Almost Tightly-Secure Re-Randomizable and Replayable CCA-secure Public Key Encryption

Antonio Faonio[1], Dennis Hofheinz[2], and Luigi Russo[1]

[1] EURECOM, Sophia Antipolis, France {faonio, russol}@eurecom.fr
[2] ETH Zurich, Switzerland hofheinz@inf.ethz.ch

Abstract. Re-randomizable Replayable CCA-secure public key encryption (Rand-RCCA PKE) schemes guarantee security against chosen-ciphertext attacks while ensuring the useful property of re-randomizable ciphertexts. We introduce the notion of multi-user and multi-ciphertext Rand-RCCA PKE and we give the first construction of such a PKE scheme with an almost tight security reduction to a standard assumption. Our construction is structure preserving and can be instantiated over Type-1 pairing groups. Technically, our work borrows ideas from the state-of-the-art Rand-RCCA PKE scheme of Faonio *et al.* (ASIACRYPT'19) and the adaptive partitioning technique of Hofheinz (EUROCRYPT'17). Additionally, we show (1) how to turn our scheme into a publicly verifiable (pv) Rand-RCCA scheme and (2) that plugging our pv-Rand-RCCA PKE scheme into the MixNet protocol of Faonio *et al.* we can obtain the first *almost tightly-secure* MixNet protocol.

1 Introduction

Security against chosen-ciphertext attacks (CCA) is considered to be the standard notion of security for PKE schemes. This security definition, formulated by Rackoff and Simon [33], is elegant and easy to understand, and it has shown, by any means, to withstand the test of time.

Replayable and Re-Randomizable CCA security. Canetti, Krawczyk and Nielsen [7] pointed out that CCA security is not necessary for implementing secure channels. They showed that "replayable chosen-ciphertext" (RCCA) security suffices for secure channels, and might in fact allow for more efficient instantiations. Subsequently, Groth [20] showed that RCCA PKE schemes (called Rand-RCCA secure) can have re-randomizable ciphertexts. Specifically, Groth constructed a scheme with a ciphertext re-randomization procedure that, given a ciphertext as input, produces a fresh and unlinkable ciphertext that decrypts to the same message. Such a re-randomization procedure opens the door for applications that require secure communication *and* anonymity. For instance, Rand-RCCA secure PKE schemes enable anonymous and secure message transmissions (see Prabhakaran and Rosulek [32]), Mix-Nets (see Faonio *et al.* [13] and Pereira and Rivest [31]), Controlled Functional Encryption (see Naveed *et al.* [30]), and one-round message-transmission protocols with reverse firewalls (see Dodis, Mironov and Stephens-Davidowitz [10]).

Tight Security. Yet another criticism of the original definition of CCA security is that while the definition postulates that the message underlying *one single* ciphertext remains protected even under CCA attacks, in the real world, a PKE scheme is used to protect a large number of ciphertexts from possibly many users. Now, it is well-known that security for one single ciphertext implies, through a hybrid argument, security for many ciphertexts and many users. However, it is unclear how much *concrete* security a PKE scheme really offers when it is used in the wild. This question, initially posed by Bellare, Boldyreva and Micali [4] created a fruitful area of research that investigates how tight the security of an encryption scheme translates to the trust that we have with respect to the cryptographic assumption that it relies on. In more detail, a tight security reduction ensures that for any attack on the PKE scheme, there exists an attack on the assumption that is similar both in terms of complexity (i.e. the running time, the space required, etc.) and success probability. Thus, in the setting of tight security reductions, the number of ciphertexts considered by the security definition matters.

By now, many CCA-PKE schemes have been proven to have tight security in the multi-ciphertext and multi-user setting: some notable examples are the works of [17,18,21,22,25,26]. However, tight security in the context of Rand-RCCA security has not been studied.

1.1 Our Contributions

We initiate the study of tight security for Rand-RCCA secure PKE schemes in the multi-ciphertext and multi-user setting. Our main contributions are a new security definition for RCCA security in multi-ciphertext and multi-user setting (hereafter, mRCCA security), and a Rand-mRCCA PKE scheme whose mRCCA security (almost[3]) tightly reduces to the $\mathcal{D}_d$-MDDH assumption in symmetric (a.k.a. type-1) pairing groups.

Moreover, as an application, we revise the protocol for universally composable MixNet based on Rand-RCCA PKE from [13]. In the following paragraphs, we elaborate more about each of the contributions.

Multi-user Multi-ciphertext RCCA security. In the security experiment of the (single-ciphertext) RCCA security notion, the decryption oracle, called "guarded decryption oracle", can be queried on any ciphertext, including the challenge ciphertext. However, when decryption leads to one of the challenge messages $(\mathtt{M}_0, \mathtt{M}_1)$, the oracle answers with a special symbol $\diamond$ (meaning "same"). As a warm-up, consider a trivial extension to the case of (single-user) multi-ciphertext RCCA security where the attacker is given:

- an encryption oracle that, on input a pair of messages $\mathtt{M}_0, \mathtt{M}_1$, returns some valid encryption of $\mathtt{M}_b$ where b is the challenge bit,

[3] As most of the tightly-secure schemes, the security reduction suffers from a small multiplicative loss that is however independent of the number of uses of the scheme.

– and a guarded decryption oracle that, on input a ciphertext $\mathtt{C}$, returns a message $\mathtt{M}$, or the special indexed symbol $\diamond_j$ if $\mathtt{C}$ corresponds to an encryption of a message that was given as input to the encryption oracle as j-th query.

We notice that this trivial extension of RCCA security to multiple ciphertexts is impossible to achieve. Namely, consider the following generic attacker $\mathcal{A}$ that makes three queries to the encryption oracle: (i) $\mathcal{A}$ sends $(\mathtt{M}_1, \mathtt{M}_2)$, and receives back $\mathtt{C}_A$; (ii) sends $(\mathtt{M}_2, \mathtt{M}_3)$, and receives back $\mathtt{C}_B$; (iii) sends $(\mathtt{M}_3, \mathtt{M}_1)$, and receives back $\mathtt{C}_C$. $\mathcal{A}$ now queries the decryption oracle with $\mathtt{C}_C$. If the bit b is 0, the decryption oracle returns $\diamond_2$; if b is 1, the decryption oracle returns $\diamond_1$.

Yet another natural extension of the single-ciphertext RCCA security notion to the multi-ciphertext setting is to consider a guarded decryption oracle that upon input a ciphertext $\mathtt{C}$ either returns a message or the special symbol $\diamond$, but without notifying the adversary of which index j triggered the special symbol. Even if this definition avoids the attack described above, it is not as convenient as we would like it to be. Roughly speaking, the guarded decryption oracle reveals to the adversary that the queried ciphertext is a replay attack, but it doesn't tell which ciphertext was replayed; therefore, the larger the number of challenge ciphertexts, the less informative the output of the guarded decryption oracle will be. In particular, this definition is not sufficient for our MixNet application.

"In medio stat virtus", as the saying goes: the definition we propose is weaker than the first attempted (yet impossible to achieve) definition, but stronger than the above-mentioned definition. To build some intuition, in an equivalent version of the single-ciphertext RCCA security definition, the guarded decryption oracle would output the minimal set of messages that the queried ciphertext could decrypt to and such that such set does not trivially break the RCCA security definition: namely, if the ciphertext is a replay attack then the oracle replies with the set of challenge messages $\{\mathtt{M}_0, \mathtt{M}_1\}$, otherwise with a message $\mathtt{M}' \notin \{\mathtt{M}_0, \mathtt{M}_1\}$. We take a similar approach in our (multi-user) multi-ciphertext RCCA definition. The guarded decryption oracle outputs the minimal set of messages that the ciphertext could decrypt to without trivially breaking security. This set of messages includes all the pairs of challenge messages for which at least one of them is equal to the decryption of the queried ciphertext. To support the claim that our definition is indeed the most natural extension of RCCA to the multi-ciphertext setting, we prove that the simulation-based notion for RCCA security from [7] is tightly implied by our mRCCA security notion.

A Tightly-Secure Rand-mRCCA PKE scheme. Our starting points are the recent work of Faonio *et al.* [13] (hereafter `FFHR19`), which is the state of art for Rand-RCCA PKE scheme, and the tightly-secure CCA PKE schemes based on the adaptive partitioning techniques of Hofheinz [22] and Gay *et al.* [19].

Very briefly, the main idea of our construction is to encrypt the message similarly to `FFHR19`, and append a non-interactive proof of consistency for (part of) the ciphertext; the latter proof needs to have a (weak) form of simulation soundness property that can be obtained information-theoretically. Namely, using the notation of [22], we append to the ciphertext a *benign proof* for the consistency of part of the ciphertext (which lies in a linear language) of a proof system that

is statistically sound even when the adversary has oracle access to simulated proofs for a larger language that includes the disjunction of two linear spaces.

Some technical details. To go from the rough idea described above to the actual scheme, we need to overcome two technical problems. The first problem is that our benign proof system needs to be re-randomizable (or, to better say, "malleable" as it needs to be able to re-randomize proofs of re-randomized statements), as we are aiming to construct a Rand-PKE scheme. We notice that none of the benign proof systems or affine notions we are aware of (such as [2,18,19,22]) are re-randomizable. To solve this problem, we introduce a new malleable proof system based on the work of Abdalla, Benhamouda and Pointcheval [1].

The second (and more challenging) technical problem is that we need to reconcile the adaptive partitioning technique with the Rand-RCCA technique of [13]. In particular, at the core of the adaptive partitioning technique there is a complex argument that shows that the decryption oracle can safely reject *ill-formed* ciphertexts even when the adversary can observe (many) ill-formed challenge ciphertexts. In some sense, these challenge ciphertexts are the only ill-formed ciphertexts that correctly decrypt, while all other ill-formed ciphertexts produced by the adversary do not. However, in our security proof the adversary can easily produce ill-formed ciphertexts that correctly decrypt, simply by re-randomizing challenge ciphertexts.

In more detail, the adaptive partitioning technique moves the challenge ciphertexts back and forth between two different linear spaces (different from the linear space of honestly-generated ciphertexts). In our proof, differently than in previous works, we need to carefully define the relationship between these different linear spaces. In particular, it is necessary to make sure that re-randomizations of the challenge ciphertexts still lie in the prescribed linear space (and thus can be identified by our technique when answering $\diamond$). More technically, a ciphertext for our scheme can be parsed as a vector $[\mathbf{x}]$ in the source group (the CPA part of the ciphertext) plus two zero-knowledge proofs of consistency. The vector $[\mathbf{x}]$ for a well-formed ciphertext lies in the affine space defined by the encrypted message and the span of a matrix $[\mathbf{D}^*]$ which is part of the public key. Re-randomization works by summing up a random vector from the span of $\mathbf{D}^*$ to $\mathbf{x}$ (and updating the proofs accordingly). To apply the adaptive partitioning techniques, we move the challenge ciphertexts back and forth from two well-crafted distinct superspaces of $\mathbf{D}^*$. Thanks to this choice, we can recognize the challenge ciphertexts after re-randomization by multiplying the decrypted ciphertext by a matrix orthogonal to $\mathbf{D}^*$: this operation could be roughly interpreted as an "extended decryption" of the ciphertexts (since $\mathbf{D}^*$ encodes partial information of the secret key), however, we are not only interested to identify the encrypted message but also to uniquely link the decrypted (possibly re-randomized) ciphertext with one of the challenge ciphertexts. Thus, like previous adaptive partitioning approaches, we separate the randomness space of the PKE scheme into an honest part (the span of $\mathbf{D}^*$) and a normally unused part (spanned by the vectors in the mentioned super spaces, independent of $\mathbf{D}^*$) that is also used to hide the messages. In our view, the main technical insight is

that the span of $\mathbf{D}^*$ is used for re-randomization, while the other space is kept fixed for the challenge ciphertexts. We highlight that in order for the aforementioned strategy to work smoothly, we preferred to follow a flavor of adaptive partitioning as in Gay *et al.* [19], where secret keys are randomized, instead of the original strategy of Hofheinz [22], where ciphertexts are randomized. Finally, the original adaptive partitioning strategy relies on the pairwise universality of a hash proof system [9] that guarantees simpler statements about linear languages. We adapt this proof system to re-randomizable statements by considering higher-dimensional languages and refining the "core lemma for Rand-RCCA" from [13]. We highlight that this lemma was designed for the single-ciphertext scenario, thus, some extra care is needed in our adaptive partitioning argument, more in detail, when defining the notion of *critical query*. In particular, a critical query is commonly defined as a decryption query for an ill-formed ciphertext that would decrypt without errors under one of the randomized secret keys; the usual goal is to show that an adversary cannot make such a query. In our case, we need to refine this notion by additionally specifying when (allegedly) re-randomizations of challenge ciphertexts are critical. Since each one of the challenge ciphertexts is an ill-formed ciphertext that decrypts correctly under one of the randomized keys, we cannot consider critical a re-randomization of such a challenge ciphertext when it decrypts correctly under the same randomized key. Thus, after having recognized a decryption query as a re-randomization, we make sure that this ciphertext is decrypted only using a specific (a univocally linked) secret key; on the other hand, other kinds of decryption queries can be safely decrypted with any of the secret keys. This rule allows eventually to use the lemma of [13] which provides security even given an interface for decryption of re-randomizations of one challenge ciphertext under one specific secret key.

Extensions and applications. Following the strategy of [13] we show that our Rand-mRCCA PKE can be used to instantiate a PKE with the nice property of publicly verifiable ciphertexts (pv-Rand-mRCCA PKE). We propose two pv-Rand-mRCCA PKE schemes: one based on the Matrix Diffie-Hellman Assumption (MDDH), and a second more efficient scheme based on a new MDDH-like assumption (see Section 1.2 for the details) which we prove secure in the generic group model.

As an application of our framework, we show that we can plug a pv-Rand-mRCCA scheme into the MixNet protocol of [13]. Instantiating such protocol with our schemes, we obtain an (almost) *"tightly-secure"* MixNet protocol: namely a protocol, the first of its kind, whose security guarantees depend linearly on the number of mixer parties but only logarithmically on the number of mixed messages. To compare with the state of the art for MixNet protocols, we notice that the Bayer and Groth [3] proof of shuffle is based on the Fiat-Shamir transform applied to a multi-round Sigma protocol, thus the security reduction degrades with the number of rounds of the underlying Sigma protocol, while the proof of shuffle in the pairing setting of Fauzi *et al.* [16] relies on new kinds of $\mathcal{D}_n$-KerMDH assumptions (proved to hold generically in the same paper) where n is the number of shuffled ciphertexts.

1.2 Related Work

Prabhakaran and Rosulek [32] introduced the first Rand-RCCA PKE in the standard model. Abstracting the scheme of [32], and solving a long-standing open problem, recently Wang *et al.* [35] introduced the first receiver-anonymous Rand-RCCA PKE. Faonio and Fiore [12] introduced a practical Rand-RCCA PKE in the random oracle model. Considering the state of the art on pairing-based Rand-RCCA PKE schemes, the most relevant works are the Rand-RCCA PKE scheme of Chase *et al.* [8], the recent works of Libert, Peters and Qian [27], and of Faonio *et al.* [13]. In Table 1 we offer a comparison, in terms of security properties and functionalities, of our schemes of Section 5, i.e. $\mathcal{PKE}_1, \mathcal{PKE}_2$ and $\mathcal{PKE}_3$, and the previous schemes. From a technical point of view, our schemes inherit from the scheme of [13], however, we notice that our schemes are instantiated on type-1 pairing group, while `FFHR19` is instantiated on type-3 pairing group (see the next section and [14] for more details). On the other hand, our schemes are the only ones that have (almost) tight-security reductions. In Table 2 we compare the most efficient Rand-RCCA PKE schemes with ours. In particular, we instantiate $\mathcal{PKE}_1$ and $\mathcal{PKE}_2$ under DLIN assumption for type-1 pairing group ($d = 2$ and, because of the security of the benign proof system, $n = 6$) while we instantiate $\mathcal{PKE}_3$ under $\mathcal{U}_{9,4}$-TMDDH assumption. We compare the number of operations required by the three algorithms (Enc, Rand and Dec) and the size of the ciphertext. In particular, we have considered the cost of exponentiations in the source and target groups, and the number of pairings. We give only a rough estimation of the costs of $\mathcal{PKE}_2$ and $\mathcal{PKE}_3$ to provide some intuition on the considerable efficiency gap between them: their cost is derived in terms of group elements and operations needed to instantiate the proof systems for $\mathcal{PKE}_2$ (resp. $\mathcal{PKE}_3$) under $\mathcal{D}_{6,2}$-MDDH (resp. $\mathcal{U}_{9,4}$-TMDDH) assumption from [11] and [13].

We note that $\mathcal{PKE}_2$ and $\mathcal{PKE}_3$ are far from being considered practical, while $\mathcal{PKE}_1$ is considerably less efficient than [13]. Indeed, our main goal is to prove feasibility. We view our work as a potential first towards a tightly secure practical solution. For instance, while the first tightly IND-CCA secure PKE schemes were highly impractical, state-of-the-art schemes (see [17,18]) have a realistic break-even point[4]. We hope for a similar development with Rand-RCCA PKE schemes.

