

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

Antonio Faonio¹ , Dennis Hofheinz², and Luigi Russo¹ 

¹ EURECOM, Sophia Antipolis, France {faonio, russo1}@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 [RS92], 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 [CKN03] 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 [Gro04] 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 which decrypts to the same message. Such a re-randomization procedure opens the door for applications that require secure communication *and* anonymity. For instance, PKE schemes that are re-randomizable and RCCA-secure enable anonymous and secure message transmissions (see Prabhakaran and Rosulek [PR07]), MixNets (see Faonio *et al.* [FFHR19] and Pereira and Rivest [PR17]), Controlled

Functional Encryption (see Naveed *et al.* [NAP⁺14]), and one-round message-transmission protocols with reverse firewalls (see Dodis, Mironov, and Stephens-Davidowitz [DMS16]).

Tight Security. Yet another criticism to 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 amount 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 [BBM00] created a fruitful area of research which 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 proved to have tight security in the multi-ciphertext and multi-user setting: some notable examples are the works of [GHKW16, GHK17, HLLG19, Hof17, LJYP14, LPJY15]. However, tight security in the context of Rand-RCCA security has not been studied, although in particular the above Rand-RCCA use cases feature a large number of ciphertexts or users.

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 (almost³) tightly reduces to the \mathcal{D}_d -MDHH 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 [FFHR19]. 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 (M_0, 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:

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.

- an encryption oracle that, on input a pair of messages M_0, M_1 , returns some valid encryption of M_b where b is the challenge bit,
- and a guarded decryption oracle that, on input a ciphertext C , returns a message M , or the special indexed symbol \diamond_j if 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 (M_1, M_2) , and receives back C_A ; (ii) sends (M_2, M_3) , and receives back C_B ; (iii) sends (M_3, M_1) , and receives back C_C . \mathcal{A} now queries the decryption oracle with 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 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 $\{M_0, M_1\}$, otherwise with a message $M' \notin \{M_0, M_1\}$. 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 [CKN03] is tightly implied by our mRCCA security notion. In a nutshell, we show that the (computational variational) distance between the view in the ideal world and in the real world is bounded by the advantage of an adversary with the same computational resources as the environment in the multi-ciphertext RCCA security game. We elaborate on the details in Section 4.

A Tightly-Secure Rand-mRCCA PKE scheme. Our starting points are the recent work of Faonio *et al.* [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 [Hof17] and Gay *et al.* [GHKP18].

Very briefly, the main idea of our construction is to encrypt the message similarly to [FFHR19], and additionally 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 [Hof17], 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 [AJOR18,GHK17,GHKP18,Hof17]) are re-randomizable. To solve this problem, we introduce a new malleable proof system based on the work of Abdalla, Benhamouda and Pointcheval [ABP15], with the necessary security guarantees.

The second (and more challenging) technical problem is that we need to reconcile the adaptive partitioning technique with the Rand-RCCA technique of [FFHR19]. 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 super spaces of \mathbf{D}^* . Thanks to this choice, we can recognize the challenge ciphertexts after re-randomization by multiplying the de-

encrypted ciphertext by a matrix orthogonal⁴ to \mathbf{D}^* . 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.* [GHKP18], where secret keys are randomized, instead of the original strategy of Hofheinz [Hof17], where ciphertexts are randomized. Finally, the original adaptive partitioning strategy relies on the pairwise universality of a hash proof system [CS98] 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 [FFHR19]. 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-made 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-made 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 (an univocally linked) secret key; on the other hand, other kind of decryption queries can be safely decrypted with any of the secret keys. This rule allows eventually to use the lemma of [FFHR19], 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 [FFHR19] we show that our Rand-mRCCA PKE can be used to instantiate a PKE with the nice property of public-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 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 [FFHR19]. Instantiating such protocol with our schemes, we obtain an (almost) “*tightly-secure*” MixNet

⁴ 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 rerandomized) ciphertext with one of the challenge ciphertexts.

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 [BG12] 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 bilinear-pairing setting of Fauzi *et al.* [FLSZ17] 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 [PR07] introduced the first Rand-RCCA PKE in the standard model. Abstracting the scheme of [PR07], and solving a long-standing open problem, recently Wang *et al.* [WCY⁺21] introduced the first receiver-anonymous Rand-RCCA PKE. Faonio and Fiore [FF20] 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.* [CKLM12], the recent works of Libert, Peters and Qian [LPQ17], and of Faonio *et al.* [FFHR19]. In Table 1 we offer a comparison, in terms of security properties and functionalities, of our schemes of Section 5, i.e. $\mathcal{PK}\mathcal{E}_1$, $\mathcal{PK}\mathcal{E}_2$ and $\mathcal{PK}\mathcal{E}_3$, and the previous schemes. From a technical point of view, our schemes inherit from the scheme of [FFHR19], 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 Section 1.3 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 $\mathcal{PK}\mathcal{E}_1$ and $\mathcal{PK}\mathcal{E}_2$ 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 $\mathcal{PK}\mathcal{E}_3$ 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 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 $\mathcal{PK}\mathcal{E}_2$ and $\mathcal{PK}\mathcal{E}_3$ 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 $\mathcal{PK}\mathcal{E}_2$ (resp. $\mathcal{PK}\mathcal{E}_3$) under $\mathcal{D}_{6,2}$ -MDDH (resp. $\mathcal{U}_{9,4}$ -TMDDH) assumption from [EHK⁺13] and [FFHR19].

We note that $\mathcal{PK}\mathcal{E}_2$ and $\mathcal{PK}\mathcal{E}_3$ are far from being considered practical, while $\mathcal{PK}\mathcal{E}_1$ is considerably less efficient than [FFHR19]. 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 [GHKW16, GHK17])

PKE	Group Setting	Assumption	Struc. Pres.	Pub. Ver.	Tight
[CKLM12], [LPQ17]	Type-3	SXDH	✓	✓	
[FFHR19]	Type-3	$\mathcal{D}_{d+1,d}$ -MDDH	✓*	✓	
$\mathcal{PK}\mathcal{E}_1$	Type-1	$\mathcal{D}_{n,d}$ -MDDH	✓*		✓
$\mathcal{PK}\mathcal{E}_2$	Type-1	$\mathcal{D}_{n,d}$ -MDDH	✓*	✓	✓
$\mathcal{PK}\mathcal{E}_3$	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	C	Enc \approx Rand	Dec
[FFHR19] (1)	$3\mathbb{G}_1 + 2\mathbb{G}_2 + \mathbb{G}_T$	$4E_1 + 5E_2 + 2E_T + 5P$	$8E_1 + 4E_2 + 4P$
$\mathcal{PK}\mathcal{E}_1$	$7\mathbb{G}_1 + 2\mathbb{G}_T$	$14E_1 + 2E_T + 14P$	$48E_1 + 36E_T + 49P$
[LPQ17]	$42\mathbb{G}_1 + 20\mathbb{G}_2$	$79E_1 + 64E_2$	$1E_1 + 142P$
[FFHR19] (2)	$14\mathbb{G}_1 + 15\mathbb{G}_2 + 4\mathbb{G}_T$	$36E_1 + 45E_2 + 6E_T + 5P$	$2E_1 + 50P$
$\mathcal{PK}\mathcal{E}_2$	$380\mathbb{G}_1 + 330\mathbb{G}_T$	$\approx 180E_1 + 110E_T + 38P$	$\approx 6E_1 + 400P$
$\mathcal{PK}\mathcal{E}_3$	$105\mathbb{G}_1 + 9\mathbb{G}_T$	$\approx 261E_1 + 9E_T + 16P$	$\approx 6E_1 + 11P$

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 that 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 $\mathcal{PK}\mathcal{E}_1, \mathcal{PK}\mathcal{E}_2$ (DLIN), for $\mathcal{PK}\mathcal{E}_3$ ($\mathcal{U}_{9,4}$ -TMDDH) and for [FFHR19] (SXDH). The costs of the last two schemes are approximated.

have a realistic break-even point⁵. We hope for a similar development with Rand-RCCA PKE schemes.

Our benign proof system uses the “OR-Proof” technique⁶ from [ABP15]. We notice that, in the context of tightly-secure reductions, the same technique from [ABP15] has been used in [HLLG19] to instantiate their (Leakage-Resilient) Ardent Quasi-Adaptive Hash Proof System. We stress that in our work, in contrast with [HLLG19], the main reason to use the technique from [ABP15] is because of its nice linear property that, in turn, allows for *malleable* proof system.

1.3 Open Problems

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 [WCY+21] for Rand-RCCA schemes and [HLLG19] for tightly-secure (LR-)CCA-secure schemes. Recently, Faonio and Russo [FR22]

⁵ For the same security parameter, the work of [GHKW16, GHK17] outperform state-of-the-art non-tightly secure schemes like Kurosawa-Desmedt [KD04] around 2^{30} ciphertexts.

⁶ We compare our malleable benign proof system with their SPHF’s in Appendix A.1.

improved over the mix-net protocol of [FFHR19], 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.

On the Instantiability with Asymmetric Pairings Our construction requires type-1 pairings, which are less efficient than type-3 ones. Hence, it is natural to ask whether we can instantiate our construction also from type-3 pairings. Unfortunately, we do not know how to do so, since it is not clear how to reconcile the adaptive partitioning technique [Hof17] with a Rand-RCCA construction in settings with type-3 pairings (such as the one from [FFHR19]).

In a nutshell, ciphertexts from the Rand-RCCA construction from [FFHR19] carry elements from both source groups \mathbb{G}_1 and \mathbb{G}_2 in the ciphertext (and this also seems like a typical property of encryption schemes in type-3 pairing settings). Hence, ciphertext re-randomization also needs to modify (and “refresh”) elements from \mathbb{G}_1 and \mathbb{G}_2 .

Now if we try to obtain tight security by applying the “adaptive partitioning” strategy to such a scheme, we would add consistency proofs to each ciphertext, essentially stating that all involved elements are chosen from the right joint distribution (or from some related distributions only used during the proof). Since the group elements in the scheme from [FFHR19] are tied together (even across \mathbb{G}_1 and \mathbb{G}_2), these proofs consider statements that involve both groups simultaneously.

This last property can be problematic for our overall proof strategy. Namely, following the adaptive partitioning strategy would mean to gradually inject entropy into the challenge ciphertexts, in our concrete case by introducing challenge-dependent randomness in the secret keys that are used in the decryption oracle and to produce (“ill-formed”) challenge ciphertexts. In case of the scheme from [FFHR19], this gradual randomization affects secret keys that refer to \mathbb{G}_1 , as well as secret keys referring to \mathbb{G}_2 .

The problem with this approach is now that randomizing both types of secret keys requires more entropy, and it is not clear where this entropy should come from. In fact, moving parts of the ciphertext outside of a certain linear space (like we did with our value of \mathbf{u} in our proof of Theorem 3) allows to inject the corresponding secret keys with additional entropy, but this entropy always refers to one single group (\mathbb{G}_1 or \mathbb{G}_2). Reconciling the entropy from both groups into one additional random function (like our function P_i) seems difficult.

