., Gellert, K., Jager, T., Lyu, L.: More efficient digital signatures with tight multi-user security. In: Garay, J. (ed.) PKC 2021, Part II. LNCS, vol. 12711, pp. 1–31. Springer, Heidelberg (May 2021)
13. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.: An algebraic framework for Diffie-Hellman assumptions. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 129–147. Springer, Heidelberg (Aug 2013)
14. Fehr, S., Hofheinz, D., Kiltz, E., Wee, H.: Encryption schemes secure against chosen-ciphertext selective opening attacks. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 381–402. Springer, Heidelberg (May / Jun 2010)
15. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. Journal of Cryptology 26(1), 80–101 (Jan 2013)
16. Gay, R., Hofheinz, D., Kohl, L.: Kurosawa-desmedt meets tight security. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part III. LNCS, vol. 10403, pp. 133–160. Springer, Heidelberg (Aug 2017)
17. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 75–92. Springer, Heidelberg (Aug 2013)
18. Han, S., Liu, S., Lyu, L., Gu, D.: Tight leakage-resilient CCA-security from quasi-adaptive hash proof system. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part II. LNCS, vol. 11693, pp. 417–447. Springer, Heidelberg (Aug 2019)

19. Hemenway, B., Libert, B., Ostrovsky, R., Vergnaud, D.: Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 70–88. Springer, Heidelberg (Dec 2011)
20. Heuer, F., Jager, T., Kiltz, E., Schäge, S.: On the selective opening security of practical public-key encryption schemes. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 27–51. Springer, Heidelberg (Mar / Apr 2015)
21. Heuer, F., Jager, T., Kiltz, E., Schäge, S.: On the selective opening security of practical public-key encryption schemes. Cryptology ePrint Archive, Report 2016/342 (2016), https://eprint.iacr.org/2016/342
22. Heuer, F., Poettering, B.: Selective opening security from simulatable data encapsulation. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part II. LNCS, vol. 10032, pp. 248–277. Springer, Heidelberg (Dec 2016)
23. Hofheinz, D.: All-but-many lossy trapdoor functions. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 209–227. Springer, Heidelberg (Apr 2012)
24. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017, Part I. LNCS, vol. 10677, pp. 341–371. Springer, Heidelberg (Nov 2017)
25. Hofheinz, D., Jager, T., Rupp, A.: Public-key encryption with simulation-based selective-opening security and compact ciphertexts. In: Hirt, M., Smith, A.D. (eds.) TCC 2016-B, Part II. LNCS, vol. 9986, pp. 146–168. Springer, Heidelberg (Oct / Nov 2016)
26. Jager, T., Kiltz, E., Riepel, D., Schäge, S.: Tightly-secure authenticated key exchange, revisited. In: Canteaut, A., Standaert, F.X. (eds.) EUROCRYPT 2021, Part I. LNCS, vol. 12696, pp. 117–146. Springer, Heidelberg (Oct 2021)
27. Libert, B., Sakzad, A., Stehlé, D., Steinfeld, R.: All-but-many lossy trapdoor functions and selective opening chosen-ciphertext security from LWE. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part III. LNCS, vol. 10403, pp. 332–364. Springer, Heidelberg (Aug 2017)
28. Liu, S., Paterson, K.G.: Simulation-based selective opening CCA security for PKE from key encapsulation mechanisms. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 3–26. Springer, Heidelberg (Mar / Apr 2015)
29. Lyu, L., Liu, S., Han, S., Gu, D.: Tightly SIM-SO-CCA secure public key encryption from standard assumptions. In: Abdalla, M., Dahab, R. (eds.) PKC 2018, Part I. LNCS, vol. 10769, pp. 62–92. Springer, Heidelberg (Mar 2018)