Our benign proof system uses the "OR-Proof" technique from [1]. We notice that, in the context of tightly-secure reductions, the same technique from [1] has been used in [21] to instantiate their (Leakage-Resilient) Ardent Quasi-Adaptive Hash Proof System. We stress that in our work, in contrast with [21], the main reason to use the technique from [1] is because of its nice linear property that, in turn, allows for *malleable* proof system.

1.3 Open Problems

Our Rand-RCCA PKE schemes require type-1 pairing groups, which are less efficient than type-3. It is natural to ask whether we can instantiate our PKE

[4] For the same security parameter, the work of [17,18] outperforms state-of-the-art non-tightly secure schemes like Kurosawa-Desmedt [24] around 2^{30} ciphertexts.

PKE	Group Setting	Assumption	Struc. Pres.	Pub. Ver.	Tight
[8] CKLM12, [27] LPQ17	Type-3	SXDH	✓	✓	
[13] FFHR19	Type-3	$\mathcal{D}_{d+1,d}$-MDDH	✓*	✓	
$\mathcal{PKE}_1$	Type-1	$\mathcal{D}_{n,d}$-MDDH	✓*		✓
$\mathcal{PKE}_2$	Type-1	$\mathcal{D}_{n,d}$-MDDH	✓*	✓	✓
$\mathcal{PKE}_3$	Type-1	$\mathcal{U}_{n,d}$-TMDDH	✓*	✓	✓

Table 1. Comparison of the properties of a selection of Rand-RCCA-secure PKE schemes. The symbol * indicates that the structure-preserving property of the schemes is not strict since ciphertexts contain some elements in $\mathbb{G}_T$.

PKE	$\lvert\mathsf{C}\rvert$	Enc $\approx$ Rand	Dec
[13] FFHR19 (1)	$3\mathbb{G}_1+2\mathbb{G}_2+\mathbb{G}_T$	$4E_1+5E_2+2E_T+5\mathsf{P}$	$8E_1+4E_2+4\mathsf{P}$
$\mathcal{PKE}_1$	$7\mathbb{G}_1+2\mathbb{G}_T$	$14E_1+2E_T+14\mathsf{P}$	$48E_1+36E_T+49\mathsf{P}$
[27] LPQ17	$42\mathbb{G}_1+20\mathbb{G}_2$	$79E_1+64E_2$	$1E_1+142\mathsf{P}$
[13] FFHR19 (2)	$14\mathbb{G}_1+15\mathbb{G}_2+4\mathbb{G}_T$	$36E_1+45E_2+6E_T+5\mathsf{P}$	$2E_1+50\mathsf{P}$
$\mathcal{PKE}_2$	$380\mathbb{G}_1+330\mathbb{G}_T$	$\approx 180E1+110E_T+38\mathsf{P}$	$\approx 6E_1+400\mathsf{P}$
$\mathcal{PKE}_3$	$105\mathbb{G}_1+9\mathbb{G}_T$	$\approx 261E_1+9E_T+16\mathsf{P}$	$\approx 6E_1+11\mathsf{P}$

Table 2. Efficiency comparison among the best Rand-RCCA-secure PKE schemes. We denote as E_i the cost of 1 exponentiation in $\mathbb{G}_i$, P the cost of computing a bilinear pairing. In the third column, we consider the cost of Enc which is almost always comparable with the cost of Rand. The first two schemes are privately verifiable, while the last four are publicly verifiable. We consider the most efficient instantiations for $\mathcal{PKE}_1, \mathcal{PKE}_2$ (DLIN), for $\mathcal{PKE}_3$ ($\mathcal{U}_{9,4}$-TMDDH) and for [13] (SXDH).

schemes from type-3 pairings. Unfortunately, we do not know how to do so, because it is not clear how to reconcile the adaptive partitioning technique [22] with a Rand-RCCA construction in settings with type-3 pairings (such as the one from [13]). We elaborate more on the challenges to overcome for obtaining a type-3 instantiation in [14] and leave the construction of a tightly-secure type-3 Rand-RCCA PKE scheme as an interesting open problem.

Our approach is semi-generic, as we work with pairing-based cryptography. We leave as open problem to provide a generic framework to instantiate (almost) tightly-secure Rand-RCCA-secure PKE. Possible starting points are the HPS-based frameworks of [35] for Rand-RCCA schemes and [21] for tightly-secure (LR-)CCA-secure schemes. Recently, Faonio and Russo [15] improved over the mix-net protocol of [13], giving a more efficient instantiation based on non publicly-verifiable Rand-RCCA PKE schemes; however, their construction requires a leakage-resilient scheme. We leave as open problem the extension of our analysis to tightly-secure LR-RCCA PKE schemes to extend their approach.

2 Preliminaries

A function is negligible in λ if it vanishes faster than the inverse of any polynomial in λ. We write $f(\lambda) \in \mathtt{negl}(\lambda)$ when f is negligible in λ. For any bit string $\tau \in \{0,1\}^*$, we denote by $\tau[i]$ the i-th bit of τ and by $\tau_{|i}$ the bit string comprising the

first i bits of τ. A symmetric (type-1) bilinear group $\mathcal{G}$ is a tuple $(q, \mathbb{G}_1, \mathbb{G}_T, e, \mathcal{P}_1)$, where $\mathbb{G}_1$ and $\mathbb{G}_T$ are groups of prime order q, the element $\mathcal{P}_1$ is a generator of $\mathbb{G}_1$, $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ is an efficiently computable, non-degenerate bilinear map. Let GGen be a PPT algorithm which on input 1^λ, where λ is the security parameter, returns a description of a symmetric bilinear group $\mathcal{G}$. Elements in $\mathbb{G}_i$, are denoted in implicit notation as $[a]_i := a\mathcal{P}_i$, where $i \in \{1, T\}$ and $\mathcal{P}_T := e(\mathcal{P}_1, \mathcal{P}_1)$. Every element in $\mathbb{G}_i$ can be written as $[a]_i$ for some $a \in \mathbb{Z}_q$, but note that given $[a]_i$, $a \in \mathbb{Z}_q$ is in general hard to compute (discrete logarithm problem). Given $a, b \in \mathbb{Z}_q$ we distinguish between $[ab]_i$, namely the group element whose discrete logarithm base $\mathcal{P}_i$ is ab, and $[a]_i \cdot b$, namely the execution of the multiplication of $[a]_i$ and b, and $[a]_1 \cdot [b]_1 = [a \cdot b]_T$, namely the execution of a pairing between $[a]_1$ and $[b]_1$. Sometimes, to simplify the notation, we will write $[a]$ instead of $[a]_1$ for elements in the source group. Vectors and matrices are denoted in boldface. We extend the pairing operation to vectors and matrices as $e([\mathbf{A}]_1, [\mathbf{B}]_1) = [\mathbf{A}^\top \cdot \mathbf{B}]_T$ and $e([y]_1, [\mathbf{A}]_1) = [y \cdot \mathbf{A}]_T$. Let $span(\mathbf{A})$ denote the linear span of the columns of $\mathbf{A}$. $\mathcal{D}_{n,d}$ is a matrix distribution if outputs (in probabilistic polynomial time, with overwhelming probability) matrices in $\mathbb{Z}_q^{n \times d}$.

Definition 1 (Matrix Decisional Diffie-Hellman Assumption, [11]). *The $\mathcal{D}_{n,d}$-MDDH assumption holds if for all non-uniform PPT adversaries $\mathcal{A}$,*

$$|\Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{Aw}]_1) = 1] - \Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{z}]_1) = 1]| \in \mathtt{negl}(\lambda),$$

where the probability is taken over $\mathcal{G} = (q, \mathbb{G}_1, \mathbb{G}_T, e, \mathcal{P}_1) \leftarrow \mathsf{GGen}(1^\lambda)$, $\mathbf{A} \leftarrow \mathcal{D}_{n,d}, \mathbf{w} \leftarrow \mathbb{Z}_q^d, [\mathbf{z}]_1 \leftarrow \mathbb{G}_1^n$ and the coin tosses of adversary $\mathcal{A}$.

For $Q \in \mathbb{N}$, $\mathbf{W} \leftarrow\!\!\$\ \mathbb{Z}_q^{d \times Q}$ and $\mathbf{U} \leftarrow\!\!\$\ \mathbb{Z}_q^{n \times Q}$, the Q-fold $\mathcal{D}_{n,d}$-MDDH assumption states that distinguishing tuples of the form $([\mathbf{A}]_1, [\mathbf{AW}]_1)$ from $([\mathbf{A}]_1, [\mathbf{U}]_1)$ is hard. That is, a challenge for the Q-fold $\mathcal{D}_{n,d}$-MDDH assumption consists of Q independent challenges of the $\mathcal{D}_{n,d}$-MDDH Assumption (with the same $\mathbf{A}$ but different randomness $\mathbf{w}$). In [11] it is shown that the two problems are equivalent, where the reduction loses at most a factor $n - d$.

Tensor Product. Let $\mathbf{a} \in \mathbb{Z}_q^n$ and $\mathbf{b} \in \mathbb{Z}_q^{n'}$, we define $\mathbf{a} \otimes \mathbf{b} \in \mathbb{Z}_q^{nn'}$ to be the tensor product between the two vectors. We can show the following property:

$$(\mathbf{A} \cdot \mathbf{R}) \otimes (\mathbf{B} \cdot \mathbf{S}) = (\mathbf{A} \otimes \mathbf{B}) \cdot (\mathbf{R} \otimes \mathbf{S}) \tag{1}$$

Lemma for Rand-RCCA security. The main technical tool employed by [13], to which they refer as their "core lemma", roughly speaking says that, for any $\mathbf{u} \in \mathbb{Z}_q^{d+1}$, the projective hash function with hash key $\mathbf{f}, \mathbf{F}$ that maps $\mathbf{v}$ to $(\mathbf{f} + \mathbf{Fv})^\top \mathbf{u}$ is pair-wise independent with respect to the quotient set $\mathbb{Z}_q^{d+2}/span(\mathbf{E})$ when given as side information the matrix $\mathbf{FE}$ where $\mathbf{E} \in \mathbb{Z}_q^{d+2 \times d}$. We generalize their result to $\mathbf{u} \in \mathbb{Z}_q^n$ and $\mathbf{E} \in \mathbb{Z}_q^{n' \times d}$ for any $n > d$ and $n' > d + 1$. The proof of the lemma follows by reduction to the original lemma from [13] and it can be found in [14]. For the sake of clarity, in this paper we prefer to call this lemma the "Rand-RCCA lemma", rather than "core lemma" (for Rand-RCCA) as in [13], because the core technical parts of our work and theirs are different.

Lemma 1 (Rand-RCCA Lemma). *Let d be a positive integer. For any matrix $\mathbf{D} \in \mathbb{Z}_q^{n\times d}$, $\mathbf{E} \in \mathbb{Z}_q^{n'\times d}$ where $n > d$ and $n' > d+1$, and any (possibly unbounded) adversary A:*

$$\Pr\left[\begin{array}{c}\mathbf{u} \notin span(\mathbf{D})\\ (\mathbf{v}-\mathbf{v}^*) \notin span(\mathbf{E}) :\\ z = (\mathbf{f}+\mathbf{F}\mathbf{v})^\top \mathbf{u}\end{array}\ \begin{array}{c}\mathbf{f} \leftarrow\!\$\ \mathbb{Z}_q^n, \mathbf{F} \leftarrow\!\$\ \mathbb{Z}_q^{n\times n'},\\ (z,\mathbf{u},\mathbf{v}) \leftarrow\!\$\ \mathcal{A}^{\mathcal{O}\cdot}(\mathbf{D},\mathbf{E},\mathbf{f}^\top\mathbf{D},\mathbf{F}^\top\mathbf{D},\mathbf{F}\mathbf{E})\end{array}\right] \leq \frac{n\cdot n'}{q}.$$

where the adversary outputs a single query $\mathbf{v}^*$ to $\mathcal{O}$ that returns $\mathbf{f}+\mathbf{F}\cdot\mathbf{v}^*$.

3 Non-Interactive Proof Systems (NIPS)

Definition 2 (Proof system). *Let $\mathcal{L} = \{\mathcal{L}_{pars}\}$ be a family of languages with $\mathcal{L}_{pars} \subseteq \mathcal{X}_{pars}$, and with efficiently computable witness relation $\mathcal{R}$. A non-interactive proof system (NIPS) $\mathbf{PS} = (\mathsf{PGen}, \mathsf{PPrv}, \mathsf{PVer}, \mathsf{PSim})$ for $\mathcal{L}$ consists of the following PPT algorithms:*

- $\mathsf{PGen}(1^\lambda, pars)$ *outputs a proving key ppk, a verification key psk.*
- $\mathsf{PPrv}(ppk, x, w)$, *$x \in \mathcal{L}$ and $\mathcal{R}(x,w) = 1$, outputs a proof π.*
- $\mathsf{PVer}(psk, x, \pi)$, *$x \in \mathcal{X}$ and a proof π, outputs a verdict $b \in \{0,1\}$.*
- $\mathsf{PSim}(psk, x)$, *$x \in \mathcal{L}$, outputs a proof π.*

Completeness: *For all $pars$, all (ppk, psk) in the range of $\mathsf{PGen}(1^\lambda, pars)$, all $x \in \mathcal{L}$, and all w with $\mathcal{R}(pars, x, w) = 1$, we have $\mathsf{PVer}(psk, x, \mathsf{PPrv}(ppk, x, w)) = 1$.*

When $ppk \neq psk$ we say that the proof system is *designated verifier*. In the definition above we let the verification and proving key depend on the parameters of the relation, namely, the proof systems are *quasi-adaptive* as defined by Jutla and Roy [23]. All the NIPSs of this paper are *structure-preserving*: i.e., all the public interfaces are vectors in the source groups, all the private material is in $\mathbb{Z}_q$ and all the algorithms can be described with pairing-product equations; also, as in [13] the proof π could lie in the target group.

Benign Proof Systems. All relevant security properties of a benign NIDVPS are condensed in the following definitions, taken verbatim from [22].

Definition 3 (Benign proof system). *Let $\mathbf{PS}$ be an NIDVPS for $\mathcal{L}$ as in Definition 2, and let $\mathcal{L}^{\mathrm{sim}} = \{\mathcal{L}^{\mathrm{sim}}_{pars}\}$, $\mathcal{L}^{\mathrm{ver}} = \{\mathcal{L}^{\mathrm{ver}}_{pars}\}$, and $\mathcal{L}^{\mathrm{snd}} = \{\mathcal{L}^{\mathrm{snd}}_{pars}\}$ be families of languages. We say that $\mathbf{PS}$ is $(\mathcal{L}^{\mathrm{sim}}, \mathcal{L}^{\mathrm{ver}}, \mathcal{L}^{\mathrm{snd}})$-*benign *if the following properties hold:*

(Perfect) zero-knowledge. *For all $pars$, all (ppk, psk) that lie in the range of $\mathsf{PGen}(1^\lambda, pars)$, and all $x \in \mathcal{L}$ and w with $\mathcal{R}(pars, x, w) = 1$, we have that the distribution $\mathsf{PPrv}(ppk, x, w)$ is equivalent to $\mathsf{PSim}(psk, x)$.*

(Statistical) $(\mathcal{L}^{\mathrm{sim}}, \mathcal{L}^{\mathrm{ver}}, \mathcal{L}^{\mathrm{snd}})$-soundness. *Let $\mathbf{Exp}^{\mathbf{snd}}_{\mathcal{A},\mathbf{PS}}$ be the game played by $\mathcal{A}$ in Fig. 1. Let $\mathbf{Adv}^{\mathbf{snd}}_{\mathbf{PS},\mathcal{A}}(\lambda)$ be the probability that $\mathbf{Exp}^{\mathbf{snd}}_{\mathcal{A},\mathbf{PS}}(\lambda) = 1$. We require that for all (possibly unbounded) $\mathcal{A}$ that only make a polynomial number of oracle queries, $\mathbf{Adv}^{\mathbf{snd}}_{\mathbf{PS},\mathcal{A}}(\lambda)$ is negligible.*

Non-Interactive Zero-Knowledge Proof Systems. We adapt Definition 2 for the case of publicly verifiable proof systems by requiring the prover key and the verification key to be identical, and we refer to such key as the *common reference string*. (Nontrivial) proof systems with this syntax are commonly called zero-knowledge proof systems (NIZKs).

Notice that in the syntax of proof system we give in Definition 3 both the simulator PSim and the verifier PVer receive as input the verification key, while in the usual definition of NIZK the simulator receives a simulation trapdoor. This difference is only syntactical. We say that a NIZK **PS** for $\mathcal{L}$ is *adaptively sound* if it is statistically $(\emptyset, \mathcal{L}, \emptyset)$-sound according to Definition 3.