In fact, an instantiation based on type-3 pairings seems to require new “entropy management” techniques to tightly randomize many challenge ciphertexts. We leave the construction of a tightly Rand-RCCA secure PKE scheme as an interesting open problem.

2 Preliminaries

A function is negligible in λ if it vanishes faster than the inverse of any polynomial in λ . We write $f(\lambda) \in \text{negl}(\lambda)$ 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 τ_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 , $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ is an efficiently computable, non-degenerate bilinear map. Let GGen be a probabilistic polynomial time 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 $\text{span}(\mathbf{A})$ denote the linear span of the columns of \mathbf{A} . Given a set of vectors \mathbf{V} in some vector space over \mathbb{Z}_q , $\text{span}(\mathbf{V})$ denotes its linear span. $\mathcal{D}_{n,d}$ is a matrix distribution if outputs (in probabilistic polynomial time, with overwhelming probability) matrices in $\mathbb{Z}_q^{n \times d}$.

Tensor Product. We define the transformation T that maps matrices $\mathbf{A} = (\mathbf{a}_1, \dots, \mathbf{a}_n) \in \mathbb{Z}_q^{n \times m}$ with n columns and m rows, to the vector $T(\mathbf{A}) = (\mathbf{a}_1^\top, \dots, \mathbf{a}_n^\top)^\top \in \mathbb{Z}_q^{n \cdot m}$. Namely, the transformation concatenates the columns of \mathbf{A} to form a vector of length $n \cdot m$. We define the tensor product between two vectors \mathbf{a}, \mathbf{b} to be $\mathbf{a} \otimes \mathbf{b} := T(\mathbf{a} \cdot \mathbf{b}^\top)$. Similarly, given $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{B} \in \mathbb{Z}_q^{n' \times m'}$, we define the vector product between two matrices $\mathbf{A} \otimes \mathbf{B}$ to be the matrices:

$$\left(T(\mathbf{a}_1 \mathbf{b}_1^\top), \dots, T(\mathbf{a}_1 \mathbf{b}_{n'}^\top), \dots, T(\mathbf{a}_n \mathbf{b}_1^\top), \dots, T(\mathbf{a}_n \mathbf{b}_{n'}^\top) \right) \in \mathbb{Z}_q^{n \cdot n' \times m \cdot m'}.$$

We can show the following property:

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

Our results will be proven secure under the following decisional assumption.

Definition 1 (Matrix Decisional Diffie-Hellman Assumption in \mathbb{G}_1 , [EHK⁺13]).

The $\mathcal{D}_{n,d}$ -MDDH assumption holds if for all non-uniform PPT adversaries \mathcal{A} ,

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

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

For $Q \in \mathbb{N}$, $\mathbf{W} \leftarrow_{\$} \mathbb{Z}_q^{d \times Q}$ and $\mathbf{U} \leftarrow_{\$} \mathbb{Z}_q^{n \times Q}$, we consider the Q -fold $\mathcal{D}_{n,d}$ -MDDH assumption, which 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 [EHK⁺13] it is shown that the two problems are equivalent, where the reduction loses at most a factor $n-d$.

Lemma 1 (Random self-reducibility of $\mathcal{D}_{n,d}$ -MDDH, [EHK⁺13]). *Let $n, d, Q \in \mathbb{N}$ with $n > d$ and $Q > n - d$. For any PPT adversary \mathcal{A} , there exists an adversary \mathcal{B} such that $T(\mathcal{B}) \approx T(\mathcal{A}) + Q \cdot \text{poly}(\lambda)$, with $\text{poly}(\lambda)$ independent of $T(\mathcal{A})$, and*

$$\text{Adv}_{\mathbb{G}_1, \mathcal{D}_{n,d}, \mathcal{A}}^{\text{Q-MDDH}}(\lambda) \leq (n-d) \cdot \text{Adv}_{\mathbb{G}_1, \mathcal{D}_{n,d}, \mathcal{B}}^{\text{mddh}}(\lambda) + \frac{1}{q-1}$$

where, given $\mathbf{A} \leftarrow_{\$} \mathcal{U}_{n,d}$, $\mathbf{W} \leftarrow_{\$} \mathbb{Z}_q^{k \times Q}$ and $\mathbf{U} \leftarrow_{\$} \mathbb{Z}_q^{n \times Q}$:

$$\text{Adv}_{\mathbb{G}_1, \mathcal{D}_{n,d}, \mathcal{A}}^{\text{Q-MDDH}}(\lambda) := |\Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{AW}]_1) = 1] - \Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{U}]_1) = 1]|,$$

Corollary 1. *Let $n, d, d' \in \mathbb{N}$ with $n > d$ and $d' \geq d$. For any PPT adversary \mathcal{A} , there exists an adversary \mathcal{B} such that $T(\mathcal{B}) \approx T(\mathcal{A}) + \text{poly}(\lambda)$, with $\text{poly}(\lambda)$ independent of $T(\mathcal{A})$, and*

$$\text{Adv}_{\mathbb{G}_1, \mathcal{D}_{n,d'}, \mathcal{A}}^{\text{mddh}}(\lambda) = \text{Adv}_{\mathbb{G}_1, \mathcal{D}_{n,d}, \mathcal{B}}^{\text{mddh}}(\lambda).$$

We state a tighter random-self reducibility property for case of the uniform distribution \mathcal{U} .

Lemma 2 (Random self-reducibility of $\mathcal{U}_{n,d}$ -MDDH, [EHK⁺13]). *Let $n, d, Q \in \mathbb{N}$ with $n > d$ and $Q > n - d$. For any PPT adversary \mathcal{A} , there exists an adversary \mathcal{B} such that $T(\mathcal{B}) \approx T(\mathcal{A}) + Q \cdot \text{poly}(\lambda)$, with $\text{poly}(\lambda)$ independent of $T(\mathcal{A})$, and*

$$\text{Adv}_{\mathbb{G}_1, \mathcal{U}_{n,d}, \mathcal{A}}^{\text{Q-MDDH}}(\lambda) \leq \text{Adv}_{\mathbb{G}_1, \mathcal{U}_{n,d}, \mathcal{B}}^{\text{mddh}}(\lambda) + \frac{1}{q-1}$$

Lemma 3 ($\mathcal{D}_{n,d}$ -MDDH \Rightarrow $\mathcal{U}_{n,d}$ -MDDH, [EHK⁺13]). *Let $\mathcal{D}_{n,d}$ be a matrix distribution. For any adversary \mathcal{A} on the $\mathcal{U}_{n,d}$ -distribution, there exists an adversary \mathcal{B} on the $\mathcal{D}_{n,d}$ -assumption such that $T(\mathcal{B}) \approx T(\mathcal{A})$ and $\text{Adv}_{\mathbb{G}_1, \mathcal{U}_{n,d}, \mathcal{A}}^{\text{mddh}}(\lambda) = \text{Adv}_{\mathbb{G}_1, \mathcal{D}_{n,d}, \mathcal{B}}^{\text{mddh}}(\lambda)$.*

Lemma for Rand-RCCA security. The main technical tool employed by [FFHR19], 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 $(\mathbf{f} + \mathbf{F}\mathbf{v})^\top \mathbf{u}$ is pair-wise independent with respect to the quotient set $\mathbb{Z}_q^{d+2} / \text{span}(\mathbf{E})$ when given as side information the matrix $\mathbf{F}\mathbf{E}$ 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 $n' > d+1$. 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 [FFHR19], because the core technical parts of our work and theirs are different.

Lemma 4 (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 :*

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

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

Proof. The original lemma was proved in [FFHR19] for slightly different parameters, namely for matrices $\mathbf{D} \in \mathbb{Z}_q^{d+1 \times d}$ and $\mathbf{E} \in \mathbb{Z}_q^{d+2 \times d}$ and with upper bound $1/q$ to the winning probability of the adversary. We show here that their lemma holds for generic $n, n' > d$. To show this we reduce to the case proved in [FFHR19]. We parse the matrices \mathbf{D} and \mathbf{E} as follows:

$$\mathbf{D} = \begin{pmatrix} \bar{\mathbf{D}} \\ \mathbf{D}' \end{pmatrix} \cdot \mathbf{P} \text{ and } \mathbf{E} = \begin{pmatrix} \bar{\mathbf{E}}' \\ \bar{\mathbf{E}} \end{pmatrix} \mathbf{P}'$$

where $\bar{\mathbf{D}}$ has $d+1$ rows and $\bar{\mathbf{E}}$ has $d+2$ rows and \mathbf{P}, \mathbf{P}' are two uniformly random permutation matrices. Now consider the reduction to the original lemma where we use the matrices $\bar{\mathbf{D}}, \bar{\mathbf{E}}$. We obtain from the challenger the public parameters $(\bar{\mathbf{f}}^\top \bar{\mathbf{D}}, \bar{\mathbf{F}}^\top \bar{\mathbf{D}}, \bar{\mathbf{F}}\bar{\mathbf{E}})$ for uniformly random $\bar{\mathbf{f}}, \bar{\mathbf{F}}$. Now consider the vector \mathbf{f} and matrix \mathbf{F} implicitly defined as follows:

$$\mathbf{f} = \begin{pmatrix} \bar{\mathbf{f}} \\ \mathbf{f}' \end{pmatrix} \text{ and } \mathbf{F} = \begin{pmatrix} \mathbf{F}' & \bar{\mathbf{F}} \\ \mathbf{F}'' & \mathbf{F}''' \end{pmatrix}$$

where $\mathbf{f}', \mathbf{F}', \mathbf{F}'', \mathbf{F}'''$ are uniformly random, and can be sampled by the reduction. Notice that we can compute the public parameters for the adversary as follows:

$$\mathbf{f}^\top \mathbf{D} \leftarrow (\bar{\mathbf{f}}^\top \bar{\mathbf{D}} + \mathbf{f}'^\top \mathbf{D}') \mathbf{P}, \quad \mathbf{F}^\top \mathbf{D} \leftarrow \begin{pmatrix} \mathbf{F}'^\top \bar{\mathbf{D}} + \mathbf{F}''^\top \mathbf{D}' \\ \bar{\mathbf{F}}^\top \bar{\mathbf{D}} + \mathbf{F}'''^\top \mathbf{D}' \end{pmatrix} \mathbf{P} \text{ and } \mathbf{F}\mathbf{E} \leftarrow \begin{pmatrix} \mathbf{F}'\bar{\mathbf{E}}' + \bar{\mathbf{F}}\bar{\mathbf{E}} \\ \mathbf{F}''\bar{\mathbf{E}}' + \mathbf{F}'''\bar{\mathbf{E}} \end{pmatrix} \mathbf{P}'$$

moreover, we can handle the oracle query $\mathbf{v}^* \in \mathbb{Z}_q^{n'}$ of the adversary using the oracle of the reduction. Specifically, let $\bar{\mathbf{v}}^*$ be the last $d+2$ rows of \mathbf{v}^* and \mathbf{v}' be remaining rows, the reduction queries $\bar{\mathbf{v}}$ to its own oracle and obtain $\bar{\mathbf{y}} = \bar{\mathbf{f}} + \bar{\mathbf{F}}\bar{\mathbf{v}}^*$. The reduction can return to the adversary

$$\mathbf{y} \leftarrow \begin{pmatrix} \bar{\mathbf{y}} + \mathbf{F}'\mathbf{v}' \\ \mathbf{f}' + (\mathbf{F}'' | \mathbf{F}''')\mathbf{v}^* \end{pmatrix}$$

Finally, the adversary returns $(z, \mathbf{u}, \mathbf{v})$, the reduction outputs tuple $(\bar{z}, \bar{\mathbf{u}}, \bar{\mathbf{v}})$ where $\bar{\mathbf{u}}$ are the first $d+1$ rows of \mathbf{u} , $\bar{\mathbf{v}}$ are the last $d+2$ rows of \mathbf{v} and:

$$\bar{z} \leftarrow z - (\mathbf{F}'\mathbf{v}')^\top \bar{\mathbf{u}} - (\mathbf{f}' + \mathbf{F}''\mathbf{v}' + \mathbf{F}'''\bar{\mathbf{v}})^\top \mathbf{u}'$$

where \mathbf{v}' , \mathbf{u}' are the remaining rows of respectively \mathbf{v} , \mathbf{u} .

We need to prove that the forging probability of our reduction is as claimed in the statement of the lemma. To do so notice that if $\mathbf{u} \notin \text{span}(\mathbf{D})$ then we can write $\mathbf{u} = \mathbf{D}\mathbf{r} + \mathbf{z}$ for a non-zero vector \mathbf{z} . Thus $\bar{\mathbf{u}} \notin \text{span}(\bar{\mathbf{D}})$ if there exists a non-zero entry in the first $d + 2$ rows of $\mathbf{z} \cdot \mathbf{P}$. In the worst case \mathbf{z} has only one non-zero coordinate. Thus the probability that $\bar{\mathbf{u}} \notin \text{span}(\bar{\mathbf{D}})$ is $\frac{d+1}{n}$. A similar argument shows that the probability of $\bar{\mathbf{v}} - \bar{\mathbf{v}}^* \notin \text{span}(\bar{\mathbf{E}})$ is $\frac{d+2}{n'}$. Notice that the permutation matrices \mathbf{P}, \mathbf{P}' are independent of the view of the adversary, thus the aforementioned events are independent of the winning probability of \mathcal{A} , which concludes the proof of the lemma.

3 Non-Interactive Proof Systems (NIPS)

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

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

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

When $\text{ppk} \neq \text{psk}$ 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 [JR13]. All the NIPSs of this paper are *structure-preserving*. The usual definition of structure-preserving postulates that 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. We consider the version of the structure-preserving property of [FFHR19] where 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 [Hof17].

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

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

(Statistical) $(\mathcal{L}^{\text{sim}}, \mathcal{L}^{\text{ver}}, \mathcal{L}^{\text{snd}})$ -soundness. Consider the game $\text{Exp}_{\mathcal{A}, \text{PS}}^{\text{snd}}$ played by an adversary \mathcal{A} in Fig. 1. Let $\text{Adv}_{\text{PS}, \mathcal{A}}^{\text{snd}}(\lambda)$ be the probability that $\text{Exp}_{\mathcal{A}, \text{PS}}^{\text{snd}}(\lambda) = 1$. We require that for all (possibly unbounded) \mathcal{A} that only make a polynomial number of oracle queries, $\text{Adv}_{\text{PS}, \mathcal{A}}^{\text{snd}}(\lambda)$ 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 Definition 2, we say that PS is (ϵ, T) -composable zero-knowledge if there exists a PPT algorithm $\overline{\text{PGen}}$ such that:

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

$$\text{PPrv}(\text{ppk}, x, w) \equiv \text{PSim}(\text{psk}, x).$$

Malleable NIPS. We use the definitional framework of Chase *et al.* [CKLM12] 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 transformation that depends on the pars . Let $T = (T_{\text{el}}, T_{\text{wit}})$ be a pair of efficiently computable functions, that we refer to as a *transformation*.

Definition 5 (Admissible transformations). An efficient relation \mathcal{R} is closed under a transformation $T = (T_{\text{el}}, T_{\text{wit}})$ if for any $(\text{pars}, x, w) \in \mathcal{R}$ the pair $(\text{pars}, T_{\text{el}}(\text{pars}, x), T_{\text{wit}}(w)) \in \mathcal{R}$. 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.

We are ready to define malleable proof systems.

Definition 6 (Malleable NIPS). Let PS be an NIPS for \mathcal{L} as in Definition 2, and let $\text{PEvl}(\text{ppk}, x, \pi, T)$ 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

Experiment $\mathbf{Exp}_{\mathcal{A}, \mathbf{PS}}^{\text{snd}}$	Experiment $\mathbf{Exp}_{\mathcal{A}, \mathbf{PS}}^{\text{der-priv}}$
$pars \leftarrow \mathcal{A}(1^\lambda); b \leftarrow 0$ $(ppk, psk) \leftarrow \mathcal{PGen}(1^\lambda, pars)$ $\mathcal{A}^{\mathcal{O}_{\text{sim}}(\cdot), \mathcal{O}_{\text{ver}}(\cdot, \cdot)}(ppk)$ return b	$(ppk, psk) \leftarrow \mathcal{PGen}(1^\lambda, pars)$ $b^* \leftarrow \{0, 1\}$ $(x, w, \pi, T) \leftarrow \mathcal{A}(ppk, psk)$ if $\mathcal{V}(ppk, x, \pi) \stackrel{?}{=} 0 \vee R(x, w) \stackrel{?}{=} 0$: $b \leftarrow \{0, 1\}$ return b
<div style="border: 1px solid black; padding: 5px;"> <p>Oracle $\mathcal{O}_{\text{ver}}(x, \pi)$</p> <p>if $x \in \mathcal{L}_{pars}^{\text{ver}}$: return $\mathcal{PVer}(psk, x, \pi)$</p> <p>if $x \in \mathcal{X}_{pars} \setminus \mathcal{L}_{pars}^{\text{snd}} \wedge \mathcal{PVer}(psk, x, \pi) \stackrel{?}{=} 1$: $b \leftarrow 1$</p> <p>return \perp</p> </div>	if $b^* \stackrel{?}{=} 0$: $\pi' \leftarrow \mathcal{PPrv}(ppk, T_{\text{el}}(pars, x), T_{\text{wit}}(w))$ else $\pi' \leftarrow \mathcal{PEvl}(ppk, x, \pi, T)$ $b \leftarrow \mathcal{A}(\pi')$ return $b \stackrel{?}{=} b^*$
<div style="border: 1px solid black; padding: 5px;"> <p>Oracle $\mathcal{O}_{\text{sim}}(x)$</p> <p>if $x \in \mathcal{L}_{pars}^{\text{sim}}$: return $\mathcal{PSim}(psk, x)$ else \perp</p> </div>	

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

π' . We say that \mathbf{PS} and \mathcal{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 $\mathcal{PGen}(1^\lambda, pars)$, all $T \in \mathcal{T}$, and all x, π we have $\mathcal{PVer}(psk, T_{\text{el}}(pars, x), \pi') = 1$ if and only if $\mathcal{PVer}(psk, x, \pi) = 1$.

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

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

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

Similarly to [FFHR19], we additionally 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.

Definition 8 (Tightness for Proofs). We say that a NIZK has tight proofs if for any (possibly unbounded) adversary \mathcal{A} the following holds:

$$\Pr \left[\pi \notin \{ \mathcal{PPrv}(ppk, x, w; \omega) \}_{\omega \in \{0, 1\}^\lambda} \wedge \begin{array}{l} (ppk, psk) \leftarrow \mathcal{PGen}(1^\lambda, pars) \\ \mathcal{PVer}(psk, x, \pi) = 1 \\ (x, w, \pi) \leftarrow \mathcal{A}(ppk) \end{array} \right] \in \text{negl}(\lambda).$$

We notice that the property above is true for Groth-Sahai Proofs, for the quasi-adaptive proof system of Kiltz and Wee [KW15] and for designated-verifier proof system based on projective hash functions. In particular, for GS proofs, for any commitment to the witness, the prover generates a proof that is uniformly distributed over the set of all the possible valid proofs. The proofs of Kiltz and Wee and the proofs for proof system based on projective hash functions are unique, therefore the condition is trivially true.

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 $\mathcal{L} = \text{span}([\mathbf{D}]_1)$:

- $\text{PGen}(pars)$ parses $pars$ as $\text{prm}_G, [\mathbf{D}]_1 \in \mathbb{G}_1^{n \times d}$ 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:

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

- $\text{PPrv}(ppk, [\mathbf{u}]_1, \mathbf{r})$ computes $\pi \leftarrow \mathbf{k}^\top [\mathbf{D} \otimes \mathbf{D}]_T \cdot (\mathbf{r} \otimes \mathbf{r})$ for $[\mathbf{u}]_1 = [\mathbf{D}]_1 \mathbf{r}$
- $\text{PSim}(psk, [\mathbf{u}]_1)$ computes $\pi \leftarrow \mathbf{k}^\top ([\mathbf{u}]_1 \otimes [\mathbf{u}]_1)$
- $\text{PVer}(psk, [\mathbf{u}]_1, \pi)$ 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 :

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

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 $\text{PEvl}(ppk, \hat{\mathbf{r}}, [\mathbf{u}]_1, \pi)$ 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 $\mathcal{L} = \text{span}([\mathbf{D}]_1)$ 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 $\mathbf{D}_i = \mathbf{D} \parallel \bar{\mathbf{D}}_i$ 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 $n^2 > 2n \cdot d + 2d'^2$, 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. It is easy to prove that 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 transformation $\hat{\mathbf{r}}$, a proof π for $[\mathbf{u}]$ verifies if and only if the proof $\hat{\pi}$ obtained executing PEV 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 $q^{n^2 - 2nd - 2d'^2}$. Therefore, we require that $n^2 > 2n \cdot d + 2d'^2$ holds.