Definition 4. *Let* **PS** *be a NIPS for* $\mathcal{L}$ *as in Def. 2, we say that* **PS** *is* (ϵ, T)-composable zero-knowledge *if there exists a PPT algorithm* $\overline{\mathsf{PGen}}$ *such that:*

- *For all pars, the distributions induced by the first output of* $\mathsf{PGen}(1^\lambda, pars)$ *and* $\overline{\mathsf{PGen}}(1^\lambda, pars)$ *are* ϵ*-close for any adversary with running time* T*.*
- *For all pars, all* (ppk, psk) *that lie in the range of* $\overline{\mathsf{PGen}}(1^\lambda, pars)$*, and all* $x \in \mathcal{L}$ *and* w *with* $\mathcal{R}(pars, x, w) = 1$*, we have that the distribution* $\mathsf{PPrv}(ppk, x, w)$ *is equivalent to* $\mathsf{PSim}(psk, x)$*.*

Malleable NIPS. We use the definitional framework of Chase *et al.* [8] for malleable proof systems. For simplicity of the exposition we consider only the unary case for transformations (see the aforementioned paper for more details). Moreover, we adapt their definition to the quasi-adaptive setting by having a transformation that depends on the *pars*. Let $T = (T_{\mathsf{el}}, T_{\mathsf{wit}})$ be a pair of efficiently computable functions, that we refer to as a *transformation*.

Definition 5 (Admissible transformation). *We say that an efficient relation* $\mathcal{R}$ *is* closed *under a transformation* $T = (T_{\mathsf{el}}, T_{\mathsf{wit}})$ *if for any* $(pars, x, w) \in \mathcal{R}$ *the pair* $(pars, T_{\mathsf{el}}(pars, x), T_{\mathsf{wit}}(w)) \in \mathcal{R}$*. If* $\mathcal{R}$ *is closed under* T *then we say that* T *is* admissible *for* $\mathcal{R}$*. Let* $\mathcal{T}$ *be a set of transformations, if for every* $T \in \mathcal{T}$*,* T *is admissible for* $\mathcal{R}$*, then* $\mathcal{T}$ *is an* allowable *set of transformations.*

Definition 6 (Malleable NIPS). *Let* **PS** *be an NIPS for* $\mathcal{L}$ *as in Definition 2, and let* $\mathsf{PEvl}(ppk, x, \pi, T)$ *be a PPT algorithm that takes as inputs ppk, an instance* x*, a proof* π*, and a transformation* $T \in \mathcal{T}$*, and it outputs a proof* π'*. We say that* **PS** *and* PEvl *form a* malleable proof system *for* $\mathcal{L}$ *with set* $\mathcal{T}$ *of allowable transformations for* $\mathcal{R}$*, if, for all pars,* (ppk, psk) *that lie in the range of* $\mathsf{PGen}(1^\lambda, pars)$*, all* $T \in \mathcal{T}$*, and all* x, π *we have* $\mathsf{PVer}(psk, T_{\mathsf{el}}(pars, x), \pi') = 1$ *if and only if* $\mathsf{PVer}(psk, x, \pi) = 1$*.*

Definition 7 (Derivation Privacy). *Let* **PS** *be a malleable NIPS for* $\mathcal{L}$ *with relation* $\mathcal{R}$ *and an allowable set of transformations* $\mathcal{T}$ *and corresponding* PEvl*. We say that* **PS** *is* derivation private *if for any PPT adversary* $\mathcal{A}$*:*

$$\mathbf{Adv}^{\texttt{der-priv}}_{\mathcal{A},\mathbf{PS}}(\lambda) := \left| \Pr\left[\mathbf{Exp}^{\texttt{der-priv}}_{\mathcal{A},\mathbf{PS}}(\lambda) = 1 \right] - \tfrac{1}{2} \right| \in \texttt{negl}(\lambda)$$

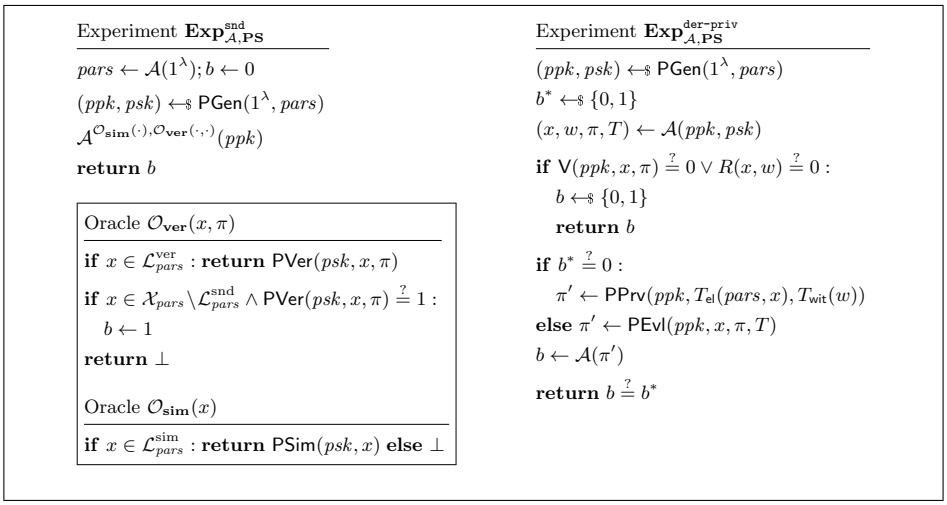

Fig. 1: Security experiments for benign soundness and derivation privacy of NIPS.

where $\mathbf{Exp}^{\mathtt{der\text{-}priv}}$ *is the game described in Fig. 1. Moreover we say that* **PS** *is* perfectly *(resp.* statistically*)* derivation private *when for any (possibly unbounded) adversary the advantage above is 0 (resp. negligible).*

Similarly to [13], we require a technical property to show re-randomizability of our encryption scheme that we call *tightness for proofs*, which roughly speaking says that it is hard to find a proof for a valid instance that does not lie in the set of the proofs created by the prover. For space reasons, we give more details in [14].

3.1 Our Malleable NIDVPS based on type-1 pairing

Let $\mathbf{D} \in \mathbb{Z}_q^{n\times d}$. We show that the following **PS** is a NIPS for $\mathcal{L} = span([\mathbf{D}]_1)$:

- $\mathsf{PGen}(pars)$ parses $pars$ as $\mathtt{prm}_G, [\mathbf{D}]_1 \in \mathbb{G}_1^{n\times d}$ where $n, d \in \mathbb{N}$, samples $\mathbf{k} \leftarrow\!\!\$\ \mathbb{Z}_q^{n^2}$, let $\mathbf{I}_n$ be the identity matrix of dimension n, set:

$$psk \leftarrow \mathbf{k} \text{ and } ppk \leftarrow (\mathbf{k}^\top[\mathbf{D}\otimes\mathbf{I}_n]_1, \mathbf{k}^\top[\mathbf{I}_n\otimes\mathbf{D}]_1, \mathbf{k}^\top[\mathbf{D}\otimes\mathbf{D}]_T)$$

- $\mathsf{PPrv}(ppk, [\mathbf{u}]_1, \mathbf{r})$ computes $\pi \leftarrow \mathbf{k}^\top[\mathbf{D}\otimes\mathbf{D}]_T \cdot (\mathbf{r}\otimes\mathbf{r})$ for $[\mathbf{u}]_1 = [\mathbf{D}]_1\mathbf{r}$
- $\mathsf{PSim}(psk, [\mathbf{u}]_1)$ computes $\pi \leftarrow \mathbf{k}^\top([\mathbf{u}]_1 \otimes [\mathbf{u}]_1)$
- $\mathsf{PVer}(psk, [\mathbf{u}]_1, \pi)$ returns 1 if and only if $\mathbf{k}^\top([\mathbf{u}]_1 \otimes [\mathbf{u}]_1) \stackrel{?}{=} \pi$

The first two vectors in the ppk are necessary to enable for the malleability of the proof system. While the third element of the public key could be efficiently derived from the previous two, we decide to publish it to speed up re-randomization and proving time. Consider the set $\mathcal{T}$ of admissible transformations for $\mathbb{Z}_q^n$:

$$\mathcal{T} = \{T : T_{\mathsf{el}}(pars, [\mathbf{u}]_1) = [\mathbf{u}]_1 + [\mathbf{D}]_1\hat{\mathbf{r}};\ T_{\mathsf{wit}}(\mathbf{r}) = \mathbf{r} + \hat{\mathbf{r}}\} \tag{2}$$

We note that any transformation T in the set above is uniquely determined by the vector $\hat{\mathbf{r}}$, thus, whenever it is clear from the context, we will simply use $\hat{\mathbf{r}}$ to identify the transformation. Let $\mathsf{PEvl}(ppk, \hat{\mathbf{r}}, [\mathbf{u}]_1, \pi)$ the algorithm that computes

$$\hat{\pi} \leftarrow \pi + \mathbf{k}^\top [\mathbf{I}_n \otimes \mathbf{D}]_1 \cdot [\mathbf{u} \otimes \hat{\mathbf{r}}]_1 + \mathbf{k}^\top [\mathbf{D} \otimes \mathbf{I}_n]_1 \cdot [\hat{\mathbf{r}} \otimes \mathbf{u}]_1 + \mathbf{k}^\top [\mathbf{D} \otimes \mathbf{D}]_T \cdot \hat{\mathbf{r}} \otimes \hat{\mathbf{r}}.$$

We show that **PS** and PEvl form a malleable proof system for the set of transformation $\mathcal{T}$ and the language $\mathcal{L}$.

Theorem 1. *Let $\mathcal{L} = span([\mathbf{D}]_1)$ and let $\mathcal{L}^{\text{snd}} = \mathcal{L}^{\text{sim}} = \{[\mathbf{u}]_1 : [\mathbf{u}]_1 = [\mathbf{D}_0]_1\mathbf{r} \vee [\mathbf{u}]_1 = [\mathbf{D}_1]_1\mathbf{r}\}$, and $\mathcal{L}^{\text{ver}} = \mathbb{Z}_q^n$, where $\mathbf{D}_i = \mathbf{D} \| \bar{\mathbf{D}}_i$ for $i \in \{0,1\}$, $\mathbf{D} \in \mathbb{Z}_q^{n \times d}$ and $\bar{\mathbf{D}}_0, \bar{\mathbf{D}}_1 \in \mathbb{Z}_q^{n \times d'}$. **PS** is a $(\mathcal{L}^{\text{sim}}, \mathcal{L}^{\text{ver}}, \mathcal{L}^{\text{snd}})$-benign proof system for $\mathcal{L}$ as long as $n^2 > 2n \cdot d + 2{d'}^2$, moreover, **PS** and PEvl form a malleable proof system for $\mathcal{L}$ and the set of transformation $\mathcal{T}$ defined in Eq. (2).*

Proof. In what follows, we prove each of the properties.

Completeness and Malleability. Our benign proof system is complete, as by Eq. (1) for any $\mathbf{u} = \mathbf{D}\mathbf{r}$ we have $(\mathbf{u} \otimes \mathbf{u}) = (\mathbf{D} \otimes \mathbf{D}) \cdot (\mathbf{r} \otimes \mathbf{r})$. We prove that our scheme is *malleable* (Definition 6) with respect to set of transformation $\mathcal{T}$ defined in Eq. (2), i.e., we prove that for any $[\mathbf{u}]$ and any $\hat{\mathbf{r}}$, a proof π for $[\mathbf{u}]$ verifies if and only if the proof $\hat{\pi}$ obtained executing PEvl on π and the transformation $\hat{\mathbf{r}}$ verifies for $[\mathbf{u} + \mathbf{D}\hat{\mathbf{r}}]$. For the first direction of the implication:

$$\begin{aligned}
\hat{\pi} &= \pi + \mathbf{k}^\top (\mathbf{I}_n \otimes \mathbf{D}) \cdot (\mathbf{u} \otimes \hat{\mathbf{r}}) + \mathbf{k}^\top (\mathbf{D} \otimes \mathbf{I}_n) \cdot (\hat{\mathbf{r}} \otimes \mathbf{u}) + \mathbf{k}^\top (\mathbf{D} \otimes \mathbf{D}) \cdot (\hat{\mathbf{r}} \otimes \hat{\mathbf{r}}) \\
&= \mathbf{k}^\top (\mathbf{u} \otimes \mathbf{u}) + \mathbf{k}^\top ((\mathbf{I}_n \mathbf{u}) \otimes (\mathbf{D}\hat{\mathbf{r}})) + \mathbf{k}^\top ((\mathbf{D}\hat{\mathbf{r}}) \otimes (\mathbf{I}_n \mathbf{u})) + \mathbf{k}^\top ((\mathbf{D}\hat{\mathbf{r}}) \otimes (\mathbf{D}\hat{\mathbf{r}})) \\
&= \mathbf{k}^\top (\mathbf{u} \otimes \mathbf{u} + \mathbf{u} \otimes (\mathbf{D}\hat{\mathbf{r}}) + (\mathbf{D}\hat{\mathbf{r}}) \otimes \mathbf{u} + (\mathbf{D}\hat{\mathbf{r}}) \otimes (\mathbf{D}\hat{\mathbf{r}})) \\
&= \mathbf{k}^\top ((\mathbf{u} + \mathbf{D}\hat{\mathbf{r}}) \otimes (\mathbf{u} + \mathbf{D}\hat{\mathbf{r}}))
\end{aligned}$$

We highlight that the second equation holds because of the definition of π and (1), while the third equation is obtained by grouping the previous line by $\mathbf{k}^\top$. The sequence of equations above also proves the other direction; indeed, if $\pi \neq \mathbf{k}^\top \mathbf{u} \otimes \mathbf{u}$, then $\hat{\pi} \neq \mathbf{k}^\top (\mathbf{u} + \mathbf{D}\hat{\mathbf{r}}) \otimes (\mathbf{u} + \mathbf{D}\hat{\mathbf{r}})$.

Soundness. We recall that $\mathbf{D} \in \mathbb{Z}_q^{n \times d}, \bar{\mathbf{D}}_i \in \mathbb{Z}_q^{n \times d'}$. If we only consider the view of the adversary given the verification key and the outputs of the simulation oracle we have that the proving key is uniformly distributed over a set of cardinality $q^{n^2 - 2nd - 2{d'}^2}$. Therefore, we require that $n^2 > 2n \cdot d + 2{d'}^2$ holds.

To see this, think of $\mathbf{k}$ as a formal variable and notice that publishing $\mathbf{k}^\top (\mathbf{D} \otimes \mathbf{I}_n)$ counts for $n \cdot d$ equations; also, $\mathbf{k}^\top (\mathbf{I}_n \otimes \mathbf{D})$ counts for $n \cdot d$ equations which in total gives us $2n \cdot d$ equations. Moreover, in order to simulate proofs for $[\mathbf{u}]_1 \in span([\mathbf{D}_i])$ the oracle gives away, at the worst case, the equations $\mathbf{k}^\top \left(\bar{\mathbf{D}}_i \otimes \bar{\mathbf{D}}_i\right)$ which count for ${d'}^2$ equations for each $i \in \{0,1\}$ which sum up to $2{d'}^2$ equations in total. Indeed, expanding $\mathbf{k}^\top (\mathbf{D}_i \otimes \mathbf{D}_i)$, we obtain $\mathbf{k}^\top \left(\mathbf{D} \otimes \mathbf{D} | \bar{\mathbf{D}}_i \otimes \mathbf{D} | \mathbf{D} \otimes \bar{\mathbf{D}}_i | \bar{\mathbf{D}}_i \otimes \bar{\mathbf{D}}_i\right)$. Now $\mathbf{k}^\top \left(\bar{\mathbf{D}}_i \otimes \mathbf{D}\right)$ and $\mathbf{k}^\top \left(\mathbf{D} \otimes \bar{\mathbf{D}}_i\right)$ can be computed given the proving key and $\mathbf{D}_0, \mathbf{D}_1$. In fact, when we compute

$\mathbf{k}^\top (\mathbf{D} \otimes \mathbf{I}) \left(\mathbf{I} \otimes \bar{\mathbf{D}}_i\right)$, we obtain $\mathbf{k}^\top \left(\mathbf{DI} \otimes \mathbf{I}\bar{\mathbf{D}}_i\right) = \mathbf{k}^\top \left(\mathbf{D} \otimes \bar{\mathbf{D}}_i\right)$. And in a similar way, we can compute $\mathbf{k}^\top \left(\bar{\mathbf{D}}_i \otimes \mathbf{D}\right)$. In total, we are giving up $2n \cdot d + 2{d'}^2$ equations and the length of our key k is n^2.

Notice that the adversary can gather additional information about the proving key $\mathbf{k}$ through the verification oracle. Indeed, whenever it sends a query $([\mathbf{u}]_1, \pi)$ with $[\mathbf{u}]_1 \in \mathcal{L}^{\text{ver}} \setminus \mathcal{L}^{\text{snd}}$ either it wins the security game or the adversary learns that $\pi \neq \mathbf{k}^\top [\mathbf{u}]_1 \otimes [\mathbf{u}]_1$.