To see this, think of \mathbf{k} as 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 $[\mathbf{u}]_1 \in \text{span}([\mathbf{D}_i])$ the oracle gives away, at the worst case, the equations $\mathbf{k}^\top (\bar{\mathbf{D}}_i \otimes \bar{\mathbf{D}}_i)$ which count for d'^2 equations for each $i \in \{0, 1\}$ which sum up to $2d'^2$ equations in total. Indeed, expanding $\mathbf{k}^\top (\mathbf{D}_i \otimes \bar{\mathbf{D}}_i)$ we obtain $\mathbf{k}^\top (\mathbf{D} \otimes \mathbf{D} | \bar{\mathbf{D}}_i \otimes \mathbf{D} | \mathbf{D} \otimes \bar{\mathbf{D}}_i | \bar{\mathbf{D}}_i \otimes \bar{\mathbf{D}}_i)$. The vectors $\mathbf{k}^\top (\bar{\mathbf{D}}_i \otimes \mathbf{D})$ and $\mathbf{k}^\top (\mathbf{D} \otimes \bar{\mathbf{D}}_i)$ can be computed given the proving key and $\mathbf{D}_0, \mathbf{D}_1$. In fact, computing $\mathbf{k}^\top (\mathbf{D} \otimes \mathbf{I}) (\mathbf{I} \otimes \bar{\mathbf{D}}_i)$ we obtain $\mathbf{k}^\top (\mathbf{D}\mathbf{I} \otimes \mathbf{I}\bar{\mathbf{D}}_i) = \mathbf{k}^\top (\mathbf{D} \otimes \bar{\mathbf{D}}_i)$. And in a similar way, we can compute $\mathbf{k}^\top (\bar{\mathbf{D}}_i \otimes \mathbf{D})$. In total we are giving up $2n \cdot d + 2d'^2$ 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 $1/q^{n^2 - 2nd + 2d'^2} \leq 1/q$, as it is the same as the event of guessing a uniformly random vector from a subspace of dimension $n^2 - 2nd + 2d'^2$ of $\mathbb{Z}_q^{n^2}$, thus $\Pr[\mathbf{H}_j = 1] \leq \Pr[\mathbf{H}_{j+1} = 1] +$

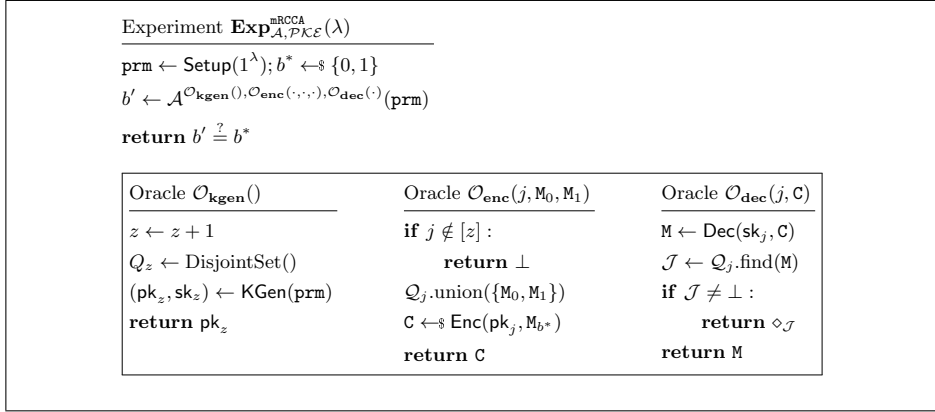


Fig. 2: The multi-user and multi-ciphertext RCCA Security Experiment.

$1/q$. 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 $\text{PPrv}(ppk, [\mathbf{u} + \mathbf{D}\hat{\mathbf{r}}]_1, \mathbf{r} + \hat{\mathbf{r}}) = \mathbf{k}^\top [\mathbf{D} \otimes \mathbf{D}]_T \cdot ((\mathbf{r} + \hat{\mathbf{r}}) \otimes (\mathbf{r} + \hat{\mathbf{r}})) = \text{PEvl}(ppk, \pi, \hat{\mathbf{r}})$. For the latter, given an instance $[\mathbf{u}]_\perp$ such that $[\mathbf{u}]_1 = [\mathbf{D}]_1 \mathbf{r}$, we have that $\text{PSim}(psk, [\mathbf{u}]_1) = \mathbf{k}^\top ([\mathbf{u}]_1 \otimes [\mathbf{u}]_1) = \mathbf{k}^\top ([\mathbf{D}\mathbf{r}]_1 \otimes [\mathbf{D}\mathbf{r}]_1) = \text{PPrv}(ppk, [\mathbf{u}]_1, \mathbf{r})$.

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

A re-randomizable PKE (Rand-PKE) scheme $\mathcal{PK}\mathcal{E}$ is a tuple of five algorithms:

$\text{Setup}(1^\lambda)$ upon input the security parameter 1^λ produces public parameters prm , which include the description of the message (\mathcal{M}) and ciphertext space (\mathcal{C}).

$\text{KGen}(\text{prm})$ upon input the parameters prm , outputs a key pair (pk, sk) .

$\text{Enc}(\text{pk}, M)$ upon inputs a public key pk and a message $M \in \mathcal{M}$, outputs a ciphertext $C \in \mathcal{C}$.

$\text{Dec}(\text{pk}, \text{sk}, C)$ upon input the secret key sk and a ciphertext C , outputs a message $M \in \mathcal{M}$ or an error symbol \perp .

$\text{Rand}(\text{pk}, C)$ upon inputs a public key pk and a ciphertext C , outputs another ciphertext C' .

Definition 9 (multi-user and multi-ciphertext Replayable CCA Security). Consider the experiment $\mathbf{Exp}_{\mathcal{A}, \mathcal{PK}\mathcal{E}}^{\text{mRCCA}}$ in Fig. 2, with parameters λ , an adversary \mathcal{A} , and a PKE scheme $\mathcal{PK}\mathcal{E}$. We say that $\mathcal{PK}\mathcal{E}$ 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} :

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

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 [Tar75]. 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 to initialize 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 [CKN03] by showing that our definition tightly implies the UC-RCCA definition of the same paper⁷. In Fig. 3 we recall the definition of the ideal functionality $\mathcal{F}_{\text{RPKE}}$ which formalizes the notion of replay security for public-key encryption scheme in the universal composability model.

Theorem 2. *Let $\mathcal{PK}\mathcal{E}$ be a PKE scheme with message space \mathcal{D} . There exists a simulator \mathcal{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:*

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

Proof. The simulator is the same as the one presented at Pag 14 of [CKN03]. In particular:

- At first activation of the simulator it sets the set of random messages \mathcal{M}^* to be the empty set.
- When \mathcal{S} receives the message (**KeyGen**, id) from the ideal functionality, then it runs $\text{pk}, \text{sk} \leftarrow_{\$} \text{KGen}(\text{prm})$, returns pk to the ideal functionality and stores the keypair.
- When \mathcal{S} receives a message (**Encrypt**, id, e', P_j) if $e' = \text{pk}$ then the simulator samples a uniformly random message M^* from the set $D_{\lambda} \setminus \mathcal{M}^*$ and adds M^* into the set \mathcal{M}^* , computes $C \leftarrow_{\$} \text{Enc}(\text{pk}, M^*)$, store the tuple (C, M^*) and returns C , otherwise it additionally receives a message M from the ideal functionality, it returns $\text{Enc}(e', M)$.
- When \mathcal{S} receives a message (**Decrypt**, id, c) from the ideal functionality it first computes $M = \text{Dec}(\text{sk}, c)$. If it exists a recorded tuple (C, M) then it returns $(\text{ciphertext}, C)$ otherwise it returns $(\text{plaintext}, M)$.

We reduce to our multi-ciphertext RCCA security notion of Definition 9. Consider the reduction \mathcal{B} that runs the environments \mathcal{Z} and answers its messages as follows:

⁷ In [CKN03], the IND-RCCA notion implies the UC-RCCA notion with a loss of security that is proportional to the running time of the environment.

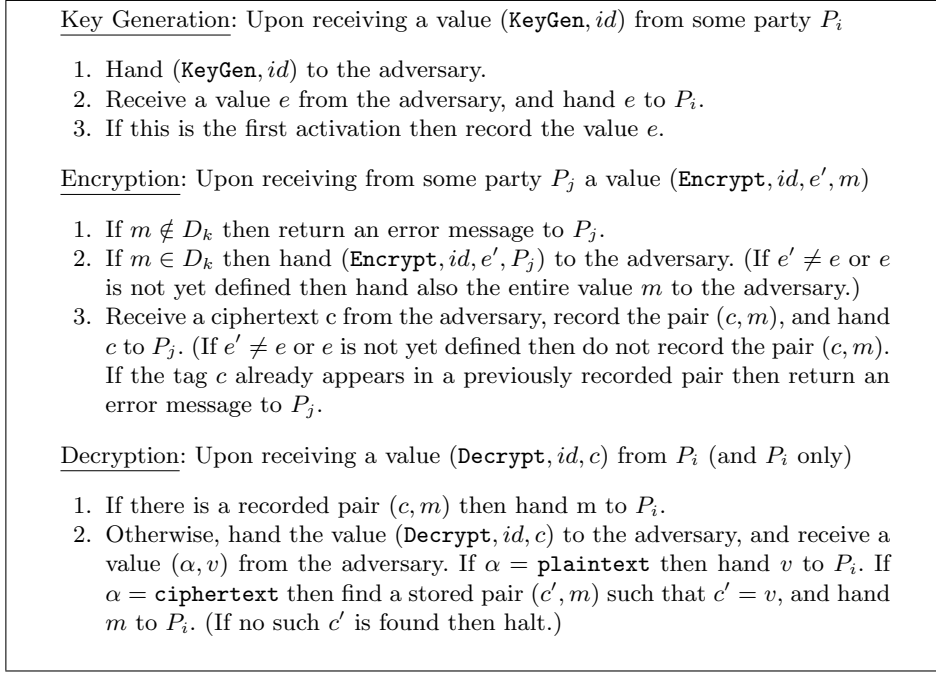


Fig. 3: The ideal functionality $\mathcal{F}_{\text{RPKE}}$, when parameterized by message domain ensemble $D = \{D_\lambda\}_{\lambda \in \mathbb{N}}$ and security parameter λ , and interacting with an adversary S , and parties P_1, \dots, P_n .

- Upon message (KeyGen, id) to party P_i , if this is the first activation then \mathcal{B} queries its own key generation oracle and receives pk which returns as output of P_i . Else it generates a keypair and returns pk as output of P_i .
- Upon message $(\text{Encrypt}, id, e', M)$ from \mathcal{Z} to a party P_j , it check that $M \in D_\lambda$ and otherwise it returns an error as output of P_j . If $e = \text{pk}$ it samples a random message $M^* \leftarrow_{\$} D_\lambda \setminus \mathcal{M}^*$ and stores the message in the list \mathcal{M}^* of random messages. Additionally, it asserts that the message from the environment M is not present in \mathcal{M}^* , if such event happens it **aborts**. It sets $(M_0, M_1) \leftarrow (M, M^*)$ and queries its own encryption oracle obtaining C which returns as output of P_j .
- Upon message $(\text{Decrypt}, id, c)$ sent by the environment to P_i , it queries with c the decryption oracle. If the decryption oracle return \diamond_J let $A = J \setminus \mathcal{M}^*$, if $|A| > 1$ then the reduction **aborts**, else $A = \{M\}$ and it returns M as output of P_j . If the decryption oracle returns M then it returns M as output of P_j .