Consider the hybrid experiment $\mathbf{H}_j$ where the first j-th queries $([\mathbf{u}]_1, \pi)$ to the verification oracle with $[\mathbf{u}]_1 \notin \mathcal{L}^{\text{snd}}$ are answered with 0, in particular, the bit b is left unmodified, while the remaining queries are handled as in the soundness experiment. Clearly, $\mathbf{H}_0$ is the original experiment, while $\mathbf{H}_Q$ where Q is an upper bound on the number of verification oracle queries made by the adversary is a trivial experiment where the adversary cannot win (since the bit b will never be set to 1), thus $\Pr[\mathbf{H}_Q = 1] = 0$. The distinguishing event between two consecutive hybrids is the event that the adversary wins the soundness experiment at the j-th query, which happens with probability $1/q^{n^2-2nd+2{d'}^2} \leq 1/q$, as it is the same as the event of guessing a uniformly random vector from a subspace of dimension $n^2-2nd+2{d'}^2$ of $\mathbb{Z}_q^{n^2}$, thus $\Pr[\mathbf{H}_j = 1] \leq \Pr[\mathbf{H}_{j+1} = 1] + 1/q$. Finally, by the triangular equation and noticing that Q is polynomial in the security parameter we can conclude our proof of soundness.

Derivation Privacy and Zero-Knowledge. The scheme is perfectly derivation private and zero-knowledge. For the former, notice that, for any $\hat{\mathbf{r}}$, we have that $\mathsf{PPrv}(ppk, [\mathbf{u}+\mathbf{D}\hat{\mathbf{r}}]_1, \mathbf{r}+\hat{\mathbf{r}}) = \mathbf{k}^\top [\mathbf{D} \otimes \mathbf{D}]_T \cdot ((\mathbf{r}+\hat{\mathbf{r}}) \otimes (\mathbf{r}+\hat{\mathbf{r}})) = \mathsf{PEvl}(ppk, \pi, \hat{\mathbf{r}})$. For the latter, given an instance $[\mathbf{u}]_1$ such that $[\mathbf{u}]_1 = [\mathbf{D}]_1 \mathbf{r}$, we have that $\mathsf{PSim}(psk, [\mathbf{u}]_1) = \mathbf{k}^\top ([\mathbf{u}]_1 \otimes [\mathbf{u}]_1) = \mathbf{k}^\top ([\mathbf{Dr}]_1 \otimes [\mathbf{Dr}]_1) = \mathsf{PPrv}(ppk, [\mathbf{u}]_1, \mathbf{r})$.

4 Rand RCCA PKE for multi-users and multi-ciphertexts

A re-randomizable PKE (Rand-PKE) scheme $\mathcal{PKE}$ is a tuple of five algorithms: (i) The algorithm Setup upon input the security parameter 1^λ produces public parameters $\mathtt{prm}$ which include the description of the message and ciphertext space $\mathcal{M}, \mathcal{C}$; (ii) The algorithm KGen upon input $\mathtt{prm}$, outputs a key pair $(\mathsf{pk}, \mathsf{sk})$; (iii) The algorithm Enc upon inputs pk and a message $\mathtt{M} \in \mathcal{M}$, outputs a ciphertext $\mathtt{C} \in \mathcal{C}$; (iv) The algorithm Dec upon input sk and a ciphertext $\mathtt{C}$, outputs a message $\mathtt{M} \in \mathcal{M}$ or an error symbol $\perp$; (v) The algorithm Rand upon inputs pk and a ciphertext $\mathtt{C}$, outputs another ciphertext $\mathtt{C}'$.

Definition 8 (multi-user and multi-ciphertext Replayable CCA Security). *Consider the experiment* $\mathbf{Exp}^{\mathtt{mRCCA}}$ *in Fig. 2, with parameters* λ*, an adversary* $\mathcal{A}$*, and a PKE scheme* $\mathcal{PKE}$*. We say that* $\mathcal{PKE}$ *is* indistinguishable secure under replayable chosen-ciphertext attacks in the multi-user and multi-ciphertext setting *(mRCCA-secure) if for any PPT adversary* $\mathcal{A}$*:*

$$\mathbf{Adv}^{\mathtt{mRCCA}}_{\mathcal{A},\mathcal{PKE}}(\lambda) := \left| \Pr\left[\mathbf{Exp}^{\mathtt{mRCCA}}_{\mathcal{A},\mathcal{PKE}}(\lambda) = 1\right] - \frac{1}{2} \right| \in \mathtt{negl}(\lambda).$$

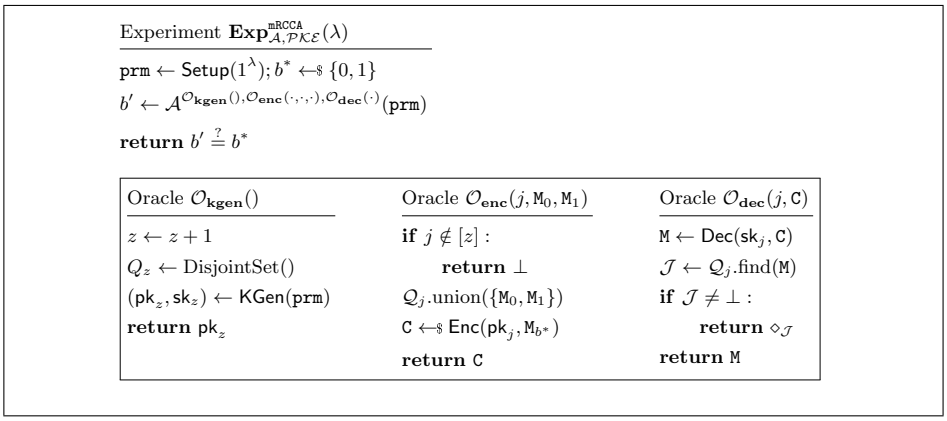

Fig. 2: The multi-user and multi-ciphertext RCCA Security Experiment.

In Fig. 2, for each user j we define $\mathcal{Q}_j$ to be a partition of the set of the challenge messages sent to the encryption oracle for the user j. To do so we use the classical "*Disjoint-Set*" (also called "*Union-Find*") data structure from Tarjan [34]. Whenever two challenge messages are submitted to the encryption oracle, indeed, we merge the sets to which they belong so that a future call to the guarded decryption oracle behaves consistently. This allows us to express in Fig. 2 the syntax of the encryption and the guarded decryption oracle in terms of three operations: DisjointSet() that allows initializing the partition (initially empty), union(S) that adds to the partition the minimal subset of the challenge messages that contains the messages in S meanwhile maintaining invariant the partition property (i.e. a collection of disjoint sets), and find(M) that returns the set in the partition where M belongs to, or $\perp$ if M is not in the set of challenge messages of the user j. We confirm that our definition is indeed the right multi-user and multi-ciphertext extension of the IND-RCCA definition of [7] by showing that our definition tightly implies the UC-RCCA definition of the same paper[5]. For space reasons, we recall the definition of the ideal functionality $\mathcal{F}_{\mathsf{RPKE}}$ which formalizes the notion of replay security for PKE scheme in the universal composability model in [14], where we also give the proof of the theorem below.

Theorem 2. *Let $\mathcal{PKE}$ be a PKE scheme with message space $\mathcal{D}$. There exists a simulator S such that for any static-corruption environment $\mathcal{Z}$ with running time $T_{\mathcal{Z}}$ there exists an adversary $\mathcal{B}$ whose running time is $O(T_{\mathcal{Z}}(\lambda))$ such that:*

$$\left|\Pr\left[\text{REAL}_{\mathcal{Z},\Pi_{\mathcal{PKE}}}(\lambda)=1\right] - \Pr\left[\text{IDEAL}^{\mathcal{F}_{\mathsf{RPKE}}}_{\mathcal{Z},\mathsf{S}}(\lambda)=1\right]\right| \leq 2\mathbf{Adv}^{\mathtt{mumc-RCCA}}_{\mathcal{B},\mathcal{PKE}}(\lambda) + \frac{T_{\mathcal{Z}}}{|\mathcal{D}|}$$

For space reasons, we only informally introduce the notions of perfect re-randomizability and public verifiability, and give more details in [14]. For the notion of perfect re-randomizability, we consider the definition given in [13] which consists of three conditions: (i) the re-randomization of a valid ciphertext and a

[5] In [7], the IND-RCCA notion implies the UC-RCCA notion with a loss of security that is proportional to the running time of the environment.

fresh ciphertext (for the same message) are equivalently distributed; (ii) the re-randomization procedure maintains correctness, i.e., the randomized ciphertext and the original decrypt to the same value, and in particular, invalid ciphertexts keep being invalid; (iii) it is hard to find a valid ciphertext that is not in the support of the encryption scheme. A PKE scheme is publicly verifiable if the validity of the ciphertexts can be checked only using public material.

5 Our Rand-RCCA PKE Scheme

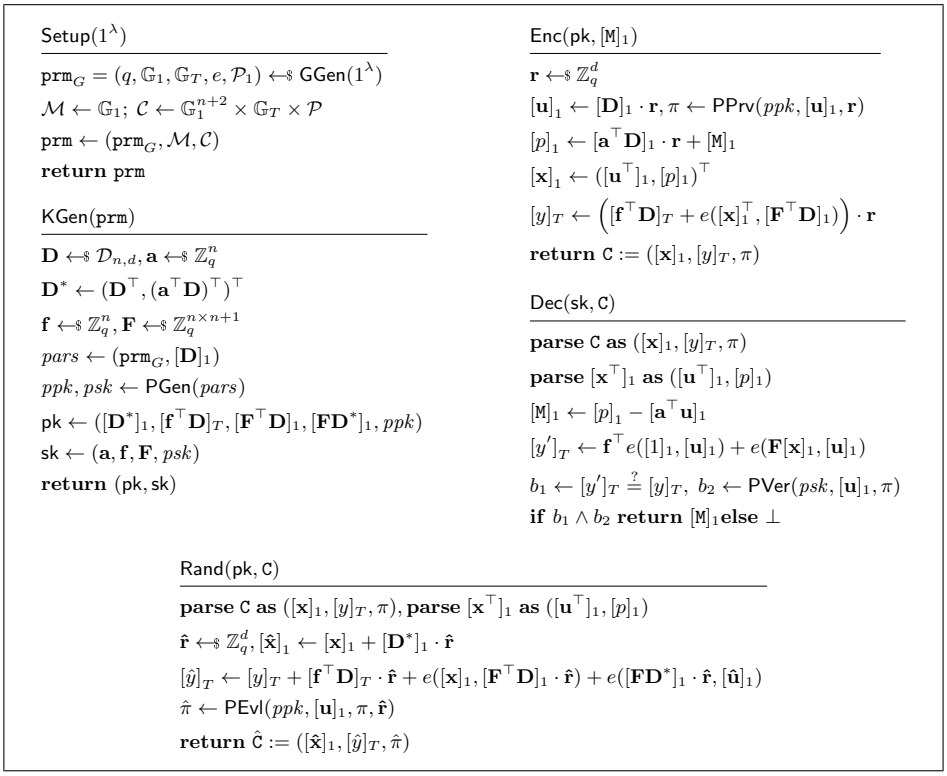

Fig. 3: Rand-RCCA PKE scheme $\mathcal{PKE}$ based on the $\mathcal{D}_{n,d}$-MDDH assumption in type-1 bilinear groups. $\mathcal{P}$ is the support of the proofs for **PS**.

We present our scheme in Fig. 3. With the goal of improving readability for developers, all the operations (and in particular the pairing operations) in the figure are described explicitly using e for the pairing and $\cdot$ for the exponentiations. The scheme can be summarized as a type-1 pairing group version of the scheme in [13] where we additionally append a benign proof to prove almost tight-security. The main technical component from [13] to obtain RCCA security is the *consistency check* at decryption time which checks that $[y]_T \stackrel{?}{=} \mathbf{f}^\top[\mathbf{u}]_T + [\mathbf{x}]_1^\top\mathbf{F}^\top[\mathbf{u}]_1$

Perfect Re-randomizability. The proof of perfect re-randomizability follows from [13] and the derivation privacy of **PS**. Here we highlight the following lemma, whose proof is in [14].

Lemma 2. *For any* $[\mathbf{x}]_1$ *and* $\hat{\mathbf{r}}$*, let* $[\hat{\mathbf{x}}]_1 = [\mathbf{x}]_1 + [\mathbf{D}^*]_1\hat{\mathbf{r}}$*, we have that:*

$$(\mathbf{f}^\top + [\hat{\mathbf{x}}]_1^\top \mathbf{F}^\top)[\hat{\mathbf{u}}]_1 = (\mathbf{f}^\top + [\mathbf{x}]_1^\top \mathbf{F}^\top)[\mathbf{u}]_1 + [\mathbf{f}^\top \mathbf{D}]_T \cdot \hat{\mathbf{r}} + e([\mathbf{x}]_1, [\mathbf{F}^\top \mathbf{D}]_1 \cdot \hat{\mathbf{r}}) + e([\mathbf{F}\mathbf{D}^*]_1 \cdot \hat{\mathbf{r}}, [\hat{\mathbf{u}}]_1)$$

The correctness of $\mathcal{PKE}$ follows from the lemma above and the fact that **PS** and PEvl form a malleable proof system. More details are in [14].

Security. We prove that the security of the scheme reduces to the $\mathcal{D}_{n,d}$-MDDH assumption. Below we state the main theorem.

Theorem 3. *For every PPT adversary* $\mathcal{A}$ *that makes at most* Q_{Enc} *encryption and* Q_{Dec} *decryption queries, there exist adversaries* $\mathcal{B}^{\mathrm{mddh}}$, $\mathcal{B}^{\mathrm{snd}}$ *with similar running time* $T(\mathcal{B}^{\mathrm{mddh}}) \approx T(\mathcal{B}^{\mathrm{snd}}) \approx T(\mathcal{A}) + (Q_{\mathsf{Enc}} + Q_{\mathsf{Dec}}) \cdot \mathtt{poly}(\lambda)$*, where* $\mathtt{poly}(\lambda)$ *is a polynomial independent of* $T(\mathcal{A})$*, and such that*

$$\mathbf{Adv}^{\mathtt{RCCA}}_{\mathcal{A},\mathcal{PKE}}(\lambda) \leq O\left(d \log Q_{\mathsf{Enc}}\right) \cdot \mathbf{Adv}^{\mathtt{MDDH}}_{\mathbb{G}_1,\mathcal{D}_{n,d},\mathcal{B}^{\mathrm{mddh}}}(\lambda) + \log Q_{\mathsf{Enc}} \cdot \mathbf{Adv}^{\mathtt{snd}}_{\mathcal{B}^{\mathrm{snd}},\mathbf{PS}}(\lambda) + O\left(\frac{n^2 Q_{\mathsf{Dec}} Q_{\mathsf{Enc}} \log Q_{\mathsf{Enc}}}{q}\right).$$

Proof. We give a proof only for the single-user, multi-ciphertext case, i.e. when the adversary calls the key generation oracle only once. The proof can be easily generalized[6] to the multi-user case almost equivalently to [4,18].

To simplify the notation, since we are in the single-user setting, we omit the index j (which specifies the user) from both encryption and decryption queries. We let $\mathbf{G}_0$ be the $\mathbf{Exp}^{\mathtt{mRCCA}}_{\mathcal{A},\mathcal{PKE}}$ experiment, and we denote with ϵ_i the advantage of $\mathcal{A}$ to win $\mathbf{G}_i$, i.e. $\epsilon_i := |\Pr[\mathbf{G}_i = 1] - \frac{1}{2}|$.

The games keep track of the number of challenge ciphertexts produced. Specifically, let ctr be a variable that counts the number of challenge ciphertexts output by the encryption oracle: ctr is set to 0 at the beginning of the games and, whenever the adversary calls the encryption oracle, it is increased.

Game $\mathbf{G}_1$. This game is identical to the previous one, but the encryption oracle computes the values $[y]_T$ and $[p]$ using secret keys (instead of public keys). Specifically, upon the j-th query to the encryption oracle, the game computes the ciphertext $\mathtt{C}_j = ([\mathbf{x}_j], [y_j]_T, \pi_j)$ as described by the encryption procedure,

[6] We rely on the self-reducibility of the MDDH assumption: in particular, we can generate m different matrices $\mathbf{D}_j$ (one for each user) from one single challenge of the (many-fold) MDDH-assumption and adapt accordingly the ciphertexts, namely, by mapping the ciphertext for the j-th user through the same linear transformation that maps the MDDH-challenge matrix to the matrix $\mathbf{D}_j$.

but where we compute $[y_j]_T \leftarrow \mathbf{f}^\top[\mathbf{u}_j] + [\mathbf{x}_j]^\top \mathbf{F}^\top[\mathbf{u}_j]$ and $[p_j] \leftarrow \mathbf{a}^\top[\mathbf{u}_j] + [\mathtt{M}_{j,b^*}]$. By linearity, this game is perfectly equivalent to the previous one, thus $\epsilon_1 = \epsilon_0$.

Game $\mathbf{G}_2$. This game is identical to the previous one, but the encryption oracle simulates the benign proofs π. We rely on the perfect zero-knowledge of the benign proof system. The reduction is standard, therefore we omit it. Since the proof system satisfies perfect zero-knowledge we have $\epsilon_2 = \epsilon_1$.

Game $\mathbf{G}_3$. At the very beginning, the game additionally samples matrices $\bar{\mathbf{D}}_b \leftarrow\!\!\$\ \mathbb{Z}_q^{n\times d}$ for $b \in \{0,1\}$, and sets $\mathbf{D}_b \leftarrow (\mathbf{D}|\bar{\mathbf{D}}_b)$. The encryption oracle in this game samples $[\mathbf{u}]$ from the span of $[\mathbf{D}_0]$. We apply a standard reduction to the Q_{Enc}-fold $\mathcal{D}_{n,d}$-MDDH assumption, twice, and we prove that no adversary can distinguish this game from the previous one: we first tightly switch the vectors in the challenge ciphertexts from the span of $[\mathbf{D}]$ to uniformly random vectors of $\mathbb{G}_1^n$; next, we use the Q_{Enc}-fold $\mathcal{D}_{n,2d}$-MDDH assumption to switch these vectors from random to the span of $[\mathbf{D}_0]$. The proof of this step is standard: in [14], we show how we can build adversaries $\mathcal{B}$, $\mathcal{B}'$ such that

$$|\epsilon_3 - \epsilon_2| \leq \mathbf{Adv}^{\mathsf{Q}_{\mathsf{Enc}}-\mathtt{MDDH}}_{\mathbb{G}_1,\mathcal{D}_{n,d},\mathcal{B}}(\lambda) + \mathbf{Adv}^{\mathsf{Q}_{\mathsf{Enc}}-\mathtt{MDDH}}_{\mathbb{G}_1,\mathcal{D}_{n,2d},\mathcal{B}'}(\lambda)$$

Game $\mathbf{G}_4$. In this experiment, we add an explicit check to the decryption oracle. First recall that $\mathbf{D}^*$ is defined in Fig. 3 as the matrix whose first n rows are equal to $\mathbf{D}$ and last row is equal to $\mathbf{a}^\top\mathbf{D}$. Upon query $\mathtt{C} := ([\mathbf{x}], [y]_T, \pi)$ to the decryption oracle, where $[\mathbf{x}]^\top := ([\mathbf{u}]^\top, [p])$, the oracle additionally checks that:

$$\mathbf{u} \in span(\mathbf{D}) \vee \exists j : \mathbf{D}^{*\perp}\mathbf{x}_j = \mathbf{D}^{*\perp}\mathbf{x} \tag{3}$$

where $\mathbf{D}^{*\perp}\mathbf{D}^* = 0$, and $\mathcal{Q}_{\mathsf{Enc}} = \{\mathtt{C}_j = ([\mathbf{x}_j], [y_j]_T, \pi_j) : j \leq [\mathrm{ctr}]\}$ is the set of challenge ciphertexts. If the condition holds, the decryption oracle proceeds by running the decryption procedure as usual, otherwise it returns $\perp$ to the adversary. We notice that the new condition can be checked efficiently since we know $\mathbf{D} \in \mathbb{Z}_q^{n\times d}$ and $\mathbf{a} \in \mathbb{Z}_q^n$.

The distinguishing event between $\mathbf{G}_4$ and $\mathbf{G}_3$ is that the adversary queries the decryption oracle with a ciphertext that would not decrypt to $\perp$ (according to the original decryption rules of $\mathbf{G}_3$), but where Eq. (3) holds. We call such query to the decryption oracle a "critical query", i.e. a decryption query where:

- $[\mathbf{u}] \notin span([\mathbf{D}])$ and $\forall j : \mathbf{D}^{*\perp}\mathbf{x}_j \neq \mathbf{D}^{*\perp}\mathbf{x}$ (the latter implies that $[\mathbf{u}]$ is not the result of an honest rerandomization of a previous challenge ciphertext)
- π is valid, and $[y]_T = \mathbf{f}^\top[\mathbf{u}]_T + [\mathbf{x}]^\top\mathbf{F}^\top[\mathbf{u}]$, i.e., the consistency check holds.

For this step, we refer to Lemma 3.

Game $\mathbf{G}_5$. This game is equivalent to $\mathbf{G}_4$, but we modify the rules of the decryption oracle once again. For any j, let $\mathtt{M}_{j,0}$ and $\mathtt{M}_{j,1}$ be the challenge messages queried by $\mathcal{A}$ at the j-th query to the encryption oracle. Upon decryption query $\mathtt{C} = ([\mathbf{x}], [y]_T, \pi)$, if $\exists j : \mathbf{D}^{*\perp}\mathbf{x}_j = \mathbf{D}^{*\perp}\mathbf{x}$ where recall $\mathcal{Q}_{\mathsf{Enc}} = \{([\mathbf{x}_j], [y_j], \pi_j) : j \leq \mathrm{ctr}\}$, and both the proof π verifies and the consistency check holds, then the decryption oracle immediately returns the symbol $\diamond_{\mathcal{J}}$ where $\mathcal{J} \leftarrow \mathcal{Q}.\mathrm{find}(\mathtt{M}_{j,0})$.

Notice that we can rewrite the decryption procedure as $\mathtt{M} = (-\mathbf{a}^\top, 1)[\mathbf{x}]$. We observe that the vector $(-\mathbf{a}^\top, 1)$ is in the span of $\mathbf{D}^{*\perp}$, since it holds that $(-\mathbf{a}^\top, 1)\mathbf{D}^* = -\mathbf{a}^\top\mathbf{D} + \mathbf{a}^\top\mathbf{D} = 0$. Thus, at any decryption query, if $\mathbf{D}^{*\perp}\mathbf{x}_j = \mathbf{D}^{*\perp}\mathbf{x}_j$ for some challenge ciphertext $\mathtt{C}_j$ then $(-\mathbf{a}^\top, 1)[\mathbf{x}_j] = (-\mathbf{a}^\top, 1)[\mathbf{x}]$, and therefore the decryption oracle would compute the message $\mathtt{M}_{j,b^*}$ and output the symbol $\diamond_{\mathcal{J}}$, where $\mathcal{J} = \mathcal{Q}.\mathrm{find}(\mathtt{M}_{j,b^*})$. Moreover, notice that $\mathcal{Q}.\mathrm{find}(\mathtt{M}_{j,b^*}) = \mathcal{Q}.\mathrm{find}(\mathtt{M}_{j,0})$ by definition of the security experiment. This shows that $\epsilon_5 = \epsilon_4$.

Game $\mathbf{G}_6$. In this last step, we encrypt random messages. Formally, at the j-th encryption query the oracle (on input messages $\mathtt{M}_{j,0}, \mathtt{M}_{j,1}$) encrypts the message $\mathtt{M}_{j,b^*} + \mathtt{R}_j$, where $\mathtt{R}_j$ is random. Clearly, it holds that $\epsilon_6 = 0$ as in fact, because of the change introduced in $\mathbf{G}_6$, the ciphertexts are independent of the challenge bit b^*, and, by the changes introduced in $\mathbf{G}_4$ and $\mathbf{G}_5$, the decryption queries are independent of the challenge bit. We prove that $\mathbf{G}_5$ and $\mathbf{G}_6$ are indistinguishable, as this step is almost the same as in [18], we defer its proof to [14].

Lemma 3. *For any PPT adversary $\mathcal{A}$, we build PPT adversaries $\mathcal{B}$, $\mathcal{B}'$ with running times similar to $\mathcal{A}$ such that:*

$$|\epsilon_3 - \epsilon_4| \leq O(d \log Q_{\mathsf{Enc}})\mathbf{Adv}^{\mathtt{MDDH}}_{\mathbb{G}_1,\mathcal{D}_{n,d},\mathcal{B}}(\lambda) + \log Q_{\mathsf{Enc}}\mathbf{Adv}^{\mathtt{snd}}_{\mathcal{B}'}(\lambda) + O\left(\frac{n^2 Q_{\mathsf{Dec}} Q_{\mathsf{Enc}} \log Q_{\mathsf{Enc}}}{q}\right)$$

Proof. We denote the probability that the adversary $\mathcal{A}$ wins game $\mathbf{H}_x$ by $\epsilon_{\mathbf{H}_x}$. In the following, we will bound $\epsilon_{\mathbf{H}_0}$ via a sequence of games.

Hybrid $\mathbf{H}_0$. This hybrid is the same as $\mathbf{G}_3$ but immediately outputs 1 if the adversary makes a "critical query". Specifically, the hybrid executes $\mathbf{G}_3$ but the decryption oracle upon input $\mathtt{C}$ parses it as $([\mathbf{x}], [y]_T, \pi)$ and checks that Eq. (3) holds; if it holds, the decryption oracle continues as before. Otherwise, returns the message "critical", and $\mathbf{H}_0$ stops the interaction, immediately returning 1. Since the hybrid outputs 1 when the distinguishing event between $\mathbf{G}_3$ and $\mathbf{G}_4$ happens, we have that $|\epsilon_3 - \epsilon_4| \leq \epsilon_{\mathbf{H}_0}$. Also notice that the checks in Eq. (3) can be efficiently performed given the knowledge of $\mathbf{D}$.

Hybrid $\mathbf{H}_1$. This hybrid is preparatory for the next one. We inject randomness into the encryption/decryption keys, adding a vector $(\mathbf{z}\mathbf{D}^\perp)$ to the secret key $\mathbf{f}^\top$, common to all the encryption queries, where $\mathbf{z} \in \mathbb{Z}_q^{n-d}$. Specifically, at the very beginning of the experiment we sample the vector $\mathbf{z} \leftarrow\!\!\$\, \mathbb{Z}_q^{n-d}$, we sample $\mathbf{f}$ and compute the public key material $[\mathbf{f}^\top\mathbf{D}]$ and moreover:

- The encryption oracle, at the j-th query, computes the values $[y_j]_T$ as follows:

$$[y_j]_T \leftarrow (\mathbf{f}^\top + \mathbf{z}\mathbf{D}^\perp)[\mathbf{u}_j]_T + [\mathbf{x}_j]^\top\mathbf{F}^\top[\mathbf{u}_j]$$

- Similarly, the decryption oracle, upon input the ciphertext $\mathtt{C} = ([\mathbf{x}], [y]_T, \pi)$ computes the bit b_1 (i.e. the bit of the consistency check) by computing the value $[y']_T$ and checking if $[y]_T \stackrel{?}{=} [y']_T$ where $[y']_T$ is computed as:

$$[y']_T \leftarrow (\mathbf{f}^\top + \mathbf{z}\mathbf{D}^\perp)[\mathbf{u}]_T + [\mathbf{x}]^\top\mathbf{F}^\top[\mathbf{u}]$$

These new rules do not change the view of the adversary since both $\mathbf{f}^\top$ and $\mathbf{f}^\top + \mathbf{z}\mathbf{D}^\perp$ are uniformly distributed over $\mathbb{Z}_q^{1\times n}$ given the public key material $[\mathbf{f}^\top\mathbf{D}]$. Thus we obtain $\epsilon_{\mathbf{H}_1} = \epsilon_{\mathbf{H}_0}$.

Hybrid $\mathbf{H}_2$ Let $P : \{0,1\}^* \to \mathbb{Z}_q^{1\times n-d}$ be an uniformly random function. In this hybrid we use the following rules for encryption and decryption:

- The encryption oracle, at the j-th query, computes the values $[y_j]_T$ as follows:

$$[y_j]_T \leftarrow (\mathbf{f}^\top + \boxed{P(j)}\,\mathbf{D}^\perp)[\mathbf{u}_j]_T + [\mathbf{x}_j]^\top\mathbf{F}^\top[\mathbf{u}_j]$$

- For each decryption oracle query, we first define a set $\mathcal{S}$ over which the decryption oracle iterates to test the consistency check. The definition of the set $\mathcal{S}$ is carefully crafted to define the behavior of the hybrid experiment in case of *replay attack* from the adversary
  Recall that ctr counts the number of challenge ciphertexts output by the encryption oracle and that $\mathcal{Q}_{\mathsf{Enc}} = \{\mathtt{C}_j = ([\mathbf{x}_j], [y_j]_T, \pi_j) : j \le \mathrm{ctr}\}$. Upon input the ciphertext $\mathtt{C} = ([\mathbf{x}], [y]_T, \pi)$, the decryption oracle first sets:

$$\begin{aligned} &\mathcal{S} := \{j\} && \text{if } \exists j \le \mathrm{ctr} : \mathbf{D}^{*\perp}[\mathbf{x}] = \mathbf{D}^{*\perp}[\mathbf{x}_j] \\ &\mathcal{S} := \{j : j \le \mathrm{ctr}\} && \text{otherwise} \end{aligned}$$

  then it computes the bit b_1 (i.e. the bit of the consistency check for $\mathtt{C}$, see Fig. 3) differently by checking that

$$\boxed{\exists j \in \mathcal{S}} : [y]_T \stackrel{?}{=} (\mathbf{f}^\top + \boxed{P(j)}\,\mathbf{D}^\perp)[\mathbf{u}]_T + [\mathbf{x}]^\top\mathbf{F}^\top[\mathbf{u}].$$

Moving from $\mathbf{H}_1$ to $\mathbf{H}_2$ requires a series of hybrids $\mathbf{H}_{1,i,i'}$, $i \in [\log(Q_{\mathsf{Enc}})], i' \in [6]$. We give in [14] the formal definitions of all these hybrids, and we highlight their differences.

Hybrid $\mathbf{H}_{1,i,0}$. Let P_i be a random function that takes in input strings of length i (for $i = 0$, we can imagine this as a constant function defined on the empty string) and returns row vectors of length $n - d$.

- On input the j-th query, the encryption oracle samples $[\mathbf{u}_j]$ from the span of $[\mathbf{D}_0]$. The element $[y_j]_T$ is computed as

$$[y_j]_T \leftarrow (\mathbf{f} + \boxed{P_i(j_{|i})}\,\mathbf{D}^\perp)[\mathbf{u}_j] + [\mathbf{x}_j]^\top\mathbf{F}^\top[\mathbf{u}_j].$$

- Upon input the ciphertext $\mathtt{C} = ([\mathbf{x}], [y]_T, \pi)$, define:

$$\begin{aligned} &\mathcal{S} := \boxed{\{j_{|i}\}} && \text{if } \exists j \le \mathrm{ctr} : \mathbf{D}^{*\perp}[\mathbf{x}] = \mathbf{D}^{*\perp}[\mathbf{x}_j] \\ &\mathcal{S} := \boxed{\{j_{|i} : j \le \mathrm{ctr}\}} && \text{otherwise} \end{aligned}$$

  it then executes the same code of the previous hybrid.

When $i = 0$, for any value j the string $j_{|0}$ is equal to the empty string, thus, in $\mathbf{H}_{1,0,0}$, the random function P_0 is always called on input the empty string. In particular, either when $\mathbf{D}^{*\perp}[\mathbf{x}] = \mathbf{D}^{*\perp}[\mathbf{x}_j]$ holds or when it does not, the consistency check performed is exactly the same. Thus the difference between hybrid $\mathbf{H}_{1,0,0}$ and $\mathbf{H}_1$ is only syntactical.

Hybrid $\mathbf{H}_{1,i,1}$ This hybrid is equivalent to the previous one, but here the encryption oracle, on input the j-the query, generates $[\mathbf{u}_j]$ in the span of $[\mathbf{D}_{j[i+1]}]$. We rely on the MDDH assumption to prove indistinguishability between the two hybrids. We proceed in two steps:

- We first switch the j-th vector $[\mathbf{u}_j]$ computed by the encryption oracle to a vector in the span of $[(\mathbf{D}|\mathbf{U})]$, where $\mathbf{U}$ is uniform over $\mathbb{Z}_q^{n\times d}$, if the $(i+1)$-th bit of the binary representation of j is equal to 1. We call this intermediate hybrid $\mathbf{H}_{A_i}$.
- Finally, we switch the j-th vector $[\mathbf{u}_j]$ computed by the encryption oracle to a vector in the span of $[(\mathbf{D}|\bar{\mathbf{D}}_1)] = [\mathbf{D}_1]$, if the $(i+1)$-th bit of the binary representation of j is equal to 1.

First we show indistinguishability between $\mathbf{H}_{1,i,0}$ and $\mathbf{H}_{A_i}$. Let $\mathcal{B}_A$ be an MDDH-adversary receiving the Q_{Enc}-fold $\mathcal{D}_{n,d}$-MDDH challenge $([\bar{\mathbf{D}}_0], [\mathbf{h}_1], \ldots, [\mathbf{h}_{Q_{\mathsf{Enc}}}])$ as input. $\mathcal{B}_A$ can sample a random matrix $\mathbf{D} \leftarrow\!\!\$\ \mathcal{D}_{n,d}$, a random matrix $\bar{\mathbf{D}}_1 \in \mathbb{Z}_q^{n\times d}$, the secret material $\mathbf{a} \leftarrow\!\!\$\ \mathbb{Z}_q^n$, $\mathbf{f} \leftarrow\!\!\$\ \mathbb{Z}_q^d, \mathbf{F} \leftarrow\!\!\$\ \mathbb{Z}_q^{n\times n+1}$ and the secret material for the benign proof system (since $\mathcal{B}_A$ knows $\mathbf{D}$, this can be easily achieved running $\mathsf{PGen}([\mathbf{D}])$). Finally, $\mathcal{B}_A$ samples a challenge bit b and gives the public key of the scheme to $\mathcal{A}$. $\mathcal{B}_A$ simulates the encryption oracle as follows. On input the j-th pair of messages $(\mathtt{M}_0, \mathtt{M}_1)$:

- if the $(i+1)$-th bit of the binary representation of j is equal to 0, the adversary sets $[\mathbf{u}_j] \leftarrow [\mathbf{D}_0]\mathbf{r}_j$,
- else, samples a random vector $\tilde{\mathbf{r}} \in \mathbb{Z}_q^d$, and computes $[\mathbf{u}_j] \leftarrow [\mathbf{D}]\tilde{\mathbf{r}} + [\mathbf{h}_j]$.