Eventually, the reduction outputs the same decision bit of the environment \mathcal{Z} .

It is easy to check that if we condition on the event that the reduction \mathcal{B} does not abort then the view of the environment is exactly the same as in the real world if the challenge bit of the RCCA experiment is 0 and exactly the same as in the ideal world when the challenge bit of the RCCA experiment is 1.

Let **Abort** be such event, we compute the probability of **Abort** conditioned on the challenge bit being 0. Notice that, since the event can be checked efficiently, there exists a reduction \mathcal{B}' with running time less or equal to the running time of \mathcal{B} such that $\Pr[\mathbf{Abort}|b^* = 1] \leq \Pr[\mathbf{Abort}|b^* = 0] + \mathbf{Adv}_{\mathcal{B}', \mathcal{PK}\mathcal{E}}^{\text{mRCCA}}(\lambda)$. The reduction \mathcal{B}' runs \mathcal{B} and returns 1 if \mathcal{B} aborts and otherwise it outputs a random bit.

Notice that when $b = 0$ the random messages in \mathcal{M}^* are not in the view of \mathcal{Z} (since the ciphertexts contain encryption of the real messages), thus the probability that **Abort** happens because the environment sends a message $(\mathbf{Encrypt}, id, e', \mathbf{M})$ where $\mathbf{M} \in \mathcal{M}^*$ is upperbounded by $|\mathcal{M}^*|/|D_\lambda|$. Moreover, notice that when the decryption oracle returns \diamond_J , since the random messages \mathbf{M}^* chosen by the reduction are all different, the common element in all the encryption queries that made them to be included in J , by the Definition 9, must be the real message sent by the \mathcal{Z} .

Summing up all together, taking the maximum between the advantage of \mathcal{B} and \mathcal{B}' , and noticing that $|\mathcal{M}^*| \leq T_{\mathcal{Z}}$ we obtain the bound in the statement of the theorem.

We move now to the definition of perfect re-randomizability as defined in [FFHR19]. Roughly speaking, a PKE is re-randomizable if there exists a procedure that creates fresh and unlinkable ciphertexts from an old ciphertext. In the RCCA setting, we also face the issue of defining what re-randomization for maliciously crafted ciphertexts should be. In [FFHR19], this is handled by imposing that it is hard to find such malicious ciphertexts.

Definition 10 (Perfect Re-randomizability, [FFHR19]). *We say that $\mathcal{PK}\mathcal{E}$ is perfectly re-randomizable (Re-Rand, for short) if the following three conditions are met:*

1. **(Indistinguishability)** *For any $\lambda \in \mathbb{N}$, any $\text{prm} \leftarrow \text{Setup}(1^\lambda)$, any $(\text{pk}, \text{sk}) \leftarrow \text{KGen}(\text{prm}, 1^\ell)$, for any $\mathbf{M} \in \mathcal{M}$ and any $\mathbf{C} \in \text{Enc}(\text{pk}, \mathbf{M})$ the following two distributions are identical*

$$\mathbf{C}_0 \leftarrow \text{Enc}(\text{pk}, \mathbf{M}) \text{ and } \mathbf{C}_1 \leftarrow \text{Rand}(\text{pk}, \mathbf{C});$$

2. **(Correctness)** *For any $\lambda \in \mathbb{N}$, any $\text{prm} \leftarrow \text{Setup}(1^\lambda)$, any $(\text{pk}, \text{sk}) \leftarrow \text{KGen}(\text{prm}, 1^\ell)$, for any (possibly malicious) ciphertext \mathbf{C} and every $\mathbf{C}' \leftarrow \text{Rand}(\text{pk}, \mathbf{C})$ it holds*

$$\text{Dec}(\text{sk}, \mathbf{C}') = \text{Dec}(\text{sk}, \mathbf{C}).$$

3. **(Tightness of Decryption)** *For any (possibly unbounded) adversary \mathcal{A} and any sequence of parameters $\{\text{prm}_\lambda \leftarrow \text{Setup}(1^\lambda)\}_{\lambda \in \mathbb{N}}$ the following holds:*

$$\Pr \left[\exists \mathbf{M} : \mathbf{C} \notin \text{Enc}(\text{pk}, \mathbf{M}) \wedge \text{Dec}(\text{sk}, \mathbf{C}) = \mathbf{M} \neq \perp : \begin{array}{l} (\text{pk}, \text{sk}) \leftarrow \text{KGen}(\text{prm}_\lambda) \\ \mathbf{C} \leftarrow \mathcal{A}(\text{pk}) \end{array} \right] \in \text{negl}(\lambda).$$

Finally, we define the notion of public verifiability for PKE schemes.

Definition 11 (Public Verifiability). $\mathcal{PKE} = (\text{Setup}, \text{KGen}, \text{Enc}, \text{Dec}, \text{Rand})$ is a public key scheme with publicly verifiable ciphertexts if there is a deterministic algorithm Ver which, on input (pk, \mathcal{C}) outputs an error symbol \perp whenever $\text{Dec}(\text{pk}, \text{sk}, \mathcal{C}) = \perp$, else it outputs valid.

5 Our Rand-RCCA PKE Scheme

<p>Setup(1^λ)</p> <hr/> $\text{prm}_G = (q, \mathbb{G}_1, \mathbb{G}_T, e, \mathcal{P}_1) \leftarrow \text{GGen}(1^\lambda)$ $\mathcal{M} \leftarrow \mathbb{G}_1; \mathcal{C} \leftarrow \mathbb{G}_1^{n+2} \times \mathbb{G}_T \times \mathcal{P}$ $\text{prm} \leftarrow (\text{prm}_G, \mathcal{M}, \mathcal{C})$ return prm	<p>Enc($\text{pk}, [\mathbf{M}]_1$)</p> <hr/> $\mathbf{r} \leftarrow \mathbb{Z}_q^d$ $[\mathbf{u}]_1 \leftarrow [\mathbf{D}]_1 \cdot \mathbf{r}, \pi \leftarrow \text{PPriv}(ppk, [\mathbf{u}]_1, \mathbf{r})$ $[p]_1 \leftarrow [\mathbf{a}^\top \mathbf{D}]_1 \cdot \mathbf{r} + [\mathbf{M}]_1$ $[\mathbf{x}]_1 \leftarrow ([\mathbf{u}^\top]_1, [p]_1)^\top$ $[y]_T \leftarrow ([\mathbf{f}^\top \mathbf{D}]_T + e([\mathbf{x}]_1^\top, [\mathbf{F}^\top \mathbf{D}]_1)) \cdot \mathbf{r}$ return $\mathcal{C} := ([\mathbf{x}]_1, [y]_T, \pi)$
<p>KGen(prm)</p> <hr/> $\mathbf{D} \leftarrow \mathcal{D}_{n,d}, \mathbf{a} \leftarrow \mathbb{Z}_q^n$ $\mathbf{D}^* \leftarrow (\mathbf{D}^\top, (\mathbf{a}^\top \mathbf{D})^\top)^\top$ $\mathbf{f} \leftarrow \mathbb{Z}_q^n, \mathbf{F} \leftarrow \mathbb{Z}_q^{n \times n+1}$ $\text{pars} \leftarrow (\text{prm}_G, [\mathbf{D}]_1)$ $ppk, psk \leftarrow \text{PGen}(\text{pars})$ $\text{pk} \leftarrow ([\mathbf{D}^*]_1, [\mathbf{f}^\top \mathbf{D}]_T, [\mathbf{F}^\top \mathbf{D}]_1, [\mathbf{F} \mathbf{D}^*]_1, ppk)$ $\text{sk} \leftarrow (\mathbf{a}, \mathbf{f}, \mathbf{F}, psk)$ return (pk, sk)	<p>Dec(sk, \mathcal{C})</p> <hr/> parse \mathcal{C} as $([\mathbf{x}]_1, [y]_T, \pi)$ parse $[\mathbf{x}^\top]_1$ as $([\mathbf{u}^\top]_1, [p]_1)$ $[\mathbf{M}]_1 \leftarrow [p]_1 - [\mathbf{a}^\top \mathbf{u}]_1$ $[y']_T \leftarrow \mathbf{f}^\top e([1]_1, [\mathbf{u}]_1) + e(\mathbf{F}[\mathbf{x}]_1, [\mathbf{u}]_1)$ $b_1 \leftarrow [y']_T \stackrel{?}{=} [y]_T, b_2 \leftarrow \text{PVer}(psk, [\mathbf{u}]_1, \pi)$ if $b_1 \wedge b_2$ return $[\mathbf{M}]_1$ else \perp
<p>Rand(pk, \mathcal{C})</p> <hr/> parse \mathcal{C} as $([\mathbf{x}]_1, [y]_T, \pi)$, parse $[\mathbf{x}^\top]_1$ as $([\mathbf{u}^\top]_1, [p]_1)$ $\hat{\mathbf{r}} \leftarrow \mathbb{Z}_q^d, [\hat{\mathbf{x}}]_1 \leftarrow [\mathbf{x}]_1 + [\mathbf{D}^*]_1 \cdot \hat{\mathbf{r}}$ $[\hat{y}]_T \leftarrow [y]_T + [\mathbf{f}^\top \mathbf{D}]_T \cdot \hat{\mathbf{r}} + e([\mathbf{x}]_1, [\mathbf{F}^\top \mathbf{D}]_1 \cdot \hat{\mathbf{r}}) + e([\mathbf{F} \mathbf{D}^*]_1 \cdot \hat{\mathbf{r}}, [\hat{\mathbf{u}}]_1)$ $\hat{\pi} \leftarrow \text{PEvl}(ppk, [\mathbf{u}]_1, \pi, \hat{\mathbf{r}})$ return $\hat{\mathcal{C}} := ([\hat{\mathbf{x}}]_1, [\hat{y}]_T, \hat{\pi})$	

Fig. 4: 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. 4. 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 [FFHR19] where we additionally append a benign proof to prove almost tight-security. The main technical component from [FFHR19] 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. We prove that the scheme presented in Fig. 4 is perfectly rerandomizable, according to Definition 10. We start by introducing the following lemma.

Lemma 5. *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*

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

Proof.

$$\begin{aligned} (\mathbf{f}^\top + \hat{\mathbf{x}} \mathbf{F}^\top) \hat{\mathbf{u}} &= (\mathbf{f}^\top + (\mathbf{x} + \mathbf{D}^* \hat{\mathbf{r}})^\top \mathbf{F}^\top) (\mathbf{u} + \mathbf{D} \hat{\mathbf{r}}) \\ &= (\mathbf{f}^\top + \mathbf{x}^\top \mathbf{F}^\top) \mathbf{u} + (\mathbf{f}^\top + \mathbf{x}^\top \mathbf{F}^\top) (\mathbf{D} \hat{\mathbf{r}}) + (\mathbf{D}^* \hat{\mathbf{r}})^\top \mathbf{F}^\top \mathbf{u} + (\mathbf{D}^* \hat{\mathbf{r}})^\top \mathbf{F}^\top \mathbf{D} \hat{\mathbf{r}} \end{aligned}$$

Notice that the third term can be rewritten as:

$$(\mathbf{D}^* \hat{\mathbf{r}})^\top \mathbf{F}^\top \mathbf{u} = (\mathbf{F} \mathbf{D}^* \hat{\mathbf{r}})^\top \mathbf{u}$$

Indistinguishability First, we want to focus on property (1) that says that the re-randomization of an honest encryption is identically distributed to a fresh encryption.