Note that $\mathcal{B}_A$ can still simulate the decryption oracle, because of the knowledge of the secret material $\mathbf{a}, \mathbf{f}, \mathbf{F}$ and of the matrix $\mathbf{D}$. Since $\mathcal{B}_A$ knows both the matrix $\mathbf{D}$ and the vector $\mathbf{a}$, can always find a matrix $\mathbf{D}^{*\perp}$ such that $\mathbf{D}^{*\perp}\mathbf{D}^* = 0$. This allows $\mathcal{B}_A$ to catch critical queries. If the tuple is a real MDDH tuple, i.e. $[\mathbf{h}_j] = [\bar{\mathbf{D}_0}]\mathbf{r}_j$, the game described is perfectly equivalent to $\mathbf{H}_{1,i,0}$. Otherwise, if the challenge vectors are uniformly random, the game simulated is equivalent to $\mathbf{H}_{A,i}$. The next step is to switch the j-th vector $[\mathbf{u}_j]$ computed by the encryption oracle to a vector in the span of $[(\mathbf{D}|\bar{\mathbf{D}}_1)] = [\mathbf{D}_1]$ if the $(i+1)$-th bit of the binary representation of j is equal to 1. This transformation is similar to the previous one, therefore we omit the details. Altogether, combining the previous adversaries, we obtain an adversary $\mathcal{C}$ such that: $|\epsilon_{\mathbf{H}_{1,i,1}} - \epsilon_{\mathbf{H}_{1,i,0}}| \leq 2(n-d)\mathbf{Adv}^{\mathtt{mddh}}_{\mathbb{G}_1,\mathcal{D}_{n,d},\mathcal{C}}(\lambda) + \frac{2}{q-1}$.

Hybrid $\mathbf{H}_{1,i,2}$ We add an explicit check to the decryption oracle. Specifically, at each decryption oracle query the hybrid additionally checks if $\mathbf{u} \notin span(\mathbf{D}_0) \cup span(\mathbf{D}_1)$, and if it is the case the decryption oracle returns immediately $\perp$ to

the adversary. We rely on the soundness of the underlying benign proof system. In particular, the only condition that would allow to distinguish between this hybrid and the previous one is to query the decryption oracle with a ciphertext $\mathtt{C} = ([\mathbf{x}], [y]_T, \pi)$ where:

- $\mathbf{u} \notin span(\mathbf{D}_0) \cup span(\mathbf{D}_1)$
- the decryption oracle in the hybrid $\mathbf{H}_{1,i,1}$ would not return $\perp$.

For such query it holds that $\mathsf{PVer}(psk, [\mathbf{u}], \pi) = 1$. We build an adversary $\mathcal{B}$ against the $(\mathcal{L}^{\text{sim}}, \mathcal{L}^{\text{ver}}, \mathcal{L}^{\text{snd}})$-soundness of the proof system. (Recall that $\mathcal{L}^{\text{snd}} = \mathcal{L}^{\text{sim}} = span(\mathbf{D}_0) \cup span(\mathbf{D}_1)$, and $\mathcal{L}^{\text{ver}} = \mathbb{Z}_q^n$.)

The adversary $\mathcal{B}$ samples the secret material $\mathbf{a}, \mathbf{f}, \mathbf{F}$; then, it queries the challenger to obtain the public key of the benign proof system, associated with the matrix $\mathbf{D}$, and finally gives $\mathcal{A}$ all the public key material. The adversary $\mathcal{B}$ can easily simulate the encryption oracle since it knows all the necessary information. To compute the proof π_j associated with the j-th encryption oracle query, it queries the simulation oracle offered by the challenger: it holds that $\mathbf{u}_j \in \mathcal{L}^{\text{sim}}$, for all $j \in [Q_{\mathsf{Enc}}]$. When the adversary makes a decryption query, $\mathcal{B}$ needs to verify that the proof π is accepted by PVer; so, it forwards $(\mathbf{u}, \pi)$ to the challenger. Since $\mathcal{L}^{\text{ver}}$ is equal to $\mathbb{Z}_q^n$, the verification oracle always returns a verdict bit, and $\mathcal{B}$ can proceed in the natural way the simulation of the decryption oracle. At some point $\mathcal{B}$ queries the verification oracle with some $([\mathbf{u}], \pi)$ such that $\mathbf{u} \notin span(\mathbf{D}_0) \cup span(\mathbf{D}_1)$, i.e., $\mathbf{u} \notin \mathcal{L}^{\text{snd}}$, but $\mathsf{PVer}(psk, [\mathbf{u}], \pi) = 1$. This is the event that lets $\mathcal{B}$ win the soundness game. The adversary $\mathcal{B}$ runs in time $T(\mathcal{B}) \approx T(A) + (Q_{\mathsf{Enc}} + Q_{\mathsf{Dec}}) \cdot \mathtt{poly}(\lambda)$, where $\mathtt{poly}$ is a polynomial independent of $T(A)$. Moreover, notice that when the distinguishing event happens the adversary $\mathcal{B}$ wins the soundness game, thus: $|\epsilon_{\mathbf{H}_{1,i,2}} - \epsilon_{\mathbf{H}_{1,i,1}}| \leq \mathbf{Adv}^{\mathtt{snd}}_{\mathcal{B},\mathbf{PS}}(\lambda)$.

Hybrid $\mathbf{H}_{1,i,3}$ In this hybrid, we increase the entropy of the secret keys during encryption queries.

- The encryption oracle, at the j-th query, computes the values $[y_j]_T$ as follows:

$$[y_j]_T \leftarrow (\mathbf{f}^\top + P_{i+1}(j_{|i+1}) \mathbf{D}^\perp)[\mathbf{u}_j] + [\mathbf{x}_j]^\top \mathbf{F}^\top [\mathbf{u}_j].$$

- The decryption oracle, upon input the ciphertext $\mathtt{C} = ([\mathbf{x}], [y]_T, \pi)$ additionally checks that $\exists d$ s.t. $\mathbf{u} \in span(\mathbf{D}_d)$ and in such a case it sets:

$$\begin{aligned} \mathcal{S} &:= \{j_{|i} \,\|d\} && \text{if } \exists j \leq \text{ctr} : \mathbf{D}^{*\perp}[\mathbf{x}] = \mathbf{D}^{*\perp}[\mathbf{x}_j] \\ \mathcal{S} &:= \{j_{|i} \,\|d : j \leq \text{ctr}\} && \text{otherwise} \end{aligned}$$

  and it continues executing the same code as the previous hybrid.

We prove that $|\epsilon_{\mathbf{H}_{1,i,2}} - \epsilon_{\mathbf{H}_{1,i,3}}|$ is negligible. We first transit to an intermediate hybrid $\mathbf{H}'_i$ where instead of using the function $P_i(\cdot)\mathbf{D}^\perp$, we use the function $P'_i(\cdot) := P_i^{(0)}(\cdot)\mathbf{D}_0^\perp + P_i^{(1)}(\cdot)\mathbf{D}_1^\perp$, where $P_i^{(0)}$ and $P_i^{(1)}$ are two uniformly random functions with domain $\{0,1\}^i$. Notice that $P'_i(\cdot)$ is an uniformly random function

that maps strings in $\{0,1\}^i$ to vectors in $rowspan(\mathbf{D}_0^\perp) + rowspan(\mathbf{D}_1^\perp)$ while $P_i(\cdot)\mathbf{D}^\perp$ is an uniformly random function that maps string in $\{0,1\}^i$ to vectors in $rowspan(\mathbf{D}^\perp)$. Thus the distinguishing event between $\mathbf{H}_{i,j,2}$ and this intermediate hybrid is the event that $rowspan(\mathbf{D}_0^\perp) + rowspan(\mathbf{D}_1^\perp) \neq rowspan(\mathbf{D}^\perp)$. The latter event happens with probability at most $1/q$: in fact, the event happens if and only if the subspace $span(\bar{\mathbf{D}}_0|\bar{\mathbf{D}}_1)$ has dimension strictly less than $2d$ and recall that the columns of such matrices are sampled uniformly at random. Next, we define the function $P_{i+1}^{(b)} : \{0,1\}^{i+1} \to \mathbb{Z}_q^{1\times(n-2d)}$, $\forall b \in \{0,1\}$:

$$P_{i+1}^{(b)}(\mathtt{x}) = \begin{cases} P_i^{(b)}(\mathtt{x}_{|i}), & \mathtt{x}[i+1] \neq b \\ \tilde{P}_i^{(b)}(\mathtt{x}_{|i}), & \text{else} \end{cases}$$

where $P_i, \tilde{P}_i$ are two uniformly (and independent) random functions. Notice that $P_{i+1}^{(b)}$ is an uniformly random function.

We define a second intermediate hybrid $\mathbf{H}'_{i+1}$ where for the encryption oracle queries instead of using the random function P'_i applied to the indexes $j_{|i}$ we use the function P'_{i+1} applied to the indexes $j_{|i+1}$, and for the decryption oracle queries we use P'_{i+1} applied to $(j_{|i}\|d)$, where d is such that $\mathbf{u}_j \in span(\mathbf{D}_d)$ (as described in the $\mathbf{H}_{1,i,3}$). We show that $\mathbf{H}'_i$ and $\mathbf{H}'_{i+1}$ are equivalently distributed. Indeed, in this second intermediate hybrid, at the j-th encryption oracle query we compute $[y_j]_T \leftarrow (\mathbf{f}^\top + P'_{i+1}(j_{|i+1}))[\mathbf{u}_j] + [\mathbf{x}_j]^\top \mathbf{F}^\top [\mathbf{u}_j]$. Moreover, we have that $P'_{i+1}(j_{|i+1})\mathbf{u}_j = P'_i(j_{|i})\mathbf{u}_j$, in fact:

$$\begin{aligned} P'_{i+1}(j_{|i+1})\mathbf{u}_j &= \left(P_i^{(1-j[i+1])}(j_{|i})\mathbf{D}^\perp_{1-j[i+1]} + \tilde{P}_i^{(j[i+1])}(j_{|i})\mathbf{D}^\perp_{j[i+1]}\right)\mathbf{D}_{j[i+1]}\mathbf{r}_j \\ &= \left(P_i^{(1-j[i+1])}(j_{|i})\mathbf{D}^\perp_{1-j[i+1]}\right)\mathbf{D}_{j[i+1]}\mathbf{r}_j \\ &= \left(P_i^{(0)}(j_{|i})\mathbf{D}_0^\perp + P_i^{(1)}(j_{|i})\mathbf{D}_1^\perp\right)\mathbf{D}_{j[i+1]}\mathbf{r}_j \\ &= P'_i(j_{|i})\mathbf{D}_{j[i+1]}\mathbf{r}_j = P'_i(j_{|i})\mathbf{u}_j \end{aligned}$$

In the above derivation, we first applied the definitions of P_{i+1} and $\mathbf{u}_j$, then we simplified the second term by noticing that $\mathbf{D}^\perp_{j[i+1]}\mathbf{D}_{j[i+1]} = 0$, then for the same exact reason we can add the component $P_j^{(j[i+1])}(j_{|i})\mathbf{D}_{j[i+1]}$, and finally we have the definition of P'_i.

Similarly, for the decryption oracle queries with input $\mathtt{C} = ([\mathbf{x}], [y]_T, \pi)$ where $\exists d : \mathbf{u} \in span(\mathbf{D}_d)$, we have that $P'_{i+1}(j_{|i+1})\mathbf{u} = P'_i(j_{|i})\mathbf{u}$. The derivation is identical to before. Thus the two intermediate hybrids are equivalent.

Finally, we show that the second intermediate hybrid, $\mathbf{H}'_{i+1}$, is statistically close to $\mathbf{H}_{i,1,3}$; in fact, the only difference is that in the latter hybrid we use the function $P_{j+1}(\cdot)\mathbf{D}^\perp$. Equivalently as before, the two random functions are not equivalently distributed only when $span(\bar{\mathbf{D}}_0\|\bar{\mathbf{D}}_1)$ has rank less then $2d$, which happens with probability at most $1/q$. Thus $|\epsilon_{\mathbf{H}_{1,i,2}} - \epsilon_{\mathbf{H}_{1,i,3}}| \leq \frac{2}{q}$.

Hybrid $\mathbf{H}_{1,i,4}$ We remove the direct check $[\mathbf{u}]_1 \in span([\mathbf{D}_1]_1) \cup span([\mathbf{D}_0]_1)$ introduced in $\mathbf{H}_{1,i,2}$. This removal can only increase the winning probability of the adversary. Thus $\epsilon_{\mathbf{H}_{1,i,3}} \leq \epsilon_{\mathbf{H}_{1,i,4}}$.

Hybrid $\mathbf{H}_{1,i,5}$ To decrypt, we increase the number of keys used by the decryption oracle to compute the bit b_1.

$$\mathcal{S} := \{j_{|i} \,\|b : b \in \{0,1\}\} \qquad \text{if } \exists j \leq \text{ctr} : \mathbf{D}^{*\perp}[\mathbf{x}] = \mathbf{D}^{*\perp}[\mathbf{x}_j]$$
$$\mathcal{S} := \{j_{|i} \,\|b : b \in \{0,1\}, j \leq \text{ctr}\} \qquad \text{otherwise}$$

This change can only increase the winning probability of the adversary since the set of the strings $\mathcal{S}$ used in $\mathbf{H}_{1,i,5}$ contains the set of strings used in $\mathbf{H}_{1,i,4}$.

As for non-critical queries, we need to show that the view of the adversary does not change: in particular, any non-critical query that decrypts to $\perp$ in $\mathbf{H}_{1,i,4}$ should decrypt to $\perp$ in $\mathbf{H}_{1,i,5}$ as well. This is easy to prove when the decryption query has $[\mathbf{u}] \in span([\mathbf{D}])$: indeed, even if we modify the set $\mathcal{S}$, this change does not affect the way we decrypt such queries (recall that any key $P_{i+1}(\cdot)$ is then multiplied by $\mathbf{D}^{\perp}$.) Also, a non-critical query could be a query for which it holds that there exists $j \in [Q_{\mathsf{Enc}}]$ such that $\mathbf{D}^{*\perp}\mathbf{x}_j$ is equal to $\mathbf{D}^{*\perp}\mathbf{x}$. If a query of this form succesfully decrypts in $\mathbf{H}_{1,i,4}$, the same happens in $\mathbf{H}_{1,i,5}$: again, this is because $\mathcal{S}$ in the latter hybrid is a superset of $\mathcal{S}$ in $\mathbf{H}_{1,i,4}$. But, it is still possible that a query of this form decrypts to $\perp$ in $\mathbf{H}_{1,i,4}$, but the 'augmented' $\mathcal{S}$ in this new hybrid makes the consistency bit b_1 be 1, for some new key: we bound the probability of a similar event since we know that the only way to learn the image of the random function $P_{i+1}(\cdot)$ is via oracle queries to $\mathcal{O}_{\mathbf{dec}}$ and $\mathcal{O}_{\mathbf{enc}}$. By union bound, we obtain a statistical distance of $O(Q_{\mathsf{Enc}}Q_{\mathsf{Dec}}/q)$.

$$\epsilon_{\mathbf{H}_{1,i,4}} - O(Q_{\mathsf{Enc}}Q_{\mathsf{Dec}}/q) \leq \epsilon_{\mathbf{H}_{1,i,5}}.$$

Hybrid $\mathbf{H}_{1,i,6}$ This hybrid is equivalent to the previous one, but the decryption oracle computes a different set $\mathcal{S}$, as follows:

$$\mathcal{S} := \{j_{|i+1}\} \qquad \text{if } \exists j \leq \text{ctr} : \mathbf{D}^{*\perp}[\mathbf{x}] = \mathbf{D}^{*\perp}[\mathbf{x}_j]$$
$$\mathcal{S} := \{j_{|i+1} : j \leq \text{ctr}\} \qquad \text{otherwise}$$

Notice that the set $\mathcal{S}$ as defined in $\mathbf{H}_{1,i,6}$ might be a (strict) subset of the set $\mathcal{S}$ as defined in $\mathbf{H}_{1,i,5}$. Thus the distinguishing event is that the consistency check would pass in $\mathbf{H}_{1,i,5}$ but it would not pass in $\mathbf{H}_{1,i,6}$. In particular, such consistency check passes for an index of the form $j_i\|1$, such that $j[i+1] = 0$ and $j \leq \text{ctr}$, and by the definition of the distinguishing event the integer representation of $(j_i\|1) \cdot 2^{|\log Q_{\mathsf{Enc}}|-i-1}$ is bigger than ctr. Thus the key $\mathbf{f}^{\top} + P_i(j_i\|1)\mathbf{D}^{\perp}$ was never used for an encryption query. The only way an adversary can learn information about one of such keys is via decryption queries. In particular, each decryption query can at most decrease the set of possibilities (namely a valid y that matches the consistency check) by one. Moreover, the number of such keys is (very loosely) upper-bounded by Q_{Enc}, thus by union bound over all such keys and over all the decryption queries we obtain: $|\epsilon_{\mathbf{H}_{1,i,6}} - \epsilon_{\mathbf{H}_{1,i,5}}| \leq \frac{Q_{\mathsf{Enc}} \cdot Q_{\mathsf{Dec}}}{q - Q_{\mathsf{Dec}}}$.

Hybrid $\mathbf{H}_{1,i+1,0}$ We then switch back the distribution of $[\mathbf{u}_j]$ to the span of $[\mathbf{D}_0]$. This transition is the reverse of what we have done to move from $\mathbf{H}_{1,i,0}$ to $\mathbf{H}_{1,i,1}$. We proceed in two steps:

- We first switch the j-th vector $[\mathbf{u}_j]$ computed by the encryption oracle to a vector in the span of $[(\mathbf{D}|\mathbf{U})]$, where $\mathbf{U}$ is uniform over $\mathbb{Z}_q^{n\times d}$, if the $(i+1)$-th bit of the binary representation of j is equal to 1.
- Then, we switch the j-th vector $[\mathbf{u}_j]$ computed by the encryption oracle to a vector in the span of $[\mathbf{D}_0]$.

Altogether we obtain and adversary $\mathcal{C}$ such that:

$$|\epsilon_{\mathbf{H}_{1,i+1,0}} - \epsilon_{\mathbf{H}_{1,i,6}}| \leq 2(n-d)\mathbf{Adv}^{\mathtt{mddh}}_{\mathbb{G}_1,\mathcal{D}_{n,d},\mathcal{C}}(\lambda) + \frac{2}{q-1}.$$

It is easy to see that $\epsilon_{\mathbf{H}_2} = \epsilon_{\mathbf{H}_{1,\lceil \log Q_{\mathsf{Enc}} \rceil,6}}$. Next, we prove that $\epsilon_{\mathbf{H}_2} \leq \frac{O(n^2)Q_{\mathsf{Enc}}Q_{\mathsf{Dec}}}{q}$. We reduce the adversary $\mathcal{A}$ playing in $\mathbf{H}_2$ to an (unbounded) adversary $\mathcal{B}$ upon which we can invoke the Lemma 1. We say that $\mathcal{B}$ *forged a valid tuple* if the output of $\mathcal{B}$ matches the event described in the lemma. For any assignments of the vector $\mathbf{a}$ and of the matrix $\mathbf{D}$ in the support of $\mathcal{D}_{n,d}$, we can consider in the Lemma 1 the matrix $\mathbf{E}$ to be set equal to $\mathbf{D}^*$.

Claim. $\Pr[\mathbf{H}_2 = 1] \leq \frac{O(n^2)Q_{\mathsf{Enc}}Q_{\mathsf{Dec}}}{q}$.

Let $(\mathbf{D}, \mathbf{D}^*, \mathbf{D}^\top\mathbf{f}, \mathbf{D}^\top\mathbf{F}, \mathbf{F}\mathbf{D}^*)$ be the tuple received by $\mathcal{B}$ from the challenger. The adversary $\mathcal{B}$ samples uniformly random values $(\bar{\mathbf{f}}, \bar{\mathbf{F}})$ such that $\bar{\mathbf{f}}^\top\mathbf{D} = \mathbf{f}^\top\mathbf{D}$, $\bar{\mathbf{F}}^\top\mathbf{D} = \mathbf{F}^\top\mathbf{D}$ and $\bar{\mathbf{F}}\mathbf{D}^* = \mathbf{F}\mathbf{D}^*$. We can think of the tuple $(\bar{\mathbf{f}}, \bar{\mathbf{F}})$ as a "fake" proving key that matches the verification key given by the challenger. Given $\mathbf{D}$ and $\mathbf{a}$, the reduction $\mathcal{B}$ can sample all the secret material needed to simulate the hybrid $\mathbf{H}_2$. In particular, it can compute the proving key and verification key of the proof system $\mathbf{PS}$ and sample the challenge bit. The reduction $\mathcal{B}$ samples an index value $j^*_{\mathsf{Enc}} \in [Q_{\mathsf{Enc}}]$ and an index $j^*_{\mathsf{Dec}} \in [Q_{\mathsf{Dec}}]$. (Recall that Q_{Enc} and Q_{Dec} denote the number of encryption, resp. decryption queries made by $\mathcal{A}$.) At the j-th query to the encryption oracle:

- If $j \neq j^*_{\mathsf{Enc}}$, the reduction $\mathcal{B}$ generates $\mathbf{x}_j$ following the prescribed algorithms. Then, it computes $y_j \leftarrow \left(\left(\bar{\mathbf{f}} + \bar{\mathbf{F}}\mathbf{x}_j\right)^\top + P(j)\mathbf{D}^\perp\right)\mathbf{u}_j$, where we recall that $P(\cdot)$ is a random function.
- Else, for $j = j^*_{\mathsf{Enc}}$, $\mathcal{B}$ computes $\mathbf{x}_j$ as prescribed, queries its own oracle with $\mathbf{x}_j$ and obtains a value $\mathbf{v} = \mathbf{f} + \mathbf{F}\cdot\mathbf{x}_j$, then, it uses $\mathbf{v} + P(j)\mathbf{D}^\perp$ to compute the proof y, associated with $\mathbf{u}_j$, namely: $y_j \leftarrow \left(\mathbf{v}^\top + P(j)\mathbf{D}^\perp\right)\mathbf{u}_j$.

At the j-th query to decryption oracle with ciphertext $\mathtt{C} = ([\mathbf{x}], [y]_T, \pi)$ there are three possible cases. The easiest case to handle is if $\mathbf{u} \in span(\mathbf{D})$ or $\exists j \neq j^*_{\mathsf{Enc}}$ such that $\mathbf{D}^{*\perp}\mathbf{x}_j = \mathbf{D}^{*\perp}\mathbf{x}$. The reduction $\mathcal{B}$ can compute the consistency check using the keys $\bar{\mathbf{f}}, \bar{\mathbf{F}}$ and the random function P.

The second case is when $\mathbf{D}^{*\perp}\mathbf{x}_{j^*_{\mathsf{Enc}}} = \mathbf{D}^{*\perp}\mathbf{x}$, in this case let $\mathbf{r}'$ be such that $\mathbf{x} - \mathbf{x}_{j^*_{\mathsf{Enc}}} = \mathbf{D}^*\mathbf{r}'$ and compute

$$y' \leftarrow y_{j^*_{\mathsf{Enc}}} + \mathbf{f}^\top \mathbf{D}\mathbf{r}' + \mathbf{x}_{j^*_{\mathsf{Enc}}}^\top \mathbf{F}^\top \mathbf{D}\mathbf{r}' + (\mathbf{F}\mathbf{D}^*\mathbf{r}')^\top (\mathbf{u}_{j^*_{\mathsf{Enc}}} + \mathbf{D}\mathbf{r}')$$

namely, compute the element $[y']_T$ as if it was computed in the re-randomization of the ciphertext $\mathtt{C}_{j^*_{\mathsf{Enc}}}$ using randomness $\mathbf{r}'$. Notice that, by definition of $\mathbf{H}_2$ the consistency check for $[y]_T$ would be computed by checking if

$$y \stackrel{?}{=} \left((\mathbf{f} + \mathbf{F}\mathbf{x})^\top + P(j^*_{\mathsf{Enc}})\mathbf{D}^\perp\right)\mathbf{u}.$$

By Lemma 2 and by definition of $y_{j^*_{\mathsf{Enc}}}$, the two checks are equivalent. The last case is when $\mathbf{u} \notin \mathit{span}(\mathbf{D}) \wedge \forall j : \mathbf{D}^{*\perp}\mathbf{x}_j \neq \mathbf{D}^{*\perp}\mathbf{x}$, i.e., the query might be "critical":

- If $j < j^*_{\mathsf{Dec}}$ then return $\perp$ to the adversary $\mathcal{A}$, in this case we assume that the query was not critical and that the decryption would fail.
- If $j = j^*_{\mathsf{Dec}}$ then output the tuple $(y - P(j^*_{\mathsf{Enc}})\mathbf{D}^\perp\mathbf{u}, \mathbf{u}, \mathbf{x})$ as the forgery of $\mathcal{B}$.

We condition on the event that j^*_{Dec} is the first critical query of $\mathcal{A}$ and that, let the ciphertext sent by $\mathcal{A}$ at the j^*_{Dec} query be $\mathtt{C} = ([\mathbf{x}], [y]_T, \pi)$ we have that the equation $[y]_T = (\mathbf{f} + P(j^*_{\mathsf{Enc}})\mathbf{D}^\perp + \mathbf{F}\mathbf{x})^\top[\mathbf{u}]$ holds. Let $\mathtt{Guess}$ be such event. Conditioned on such a lucky event, $\mathcal{B}$ indeed produces a valid forgery, in fact by the definition of a critical query $(\mathbf{x}_{j^*_{\mathsf{Enc}}} - \mathbf{x}) \notin \mathit{span}(\mathbf{D}^*)$ and $\mathbf{u} \notin \mathit{span}(\mathbf{D})$.

We show that the view provided by $\mathcal{B}$ to the adversary $\mathcal{A}$ up to the j^*_{Dec}-th decryption query and conditioned on $\mathtt{Guess}$ is equivalent to the view of the adversary up to the j^*_{Dec}-th decryption query in the hybrid game $\mathbf{H}_2$. The intuition is that the values $P(j)\mathbf{D}^\perp$, for all j, mask the components of $(\mathbf{f}, \mathbf{F})$ and $(\bar{\mathbf{f}}, \bar{\mathbf{F}})$ that differ. Indeed, we know that for some row vectors $\mathbf{v}, \mathbf{w}, \mathbf{w}'$, it holds that $\mathbf{f} = \mathbf{D}\mathbf{v} + (\mathbf{w}\mathbf{D}^\perp)^\top$ and $\bar{\mathbf{f}} = \mathbf{D}\mathbf{v} + (\mathbf{w}'\mathbf{D}^\perp)^\top$. Similarly, for some $\mathbf{V}, \mathbf{W}$ and $\mathbf{W}'$, $\mathbf{F} = \mathbf{D}\mathbf{V} + (\mathbf{W}\mathbf{D}^\perp)^\top$, and $\bar{\mathbf{F}} = \mathbf{D}\mathbf{V} + (\mathbf{W}'\mathbf{D}^\perp)^\top$.

Let P' be a uniformly random function, and consider the following function:

$$P(j) = \begin{cases} P'(j), & j = j^*_{\mathsf{Enc}} \\ P'(j) + \Delta_j, & j \neq j^*_{\mathsf{Enc}} \end{cases}$$

where $\Delta_j = \mathbf{w} - \mathbf{w}' + \mathbf{x}_j^\top(\mathbf{W} - \mathbf{W}')$. It is not hard to see that P is a uniformly random function. Now consider the mental experiment where $\mathcal{B}$ runs the same but using the random function P defined above. Since P is uniformly random, the probability that $\mathcal{B}$ forges a valid tuple in this mental experiment is the same as the probability that $\mathcal{B}$ forges a valid tuple in the real experiment. Also, for any $j \neq j^*_{\mathsf{Enc}}$ the value y computed at the j-th encryption oracle query is:

$$\begin{aligned} y &= \left((\bar{\mathbf{f}} + \bar{\mathbf{F}}\mathbf{x}_j)^\top + P(j)\mathbf{D}^\perp\right)[\mathbf{u}_j] = \left((\bar{\mathbf{f}} + \bar{\mathbf{F}}\mathbf{x}_j)^\top + (P'(j) + \Delta_j)\mathbf{D}^\perp\right)[\mathbf{u}_j] = \\ &= \left(\left(\bar{\mathbf{f}} + ((\mathbf{w} - \mathbf{w}')\mathbf{D}^\perp)^\top + (\bar{\mathbf{F}} + ((\mathbf{W} - \mathbf{W}')\mathbf{D}^\perp)^\top)\mathbf{x}_j\right)^\top + P'(j)\mathbf{D}^\perp\right)[\mathbf{u}_j] = \\ &= \left((\mathbf{f} + \mathbf{F}\mathbf{x}_j)^\top + P'(j)\mathbf{D}^\perp\right)[\mathbf{u}_j]. \end{aligned}$$

The probability that the reduction $\mathcal{B}$ creates a forgery is $\Pr[\mathbf{H}_2 = 1 \wedge \mathtt{Guess}]$, and the two events are independent. Moreover, since $\Pr[\mathtt{Guess}] = (Q_{\mathsf{Enc}} Q_{\mathsf{Dec}})^{-1}$, by the Rand-RCCA Lemma in [14] we have that $\Pr[\mathbf{H}_2 = 1] \leq \frac{n(n+1)Q_{\mathsf{Enc}}Q_{\mathsf{Dec}}}{q}$.

5.1 Publicly-Verifiable Rand-RCCA PKE

We show two publicly verifiable Rand-RCCA PKE schemes based on the scheme from Section 5. Following the ideas in [13], we append a malleable NIZK proof (essentially a Groth-Sahai proof) that $[y]_T$ and π are well-formed to the ciphertexts of $\mathcal{PKE}$ from the previous section. The decryption algorithm outputs the decrypted message only if the NIZK proofs are valid. Public verifiability follows because the NIZK proofs can be verified using the public parameters.

Let $\mathcal{PKE}_1 = (\mathsf{KGen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1, \mathsf{Rand}_1)$ be the scheme of Section 5 instantiated using the benign proof system of Section 3.1, and let $\mathbf{PS}_2 = (\mathsf{PGen}_2, \mathsf{PPrv}_2, \mathsf{PVer}_2)$ and PEvl_2 form a malleable NIZK system for membership in the relation

$$\mathcal{R}_2 = \left\{ (\mathsf{pk}, [\mathbf{x}]), ([y]_T, \pi, \mathbf{r}) : \begin{array}{c} y = \mathbf{f}^\top \mathbf{u} + \mathbf{x}^\top \mathbf{F} \mathbf{u} \\ \mathsf{PPrv}_1(ppk, [\mathbf{u}], \mathbf{r}) = \pi \end{array} \right\},$$

and where the allowable set of transformations contains all the transformations $(T_{\mathsf{el}}, T_{\mathsf{wit}})$ such that it exists $\hat{\mathbf{r}}$ with $T_{\mathsf{el}}(\mathsf{pk}, [\mathbf{x}]) = \mathsf{pk}, [\hat{\mathbf{x}}]$, $T_{\mathsf{wit}}([y]_T, \pi, \mathbf{r}) = [\hat{y}]_T, \hat{\mathsf{pk}}, \mathbf{r} + \hat{\mathbf{r}}$ and $([\hat{\mathbf{x}}], [\hat{y}]_T, \hat{\pi}) = \mathsf{Rand}_1(\mathsf{pk}, ([\mathbf{x}], [y]_T, \pi); \hat{\mathbf{r}})$; each transformation in the set of allowable transformation is uniquely identified by a vector $\hat{\mathbf{r}}$.

The pv-Rand-PKE scheme $\mathcal{PKE}_2 = (\mathsf{Init}, \mathsf{KGen}_2, \mathsf{Enc}_2, \mathsf{Dec}_2, \mathsf{Rand}_2, \mathsf{Ver})$ is identical to $\mathcal{PKE}_1$, except that (i) KGen_2 additionally samples the common reference string for $\mathbf{PS}_2$, (ii) the encryption procedure computes a ciphertext as in $\mathcal{PKE}_1$ but additionally computes a proof π_2 for $\mathbf{PS}_2$ and outputs a ciphertext $\mathtt{C} = ([\mathbf{x}_1], \pi_2)$, (iii) the decryption procedure first checks the proof π_2 holds w.r.t. the instance $(\mathsf{pk}, [\mathbf{x}])$ and, if so, it outputs $\mathtt{M} = (-\mathbf{a}^\top, 1)[\mathbf{x}]$ (and $\perp$ otherwise), (iv) the re-randomization procedure randomizes $[\mathbf{x}]$ as in $\mathcal{PKE}_1$ and uses PEvl_2 for the remaining part of the ciphertext, and (v) Ver_2 simply checks π_2.