Let $\mathbf{C} := ([\mathbf{x}]_1, [y]_1, \pi)$ be an encryption of $[\mathbf{M}]_1$ with randomness fixed to \mathbf{r} , and let $\hat{\mathbf{C}} := ([\hat{\mathbf{x}}]_1, [\hat{y}]_T, \hat{\pi}) \leftarrow \text{Rand}(\text{pk}, \mathbf{C})$ be its re-randomization with randomness $\hat{\mathbf{r}}$. We show that $\hat{\mathbf{C}}$ is identically distributed to a fresh encryption of $[\mathbf{M}]_1$ with randomness $(\mathbf{r} + \hat{\mathbf{r}})$. Notice that for any vector \mathbf{r} , the random variable $(\mathbf{r} + \hat{\mathbf{r}})$ is uniformly distributed.

It is straightforward to verify that this holds for the element $[\hat{\mathbf{x}}]_1$, indeed:

$$\hat{\mathbf{x}} = \mathbf{D}^*(\mathbf{r} + \hat{\mathbf{r}}) + (\mathbf{0}^\top, \mathbf{M})^\top.$$

The proof that $[\hat{y}]_T$ is correctly distributed, follows easily from Lemma 5. What is left to prove is that the element $\hat{\pi}$ is also correctly distributed. We recall that $\hat{\pi}$ is the result of $\text{PEvl}(ppk, [\mathbf{D} \mathbf{r}]_1, \pi, \hat{\mathbf{r}})$. Because of the correctness of the proof system \mathbf{PS} , it holds that $\hat{\pi}$ is identical to $\text{PPrv}(ppk, [\mathbf{D}(\mathbf{r} + \hat{\mathbf{r}})]_1, (\mathbf{r} + \hat{\mathbf{r}}))$ that is exactly the distribution of $[y]_T$ computed by $\text{Enc}(ppk, \mathbf{M}; (\mathbf{r} + \hat{\mathbf{r}}))$.

Correctness Next, we prove the second property of Definition 10, namely that for any ciphertext \mathbf{C} , it holds that $\text{Dec}(\text{sk}, \mathbf{C}) = \text{Dec}(\text{sk}, \mathbf{C}')$, where \mathbf{C}' is the output of a valid rerandomization of \mathbf{C} , with some randomness $\hat{\mathbf{r}}$.

First, we notice that Rand adds to \mathbf{x} an encryption of $[0]_1$, therefore if the third line of $\text{Dec}(\text{sk}, \mathbf{C})$ computes $[\mathbf{M}]_1$, the same happens in $\text{Dec}(\text{sk}, \mathbf{C}')$. Second, if \mathbf{C} is valid, then the correctness follows from the proof given above: \mathbf{C}' is also valid and $\text{Dec}(\text{sk}, \mathbf{C}) = \text{Dec}(\text{sk}, \mathbf{C}')$. Finally, we are left with proving that if $\text{Dec}(\text{sk}, \mathbf{C}) = \perp$, then also $\text{Dec}(\text{sk}, \mathbf{C}')$ returns \perp . Assume by contradiction that $\text{Dec}(\text{sk}, \mathbf{C}) = \mathbf{M} \neq \perp$. Then it must be true that $[\hat{y}]_T = (\mathbf{f} + \mathbf{F} \hat{\mathbf{x}})^\top [\hat{\mathbf{u}}]_1$ and $\hat{\pi} = \mathbf{k}^\top ([\hat{\mathbf{u}}]_1 \otimes [\hat{\mathbf{u}}]_1)$. For the same proof given above (going from bottom to the top of the equations), and for the correctness of the benign proof system \mathbf{PS} , we obtain that $[y]_T = (\mathbf{f} + \mathbf{F} \mathbf{x})^\top [\mathbf{u}]_1$, $\pi = \mathbf{k}^\top ([\mathbf{u}]_1 \otimes [\mathbf{u}]_1)$, and thus $\text{Dec}(\text{sk}, \mathbf{C}) = \mathbf{M} \neq \perp$, that results in a contradiction.

Tightness of Decryption The last property we need to prove is the third one of Definition 10. We observe that, given a vector $[\mathbf{u}]_1$, there exists a unique value $\pi := \mathbf{k}^\top([\mathbf{u}]_1 \otimes [\mathbf{u}]_1)$ such that $\text{PVer}(psk, [\mathbf{u}]_1, \pi) = 1$. Assume there exists an adversary who is able to produce a ciphertext that does not decrypt to \perp , yet it is not in the encryption range of the scheme, namely there exists no message \mathbf{M} , and no randomness \mathbf{r} such that $\text{Enc}(\text{pk}, \mathbf{M}; \mathbf{r})$ outputs \mathbf{C} . Given the above observation, it must be the case that the first component of the ciphertext, i.e. $[\mathbf{u}]_1$, is not in the span of $[\mathbf{D}]_1$: in [FFHR19] it is proven that the probability that an adversary succeeds in forging a valid $[y]_T$ in such case, is lower or equal than $\frac{1}{q}$.

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 $\mathcal{B}^{\text{mddh}}$, \mathcal{B}^{snd} with similar running time $T(\mathcal{B}^{\text{mddh}}) \approx T(\mathcal{B}^{\text{snd}}) \approx T(\mathcal{A}) + (Q_{\text{Enc}} + Q_{\text{Dec}}) \cdot \text{poly}(\lambda)$, where $\text{poly}(\lambda)$ is a polynomial independent of $T(\mathcal{A})$, and such that*

$$\begin{aligned} \text{Adv}_{\mathcal{A}, \mathcal{PK}\mathcal{E}}^{\text{RCCA}}(\lambda) &\leq O(d \log Q_{\text{Enc}}) \cdot \text{Adv}_{\mathbf{G}_1, \mathcal{D}_{n,d}, \mathcal{B}^{\text{mddh}}}^{\text{MDDH}}(\lambda) \\ &\quad + \log Q_{\text{Enc}} \cdot \text{Adv}_{\mathcal{B}^{\text{snd}}, \mathbf{PS}}^{\text{snd}}(\lambda) + O\left(\frac{n^2 Q_{\text{Dec}} Q_{\text{Enc}} \log Q_{\text{Enc}}}{q}\right). \end{aligned}$$

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 generalized⁸ to the multi-user case almost equivalently to [BBM00, GHK17].

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 $\text{Exp}_{\mathcal{A}, \mathcal{PK}\mathcal{E}}^{\text{mRCCA}}$ 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 $\mathbf{C}_j = ([\mathbf{x}_j], [y_j]_T, \pi_j)$ as described by the encryption procedure, 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 $[p_j] \leftarrow \mathbf{a}^\top[\mathbf{u}_j] + [\mathbf{M}_j, b^*]$. By linearity, this game is perfectly equivalent to the previous one, thus $\epsilon_1 = \epsilon_0$.

⁸ 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 .

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 $[\mathbf{D}_0]$. To be formal, we build adversaries $\mathcal{B}, \mathcal{B}'$ such that for a polynomial $p(\lambda)$ independent of $T(\mathcal{A})$, we have $T(\mathcal{B}) \approx T(\mathcal{B}') \approx T(\mathcal{A}) + (Q_{\text{Enc}} + Q_{\text{Dec}}) \cdot p(\lambda)$ and

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

Let $([\mathbf{D}], [\mathbf{V}])$ be the Q_{Enc} -fold $\mathcal{D}_{n,d}$ -MDDH challenge received by \mathcal{B} , with $[\mathbf{D}] \in \mathbb{G}_1^{n \times d}$ and $[\mathbf{V}] := ([\mathbf{v}_1], \dots, [\mathbf{v}_{Q_{\text{Enc}}}]) \in \mathbb{G}_1^{n \times Q_{\text{Enc}}}$. Then, \mathcal{B} samples the secret material $\mathbf{a} \leftarrow \mathbb{Z}_q^n$, $\mathbf{f} \leftarrow \mathbb{Z}_q^n$, $\mathbf{F} \leftarrow \mathbb{Z}_q^{n \times n+1}$, generates the keys for the benign proof system, running $\text{PGen}([\mathbf{D}])$, and finally sends the public key to \mathcal{A} . On the j -th query to the encryption oracle, \mathcal{B} sets

$$\begin{aligned} [\mathbf{u}_j] &\leftarrow [\mathbf{v}_j] \\ [y_j]_T &\leftarrow \mathbf{f}^\top [\mathbf{u}_j]_T + [\mathbf{x}_j]^\top \mathbf{F}^\top [\mathbf{u}_j] \end{aligned}$$

\mathcal{B} has generated the secret key itself, so it can perfectly simulate the decryption oracle. In case $[\mathbf{V}] = [\mathbf{D}\mathbf{R}]$, \mathcal{B} perfectly simulates \mathbf{G}_2 . In case $[\mathbf{V}]$ is uniformly random over $\mathbb{G}_1^{n \times Q_{\text{Enc}}}$, \mathcal{B} simulates an intermediary game \mathbf{H} . Analogously, we can build an adversary \mathcal{B}' on the Q_{Enc} -fold $\mathcal{D}_{n,2d}$ -MDDH assumption, who simulates \mathbf{H} if $[\mathbf{V}]$ is uniformly random over $\mathbb{G}_1^{n \times Q_{\text{Enc}}}$, and game \mathbf{G}_3 if $[\mathbf{V}] = [\mathbf{D}_0\mathbf{R}]$. Altogether, this proves the claim stated above: from Lemma 1 and Corollary 1, we obtain an adversary \mathcal{B}'' such that $T(\mathcal{B}'') \approx T(\mathcal{A}) + (Q_{\text{Enc}} + Q_{\text{Dec}}) \cdot \text{poly}(\lambda)$ where poly is independent of $T(\mathcal{A})$ and

$$|\epsilon_3 - \epsilon_2| \leq 2(n-d) \mathbf{Adv}_{\mathbb{G}_1, \mathcal{D}_{n,d}, \mathcal{B}''}^{\text{mddh}}(\lambda) + \frac{2}{q-1}.$$

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. 4 as the matrix whose first n rows are equal to the matrix \mathbf{D} and last row is equal to $\mathbf{a}^\top \mathbf{D}$. Upon query $\mathbf{c} := ([\mathbf{x}], [y]_T, \pi)$ to the decryption oracle, where $[\mathbf{x}]^\top := ([\mathbf{u}]^\top, [p])$, the oracle additionally checks that:

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

where $\mathbf{D}^{*\perp}\mathbf{D}^* = 0$, and $\mathcal{Q}_{\text{Enc}} = \{\mathcal{C}_j = ([\mathbf{x}_j], [y_j]_T, \pi_j) : j \leq [\text{ctr}]\}$ 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:

- $[\mathbf{u}] \notin \text{span}([\mathbf{D}])$ and $\forall j : \mathbf{D}^{*\perp}\mathbf{x}_j \neq \mathbf{D}^{*\perp}\mathbf{x}$ (the latter condition implies that $[\mathbf{u}]$ is not the result of an honest rerandomization of a previous challenge ciphertext)
- the proof 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.