Theorem 4. *If $\mathbf{PS}_2$ is adaptively sound, $(\epsilon, O(T))$-composable zero-knowledge, and perfect derivation private, and $\mathcal{PKE}_1$ is mRCCA secure then $\mathcal{PKE}_2$ is publicly verifiable, perfectly re-randomizable, and mRCCA-secure. Specifically, for any PPT $\mathcal{A}$ making up to Q_{Enc} encryption queries and Q_{Dec} decryption queries and with running time T exist PPT $\mathcal{B}^{\mathrm{rcca}}$ making the same number of queries and adversaries $\mathcal{B}^{\mathrm{snd}}, \mathcal{B}^{\mathrm{zk}}$ with similar running times*

$$\mathbf{Adv}^{\mathtt{mRCCA}}_{\mathcal{A},\mathcal{PKE}_2}(\lambda) \leq \mathbf{Adv}^{\mathtt{mRCCA}}_{\mathcal{B}^{\mathrm{rcca}},\mathcal{PKE}_1}(\lambda) + \mathbf{Adv}^{\mathtt{snd}}_{\mathcal{B}^{\mathrm{snd}},\mathbf{PS}_2}(\lambda) + \epsilon$$

The proof follows by inspection of the proof of Theorem 2 in [13]. In more detail, their proof proceeds in two steps. First, it reduces to the adaptive soundness of the NIZK proof system to claim that if a *publicly-verifiable* ciphertext decrypts correctly then its respective *non-publicly verifiable* ciphertext should decrypt correctly too. We notice that this step can be performed tightly relying either

on statistical adaptive soundness of the proof system or relying on the computational soundness of the proof system when the language proved is witness samplable. The reason is that the reduction can check which one of the many NIZK-proofs from the adversary breaks adaptive soundness before submitting it as its forgery. The second step uses composable zero-knowledge to first tightly switch the way the public parameters are generated and then to switch (all together) the proofs for the ciphertexts from real to simulated.

To instantiate the malleable NIZK, we consider a construction along the same line of [13]. In more detail, [13] introduced an extension of the Groth-Sahai proof system that is zero-knowledge even for pairing product equations where the $\mathbb{G}_T$-elements are variables. Their idea is to commit the elements in $\mathbb{G}_T$ using a commitment scheme with nice bilinear properties. Groth-Sahai proofs can be instantiated under any $\mathcal{D}_k$-MDDH Assumption [11] and, given their nice algebraic properties they are malleable [8]. More details are given in [14].

A more efficient tight-secure pv-Rand-RCCA PKE. To facilitate our more efficient scheme, we introduce a stronger variant of the MDDH assumption (cf. Definition 1) in which the adversary gets not only a matrix $[\mathbf{A}]$, but also the tensor product $[\mathbf{A} \otimes \mathbf{A}]$ to distinguish an element from $span([\mathbf{A}])$ and random:

Definition 9 (Tensor Matrix Diffie-Hellman assumption in $\mathbb{G}_\gamma$). *The $\mathcal{D}_{\ell,k}$-Tensor-Matrix-Decisional-Diffie-Hellman (TMDDH) assumption in group $\mathbb{G}_\gamma$ holds if for all non-uniform PPT adversaries $\mathcal{A}$,*

$$|\Pr\left[\mathcal{A}(\mathcal{G}, [\mathbf{A} \otimes \mathbf{A}]_\gamma, [\mathbf{A}]_\gamma, [\mathbf{A}\mathbf{w}]_\gamma) = 1\right] - \Pr\left[\mathcal{A}(\mathcal{G}, [\mathbf{A} \otimes \mathbf{A}]_\gamma, [\mathbf{A}]_\gamma, [\mathbf{z}]_\gamma) = 1\right]|$$

is negligible, where the probability is taken over $\mathcal{G} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \mathcal{P}_1, \mathcal{P}_2) \leftarrow \mathsf{GGen}(1^\lambda)$, $\mathbf{A} \leftarrow \mathcal{D}_{\ell,k}, \mathbf{w} \leftarrow \mathbb{Z}_q^k$, $[\mathbf{z}]_\gamma \leftarrow \mathbb{G}_\gamma^\ell$, and the coin tosses of adversary $\mathcal{A}$.

The TMDDH assumption can be seen as a generalization of the "square-Diffie-Hellman" assumption [6,29], and as a special case of the "Uber assumption family" [5]. Since a TMDDH adversary gets quadratic terms $[\mathbf{A} \otimes \mathbf{A}]$ "in the exponent", it is not clear how this assumption relates to the more standard MDDH assumption. However, we remark that the TMDDH assumption holds generically for large enough dimensions, at least for uniformly random $\mathbf{A}$.

Lemma 4 (Generic security of TMDDH). *For $k \geq 4$, the $\mathcal{U}_{k+1,k}$-TMDDH assumption holds against generic adversaries in a symmetric pairing setting.*

In [14] we explain what we mean by "holds generically" according to the formulation of Maurer [28] and we sketch a proof of the lemma.

The idea of the second publicly-verifiable PKE scheme is to (1) add in the public key the values $\mathbf{k}^\top[\mathbf{D} \otimes \mathbf{D}]$ and (2) use a malleable proof system $\mathbf{PS}_3$ for membership in the relation

$$\mathcal{R}_3 = \left\{ (\mathsf{pk}, [\mathbf{x}]), ([y]_T, \pi, \mathbf{r}) : \begin{array}{c} y = \mathbf{f}^\top\mathbf{u} + \mathbf{x}^\top\mathbf{F}\mathbf{u} \\ \mathbf{k}^\top[\mathbf{D} \otimes \mathbf{D}]\mathbf{r} \otimes \mathbf{r} \cdot [1] = \pi \end{array} \right\},$$

with the same set of allowable transformations as in the previous publicly verifiable PKE scheme. The languages associated with the relation $\mathcal{R}_3$ and $\mathcal{R}_2$ are identical, but we can obtain a more efficient NIZK proof for $\mathcal{R}_3$.

Theorem 5. *The pv-Rand-PKE scheme $\mathcal{PKE}_3$ is publicly verifiable, perfectly re-randomizable and RCCA-secure. Specifically:*

$$\begin{aligned}\mathbf{Adv}^{\mathtt{RCCA}}_{\mathcal{A},\mathcal{PKE}}(\lambda) \leq & \mathbf{Adv}^{\mathtt{TMDDH}}_{\mathbb{G}_1,\mathcal{U}_{n,d},\mathcal{B}}(\lambda) + \ O\left(d \log Q_{\mathsf{Enc}}\right) \cdot \mathbf{Adv}^{\mathtt{MDDH}}_{\mathbb{G}_1,\mathcal{U}_{n,d},\mathcal{B}'}(\lambda) \\ & + \log Q_{\mathsf{Enc}} \cdot \mathbf{Adv}^{\mathtt{snd}}_{\mathcal{B}'',\mathbf{PS}}(\lambda) + O\left(\tfrac{n^2 Q_{\mathsf{Dec}} Q_{\mathsf{Enc}} \log Q_{\mathsf{Enc}}}{q}\right)\end{aligned}$$

We only sketch the proof, which is only a slight variation of the proof of Theorem 3. Notice that in the proof of Theorem 3 to move from $\mathbf{G}_3$ to $\mathbf{G}_4$ we use the $\mathcal{D}_{n,d}$-MDDH assumption. This step changes with our modified scheme, since we add $[\mathbf{D} \otimes \mathbf{D}]$ to the public key. We thus need to rely on the stronger TMDDH assumption. Also notice that this is the only step in the proof of Theorem 3 where the assumption over the matrix $[\mathbf{D}]$ is used. Finally, observe that we can prove both composable zero-knowledge and computational adaptive soundness of the NIZK proof system for $\mathcal{R}_3$ using the classical $\mathcal{D}_k$-MDDH assumption.

6 Application: Universally Composable MixNet

We can plug-and-play our pv-Rand-RCCA PKE schemes in the MixNet protocol of [13] because their protocol works for any pv-Rand RCCA scheme that has the property of being *linear* and a property that holds for both $\mathcal{PKE}_2$ and $\mathcal{PKE}_3$. For space reasons, we defer the details in [14].

The MixNet ideal functionality interacts with n sender parties and m mixer parties. The i-th sender sends the message $\mathtt{M}_i$, while the mixer can decide to mix the messages. At the end, when all the mixer have sent their inputs, the functionality returns the list of sorted messages. For space reasons, the ideal functionality is formally defined in [14].

The protocol is divided into 3 phases: (i) at the input phase, the *sender parties* send pv-Rand-RCCA ciphertexts of their messages and a simulation-extractable[7] NIZK of knowledge; (ii) at the mixing phase, the mixers, one after the other, shuffle the ciphertexts and compute the so-called *check-sum* NIZK proofs that paired with the public-verifiability and the RCCA property are sufficient to prove the validity of the shuffles; (iii) at the output phase, the ciphertexts are decrypted. The nice feature of the protocol is that the statements proved by the *check-sum* proofs are of constant size, independent of the number of shuffled ciphertexts.

The NIZK proofs employed in the input-submission phase are needed only to make sure independence of the inputs. We notice that to obtain our "tightly-secure" MixNet we need only to make sure that the Rand-mRCCA PKE and the simulation-extractable NIZK proofs are tightly secure. Let $\mathbf{Adv}^{\mathtt{sim-ext}}_{\mathcal{A},\mathbf{PS}}(\lambda)$ be the advantage of an adversary $\mathcal{A}$ against the simulation extractability experiment for $\mathbf{PS}$, we are ready now to state the main contribution of this section.

[7] Actually, they need a weaker form of soundness called all-but-one soundness, however simulation extractability is sufficient.

Theorem 6. *Let $\mathcal{PKE}$ be a linear pv-Rand RCCA PKE,* **PS** *be a simulation-extractable NIZK, and let Π be the MixNet protocol from [13] instantiated with $\mathcal{PKE}$ and* **PS**. *The protocol Π* realizes *$\mathcal{F}_{\mathsf{Mix}}$ with setup assumptions a threshold decryption functionality $\mathcal{F}_{\mathsf{TDec}}[\mathcal{PKE}]$ and a common-reference string functionality $\mathcal{F}_{\mathsf{CRS}}$. More in detail, there exist a simulator S and negligible function $\mathtt{negl}(\lambda, m)$ such that for any static-corruption environment $\mathcal{Z}$ with running time $T_{\mathcal{Z}}$ there exist an adversaries $\mathcal{B}, \mathcal{B}'$ whose running time is $O(T_{\mathcal{Z}}(\lambda))$, such that:*

$$\begin{aligned}|\Pr\left[\textsc{Real}_{\mathcal{Z},\Pi}(\lambda)=1\right] - \Pr\left[\{\mathcal{F}_{\mathsf{CRS}},\mathcal{F}_{\mathsf{TDec}}\}\text{-}\textsc{Hybrid}^{\mathcal{F}_{\mathsf{Mix}}}_{\mathcal{Z},\mathsf{S}}(\lambda)=1\right]| \\ \leq 3\mathbf{Adv}^{\mathtt{mRCCA}}_{\mathcal{B},\mathcal{PKE}}(\lambda) + \mathbf{Adv}^{\mathtt{sim-ext}}_{\mathcal{B}',\mathbf{PS}'}(\lambda) + \mathtt{negl}(\lambda,m)\end{aligned}$$

We stress that the function $\mathtt{negl}(\lambda, m)$ in the statement of Theorem 6 is independent of $T_{\mathcal{Z}}$ and only depends on the number of mixers (which we can think as a small number). The proof of the theorem follows by inspection of the proof of Theorem 5 in [13] and observing that the three steps of the proof that reduce to the pv-Rand-RCCA security of $\mathcal{PKE}$ can be performed tightly by relying on the multi-ciphertext RCCA security definition (cf. Definition 8). In [14] we give more details and we show how to instantiate the necessary simulation-extractable NIZK using the tightly-secure QA-NIZK based on the MDDH assumption of Abe *et al.* [2]. Thus, instantiating the protocol with $\mathcal{PKE}_2$ (resp. $\mathcal{PKE}_3$) we obtain a MixNet protocol that reduces almost-tightly in the number of mixed messages to the MDDH (resp. TMDDH) Assumption.

References

1. M. Abdalla, F. Benhamouda, and D. Pointcheval. Disjunctions for hash proof systems: New constructions and applications. In *EUROCRYPT 2015, Part II*, 2015.
2. M. Abe, C. S. Jutla, M. Ohkubo, and A. Roy. Improved (almost) tightly-secure simulation-sound QA-NIZK with applications. In *ASIACRYPT 2018, Part I*, 2018.
3. S. Bayer and J. Groth. Efficient zero-knowledge argument for correctness of a shuffle. In *EUROCRYPT 2012*, 2012.
4. M. Bellare, A. Boldyreva, and S. Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In *EUROCRYPT 2000*, 2000.
5. X. Boyen. The uber-assumption family (invited talk). In *PAIRING 2008*, 2008.
6. M. Burmester, Y. Desmedt, and J. Seberry. Equitable key escrow with limited time span (or, how to enforce time expiration cryptographically). In *ASIACRYPT'98*, 1998.
7. R. Canetti, H. Krawczyk, and J. B. Nielsen. Relaxing chosen-ciphertext security. In *CRYPTO 2003*, 2003.
8. M. Chase, M. Kohlweiss, A. Lysyanskaya, and S. Meiklejohn. Malleable proof systems and applications. In *EUROCRYPT 2012*, 2012.
9. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO'98*, 1998.
10. Y. Dodis, I. Mironov, and N. Stephens-Davidowitz. Message transmission with reverse firewalls—secure communication on corrupted machines. In *CRYPTO 2016*, 2016.

11. A. Escala, G. Herold, E. Kiltz, C. Ràfols, and J. Villar. An algebraic framework for Diffie-Hellman assumptions. In *CRYPTO 2013, Part II*, 2013.
12. A. Faonio and D. Fiore. Improving the efficiency of re-randomizable and replayable CCA secure public key encryption. In *ACNS 20, Part I*, 2020.
13. A. Faonio, D. Fiore, J. Herranz, and C. Ràfols. Structure-preserving and re-randomizable RCCA-secure public key encryption and its applications. In *ASIACRYPT 2019, Part III*, 2019.
14. A. Faonio, D. Hofheinz, and L. Russo. Almost tightly-secure re-randomizable and replayable CCA-secure public key encryption. Cryptology ePrint Archive, Paper 2023/152, 2023. https://eprint.iacr.org/2023/152.
15. A. Faonio and L. Russo. Mix-nets from re-randomizable and replayable CCA-secure public-key encryption. In *Security and Cryptography for Networks*, 2022.
16. P. Fauzi, H. Lipmaa, J. Siim, and M. Zajac. An efficient pairing-based shuffle argument. In *ASIACRYPT 2017, Part II*, 2017.
17. R. Gay, D. Hofheinz, E. Kiltz, and H. Wee. Tightly CCA-secure encryption without pairings. In *EUROCRYPT 2016, Part I*, 2016.
18. R. Gay, D. Hofheinz, and L. Kohl. Kurosawa-desmedt meets tight security. In *CRYPTO 2017, Part III*, 2017.
19. R. Gay, D. Hofheinz, L. Kohl, and J. Pan. More efficient (almost) tightly secure structure-preserving signatures. In *EUROCRYPT 2018, Part II*, 2018.
20. J. Groth. Rerandomizable and replayable adaptive chosen ciphertext attack secure cryptosystems. In *TCC 2004*, 2004.
21. S. Han, S. Liu, L. Lyu, and D. Gu. Tight leakage-resilient CCA-security from quasi-adaptive hash proof system. In *CRYPTO 2019, Part II*, 2019.
22. D. Hofheinz. Adaptive partitioning. In *EUROCRYPT 2017, Part III*, 2017.
23. C. S. Jutla and A. Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In *ASIACRYPT 2013, Part I*, 2013.
24. K. Kurosawa and Y. Desmedt. A new paradigm of hybrid encryption scheme. In *CRYPTO 2004*, 2004.
25. B. Libert, M. Joye, M. Yung, and T. Peters. Concise multi-challenge CCA-secure encryption and signatures with almost tight security. In *ASIACRYPT 2014*, 2014.
26. B. Libert, T. Peters, M. Joye, and M. Yung. Compactly hiding linear spans - tightly secure constant-size simulation-sound QA-NIZK proofs and applications. In *ASIACRYPT 2015, Part I*, 2015.
27. B. Libert, T. Peters, and C. Qian. Structure-preserving chosen-ciphertext security with shorter verifiable ciphertexts. In *PKC 2017, Part I*, 2017.
28. U. M. Maurer. Abstract models of computation in cryptography (invited paper). In *10th IMA International Conference on Cryptography and Coding*, 2005.
29. U. M. Maurer and S. Wolf. Diffie-Hellman oracles. In *CRYPTO'96*, 1996.
30. M. Naveed, S. Agrawal, M. Prabhakaran, X. Wang, E. Ayday, J.-P. Hubaux, and C. A. Gunter. Controlled functional encryption. In *ACM CCS 2014*, 2014.
31. O. Pereira and R. L. Rivest. Marked mix-nets. In *FC 2017 Workshops*, 2017.
32. M. Prabhakaran and M. Rosulek. Rerandomizable RCCA encryption. In *CRYPTO 2007*, 2007.
33. C. Rackoff and D. R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *CRYPTO'91*, 1991.
34. R. E. Tarjan. Efficiency of a good but not linear set union algorithm. In *Journal of the ACM*, 1975.
35. Y. Wang, R. Chen, G. Yang, X. Huang, B. Wang, and M. Yung. Receiver-anonymity in rerandomizable RCCA-secure cryptosystems resolved. In *CRYPTO 2021, Part IV*, 2021.