In other words, a critical decryption query consists of a ciphertext that does not decrypt to \perp (according to the original decryption rules), is not a rerandomization of a challenge ciphertext, but has a $[\mathbf{u}]$ that could not have been generated in an “honest” encryption. For this step, we refer to Lemma 6.

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 $M_{j,0}$ and $M_{j,1}$ be the challenge messages queried by \mathcal{A} at the j -th query to the encryption oracle. Upon decryption oracle query $\mathcal{C} = ([\mathbf{x}], [y]_T, \pi)$, if $\exists j : \mathbf{D}^{*\perp}\mathbf{x}_j = \mathbf{D}^{*\perp}\mathbf{x}$ where recall $\mathcal{Q}_{\text{Enc}} = \{([\mathbf{x}_j], [y_j], \pi_j) : j \leq \text{ctr}\}$, and both the proof π verifies and the consistency check holds, then the decryption oracle immediately returns the symbol $\diamond_{\mathcal{J}}$ where $\mathcal{J} \leftarrow \mathcal{Q}.\text{find}(M_{j,0})$.

Notice that we can rewrite the decryption procedure as $\mathbf{M} = (-\mathbf{a}^\top, 1)[\mathbf{x}]$. 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 $\mathbf{D}^{*\perp}\mathbf{x}_j = \mathbf{D}^{*\perp}\mathbf{x}$ for some challenge ciphertext \mathcal{C}_j then it holds that $(-\mathbf{a}^\top, 1)[\mathbf{x}_j] = (-\mathbf{a}^\top, 1)[\mathbf{x}]$, and therefore the decryption oracle would compute the message M_{j,b^*} and output the symbol $\diamond_{\mathcal{J}}$, where $\mathcal{J} = \mathcal{Q}.\text{find}(M_{j,b^*})$. Moreover, notice that $\mathcal{Q}.\text{find}(M_{j,b^*}) = \mathcal{Q}.\text{find}(M_{j,0})$ 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 $M_{j,0}, M_{j,1}$) encrypts the message $M_{j,b^*} + \mathbf{R}_j$, where \mathbf{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 [GHK17], we defer its proof to Appendix B.1.

Lemma 6. *The security games \mathbf{G}_3 and \mathbf{G}_4 defined for the proof of Theorem 3 (security of the RCCA symmetric scheme, see Fig. 4) are computationally indistinguishable. For any PPT adversary \mathcal{A} , we build PPT adversaries $\mathcal{B}, \mathcal{B}'$ with*

running times similar to \mathcal{A} such that:

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

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 Experiments $\mathbf{H}_{1,i,x}$	Oracle $\mathcal{O}_{\text{dec}}(\mathcal{C})$
<pre> prn \leftarrow Setup(1^λ) $\mathbf{D} \leftarrow \mathcal{D}_{n,d}$, $\mathbf{D}^* \leftarrow (\mathbf{D}^\top, (\mathbf{a}^\top \mathbf{D})^\top)^\top$ $\mathbf{a} \leftarrow \mathbb{Z}_q^n$, $\mathbf{f} \leftarrow \mathbb{Z}_q^n$, $\mathbf{F} \leftarrow \mathbb{Z}_q^{n \times n+1}$ $\text{pars} \leftarrow \text{PGen}(\text{prm}_G, ([\mathbf{D}]_1))$ $\text{pk} \leftarrow ([\mathbf{f}^\top \mathbf{D}], [\mathbf{F}^\top \mathbf{D}]_1, [\mathbf{F} \mathbf{D}^*]_1, \text{ppk})$ $\text{sk} \leftarrow (\mathbf{a}, \mathbf{f}, \mathbf{F}, \text{psk})$ $\mathcal{Q} \leftarrow \text{DisjointSet}()$ $\mathcal{Q}_{\text{Enc}} \leftarrow []$ $\text{ctr} \leftarrow 0$ $b^* \leftarrow \{0, 1\}$ critical $\leftarrow 0$ $\mathcal{A}^{\mathcal{O}_{\text{Enc}}, \mathcal{O}_{\text{Dec}}}(\text{pk})$ return critical </pre>	<pre> parse \mathcal{C} as $([\mathbf{x}]_1, [y]_T, \pi, [\mathbf{x}^\top]_1)$ as $([\mathbf{u}^\top]_1, [p]_1)$ <div style="border: 1px dashed black; padding: 5px; margin: 5px 0;"> $\mathbf{H}_{1,i,2} - \mathbf{H}_{1,i,3}$ if $[\mathbf{u}]_1 \notin \text{span}([\mathbf{D}_0]_1) \cup \text{span}([\mathbf{D}]_1)$: return \perp </div> if $\text{PVer}(\text{psk}, [\mathbf{u}]_1, \pi) \stackrel{?}{=} 0$ return \perp parse \mathcal{Q}_{Enc} as $\{([\mathbf{x}_j]_1, [y_j]_T, \pi_j) : j \in [\text{ctr}]\}$ if $\exists j : \mathbf{D}^{*\perp} \mathbf{x}_j \stackrel{?}{=} \mathbf{D}^{*\perp} \mathbf{x}$: $\mathcal{I} \leftarrow \{j : \mathbf{D}^{*\perp} \mathbf{x}_j \stackrel{?}{=} \mathbf{D}^{*\perp} \mathbf{x}\}$ else $\mathcal{I} \leftarrow \{j : j \leq \text{ctr}\}$ $\mathcal{S} \leftarrow$ <div style="border: 1px dashed black; padding: 5px; margin: 5px 0;"> Decryption Keys <hr/> $\mathbf{H}_{1,i,0} - \mathbf{H}_{1,i,2} : \{j_i\}_{j \in \mathcal{I}}$ $\mathbf{H}_{1,i,3} - \mathbf{H}_{1,i,4} : \{j_i \ d\}_{j \in \mathcal{I}}$ $\mathbf{H}_{1,i,5} : \{j_i \ b\} : b \in \{0, 1\}_{j \in \mathcal{I}}$ $\mathbf{H}_{1,i,6} : \{j_{i+1}\}_{j \in \mathcal{I}}$ </div> for \mathbf{k} in \mathcal{S} : $[y']_T \leftarrow$ <div style="border: 1px dashed black; padding: 5px; margin: 5px 0;"> Consistency Check <hr/> $\mathbf{H}_{1,i,0} - \mathbf{H}_{1,i,2} :$ $(\mathbf{f}^\top + P_i(\mathbf{k}) \mathbf{D}^\perp)[\mathbf{u}]_1 + [\mathbf{x}]_1^\top \mathbf{F}^\top [\mathbf{u}]_1$ $\mathbf{H}_{1,i,3} - \mathbf{H}_{1,i,6} :$ $(\mathbf{f}^\top + P_{i+1}(\mathbf{k}) \mathbf{D}^\perp)[\mathbf{u}]_1 + [\mathbf{x}]_1^\top \mathbf{F}^\top [\mathbf{u}]_1$ </div> $b_1 \leftarrow [y']_T \stackrel{?}{=} [y]_T$ if $[\mathbf{u}]_1 \notin \text{span}([\mathbf{D}]) \wedge b_1 \stackrel{?}{=} 1$: critical $\leftarrow 1$ abort if $b_1 \neq 1$: return \perp $\mathbf{M} \leftarrow [p]_1 - \mathbf{a}^\top [\mathbf{u}]$ $\mathcal{J} \leftarrow \mathcal{Q}.\text{find}(\mathbf{M})$ if $\mathcal{J} \neq \perp$: return $\diamond_{\mathcal{J}}$ else return \mathbf{M} </pre>
<pre> Oracle $\mathcal{O}_{\text{Enc}}(\mathbf{M}_0, \mathbf{M}_1)$ <hr/> $\text{ctr} \leftarrow \text{ctr} + 1$ $\mathcal{Q}.\text{union}(\{\mathbf{M}_0, \mathbf{M}_1\})$ $\mathbf{r} \leftarrow \mathbb{Z}_q^d$ <div style="border: 1px dashed black; padding: 5px; margin: 5px 0;"> $[\mathbf{u}]_1 \leftarrow$ <hr/> Sample $[u]_1$ <hr/> $\mathbf{H}_{1,i,0} :$ $[\mathbf{D}_0]_1 \cdot \mathbf{r}$ <hr/> $\mathbf{H}_{1,i,1} - \mathbf{H}_{1,i,6} :$ $[\mathbf{D}_{\text{ctr}[i+1]}]_1 \cdot \mathbf{r}$ </div> $[p]_1 \leftarrow \mathbf{a}^\top [\mathbf{u}] + \mathbf{M}_{b^*}, [\mathbf{x}]_1 \leftarrow ([\mathbf{u}^\top]_1, [p]_1)^\top$ $[y]_T \leftarrow (\mathbf{f}^\top + P_i(\text{ctr}_i) \mathbf{D}^\perp)[\mathbf{u}] + [\mathbf{x}]^\top \mathbf{F}^\top [\mathbf{u}]$ $\pi \leftarrow \text{PSim}(\text{psk}, [\mathbf{u}]_1)$ $\mathcal{C} \leftarrow ([\mathbf{x}]_1, [y]_T, \pi)$ add \mathcal{C} to \mathcal{Q}_{Enc} and return \mathcal{C} </pre>	

Fig. 5: The Security Experiments $\mathbf{H}_{1,i,x}$. We highlight the differences between the hybrids using dashed boxes: inside the boxes we describe the involved hybrids and the operations performed.

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 \mathbf{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 the matrix \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{zD}^\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:

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

- Similarly, the decryption oracle, upon input the ciphertext $\mathbf{C} = ([\mathbf{x}], [y]_T, \pi)$ 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:

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

These new rules do not change the view of the adversary since both \mathbf{f}^\top and $\mathbf{f}^\top + \mathbf{zD}^\perp$ 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 $P : \{0, 1\}^* \rightarrow \mathbb{Z}_q^{1 \times n-d}$ be a 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:

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

- 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 adversary⁹.

Recall that ctr counts the number of challenge ciphertexts output by the encryption oracle and that $\mathcal{Q}_{\text{Enc}} = \{\mathbf{C}_j = ([\mathbf{x}_j], [y_j]_T, \pi_j) : j \leq \text{ctr}\}$. Upon

⁹ The reader might have notice that this is where our proof strategy needs to differentiate from the original partitioning technique from [GHK17]. In particular, we conclude the proof of Lemma 6 (see pag. 33) by reducing to Lemma 4, the reduction will know all the secret keys but one for a (uniformly sampled) random index.

input the ciphertext $\mathbf{C} = ([\mathbf{x}], [y]_T, \pi)$, the decryption oracle first sets:

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

then it computes the bit b_1 (i.e. the bit of the consistency check for \mathbf{C} , see Fig. 4) differently by checking that

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

Moving from \mathbf{H}_1 to \mathbf{H}_2 requires a series of hybrids $\mathbf{H}_{1,i,i'}$, $i \in [\log(Q_{\text{Enc}})]$, $i' \in [6]$. We give in Fig. 5 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

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

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

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

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 -th 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 $([\bar{\mathbf{D}}_0], [\mathbf{h}_1], \dots, [\mathbf{h}_{Q_{\text{Enc}}}]$) 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 $\text{PGen}([\mathbf{D}])$). 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 (M_0, 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, and considering Lemma 1 and Corollary 1, we obtain an adversary \mathcal{C} such that:

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

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 $\mathbf{u} \notin \text{span}(\mathbf{D}_0) \cup \text{span}(\mathbf{D}_1)$, 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 and the reduction is standard. 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 $\mathbf{C} = ([\mathbf{x}], [y]_T, \pi)$ where:

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

For such query it holds that $\text{PVer}(psk, [\mathbf{u}], \pi) = 1$. 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 $\mathcal{L}^{\text{snd}} = \mathcal{L}^{\text{sim}} = \text{span}(\mathbf{D}_0) \cup \text{span}(\mathbf{D}_1)$, 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 $j \in [Q_{\text{Enc}}]$. 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}, π) to the challenger. Since \mathcal{L}^{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 $\mathbf{u} \notin \text{span}(\mathbf{D}_0) \cup \text{span}(\mathbf{D}_1)$, i.e., $\mathbf{u} \notin \mathcal{L}^{\text{snd}}$, but $\text{PVer}(\text{psk}, [\mathbf{u}], \pi) = 1$. This is the event that lets \mathcal{B} win the soundness game. The adversary \mathcal{B} runs in time $T(\mathcal{B}) \approx T(A) + (Q_{\text{Enc}} + Q_{\text{Dec}}) \cdot \text{poly}(\lambda)$, where 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:

$$|\epsilon_{\mathbf{H}_{1,i,2}} - \epsilon_{\mathbf{H}_{1,i,1}}| \leq \text{Adv}_{\mathcal{B}, \text{PS}}^{\text{snd}}(\lambda).$$

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 $\mathbf{C} = ([\mathbf{x}], [y]_T, \pi)$ additionally checks that $\exists d$ s.t. $\mathbf{u} \in \text{span}(\mathbf{D}_d)$ and in such case it sets:

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

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

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

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 $\mathbf{u}_j \in \text{span}(\mathbf{D}_d)$ (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:

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

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 $\mathbf{D}_{j[|i+1]}^\perp \mathbf{D}_{j[|i+1]} = 0$, 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 $\mathbf{C} = ([\mathbf{x}], [y]_T, \pi)$ where $\exists d : \mathbf{u} \in \text{span}(\mathbf{D}_d)$, we have that $P'_{i+1}(j_{|i+1})\mathbf{u} = P'_i(j_{|i})\mathbf{u}$. The derivation is identical as 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 $\text{span}(\bar{\mathbf{D}}_0 || \bar{\mathbf{D}}_1)$ has rank less than $2d$, which happens with probability at most $1/q$. Thus we can conclude that $|\epsilon_{\mathbf{H}_{1,i,2}} - \epsilon_{\mathbf{H}_{1,i,3}}| \leq \frac{2}{q}$.

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

$$\epsilon_{\mathbf{H}_{1,i,3}} \leq \epsilon_{\mathbf{H}_{1,i,4}}.$$

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 .

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

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 $[\mathbf{u}] \in \text{span}([\mathbf{D}])$: 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 $j \in [Q_{\text{Enc}}]$ such that $\mathbf{D}^{*\perp} \mathbf{x}_j$ is equal to $\mathbf{D}^{*\perp} \mathbf{x}$. If a query of this form successfully 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}_{dec} and \mathcal{O}_{enc} . By union bound, we obtain a statistical distance of $O(Q_{\text{Enc}}Q_{\text{Dec}}/q)$.

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

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:

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

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 $(j_i \| 1) \cdot 2^{|\log Q_{\text{Enc}}| - i - 1}$ 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:

$$|\epsilon_{\mathbf{H}_{1,i,6}} - \epsilon_{\mathbf{H}_{1,i,5}}| \leq \frac{Q_{\text{Enc}} \cdot Q_{\text{Dec}}}{q - Q_{\text{Dec}}}.$$

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 an adversary \mathcal{C} such that:

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

It is easy to see that $\epsilon_{\mathbf{H}_2} = \epsilon_{\mathbf{H}_{1, \lceil \log Q_{\text{Enc}} \rceil, 6}}$. Next, we prove that $\epsilon_{\mathbf{H}_2} \leq \frac{O(n^2)Q_{\text{Enc}}Q_{\text{Dec}}}{q}$. We reduce the adversary \mathcal{A} playing in \mathbf{H}_2 to an (unbounded) adversary \mathcal{B} upon which we can invoke the Lemma 4. 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 4 the matrix \mathbf{E} to be set equal to \mathbf{D}^* .

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

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 $j_{\text{Enc}}^* \in [Q_{\text{Enc}}]$ and an index $j_{\text{Dec}}^* \in [Q_{\text{Dec}}]$. (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 $j \neq j_{\text{Enc}}^*$, the reduction \mathcal{B} generates \mathbf{x}_j following the prescribed algorithms. Then, it computes $y_j \leftarrow \left((\bar{\mathbf{f}} + \bar{\mathbf{F}}\mathbf{x}_j)^\top + P(j)\mathbf{D}^\perp \right) \mathbf{u}_j$, where we recall that $P(\cdot)$ is a random function.
- Else, for $j = j_{\text{Enc}}^*$, \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 $\mathbf{C} = ([\mathbf{x}], [y]_T, \pi)$ there are three possible cases. The easiest case to handle is if $\mathbf{u} \in \text{span}(\mathbf{D})$ or $\exists j \neq j_{\text{Enc}}^*$ 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 $\mathbf{D}^{*\perp} \mathbf{x}_{j_{\text{Enc}}^*} = \mathbf{D}^{*\perp} \mathbf{x}$, in this case let \mathbf{r}' be such that $\mathbf{x} - \mathbf{x}_{j_{\text{Enc}}^*} = \mathbf{D}^* \mathbf{r}'$ and compute

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

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

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

By Lemma 5 and by definition of $y_{j_{\text{Enc}}^*}$, the two checks are equivalent. The last case is when $\mathbf{u} \notin \text{span}(\mathbf{D}) \wedge \forall j : \mathbf{D}^{*\perp} \mathbf{x}_j \neq \mathbf{D}^{*\perp} \mathbf{x}$, i.e., the query might be “critical”:

- If $j < j_{\text{Dec}}^*$ 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 $j = j_{\text{Dec}}^*$ then output the tuple $(y - P(j_{\text{Enc}}^*)\mathbf{D}^\perp \mathbf{u}, \mathbf{u}, \mathbf{x})$ 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 $\mathbf{C} = ([\mathbf{x}], [y]_T, \pi)$ we have that the equation $[y]_T = (\mathbf{f} + P(j_{\text{Enc}}^*)\mathbf{D}^\perp + \mathbf{F}\mathbf{x})^\top [\mathbf{u}]$ holds. Let **Guess** be such event. Conditioned on such a lucky event, the adversary \mathcal{B} indeed produces a valid forgery, in fact by the definition of a critical query $(\mathbf{x}_{j_{\text{Enc}}^*} - \mathbf{x}) \notin \text{span}(\mathbf{D}^*)$ and $\mathbf{u} \notin \text{span}(\mathbf{D})$.

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 **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:

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

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 $j \neq j_{\text{Enc}}^*$ the value y computed at the j -th encryption oracle query is:

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

The probability that the reduction \mathcal{B} creates a forgery is $\Pr[\mathbf{H}_2 = 1 \wedge \text{Guess}]$, and the two events are independent. Moreover, since $\Pr[\text{Guess}] = (Q_{\text{Enc}}Q_{\text{Dec}})^{-1}$, by Lemma 4 we have that $\Pr[\mathbf{H}_2 = 1] \leq \frac{n(n+1)Q_{\text{Enc}}Q_{\text{Dec}}}{q}$.

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 [FFHR19], we append a malleable NIZK proof (essentially a Groth-Sahai proof) that $[y]_T$ and π are well-formed to the ciphertexts of $\mathcal{PK}\mathcal{E}$ 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 $\mathcal{PK}\mathcal{E}_1 = (\text{KGen}_1, \text{Enc}_1, \text{Dec}_1, \text{Rand}_1)$ be the scheme of Section 5 instantiated using the benign proof system of Section 3.1, and let $\mathbf{PS}_2 = (\text{PGen}_2, \text{PPrv}_2, \text{PVer}_2)$ and PEvl_2 form a malleable NIZK system for membership in the relation

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

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

The pv-Rand-PKE scheme $\mathcal{PK}\mathcal{E}_2 = (\text{Init}, \text{KGen}_2, \text{Enc}_2, \text{Dec}_2, \text{Rand}_2, \text{Ver})$ is identical to $\mathcal{PK}\mathcal{E}_1$, except that

- KGen_2 additionally samples the common reference string for \mathbf{PS}_2 ,
- the encryption procedure computes a ciphertext as in $\mathcal{PK}\mathcal{E}_1$ but additionally computes a proof π_2 for \mathbf{PS}_2 and outputs a ciphertext $\mathbf{C} = ([\mathbf{x}_1], \pi_2)$,
- the decryption procedure first checks the proof π_2 holds w.r.t. the instance $(\text{pk}, [\mathbf{x}])$ and, if so, it outputs $\mathbf{M} = (-\mathbf{a}^\top, 1)[\mathbf{x}]$ (and \perp otherwise),
- the re-randomization procedure randomizes $[\mathbf{x}]$ as in $\mathcal{PK}\mathcal{E}_1$ and uses PEvl_2 for the remaining part of the ciphertext, and
- Ver_2 simply checks the proof π_2 .

Theorem 4. *If $\mathcal{PK}\mathcal{E}_1$ is mRCCA secure, and \mathbf{PS}_2 is adaptively sound, $(\epsilon, O(T))$ -composable zero-knowledge, and perfect derivation private, then the pv-Rand-PKE scheme $\mathcal{PK}\mathcal{E}_2$ 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 $\mathcal{B}^{\text{rcca}}$ making the same number of queries and adversaries $\mathcal{B}^{\text{snd}}, \mathcal{B}^{\text{zk}}$ with similar running times*

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

The proof of the theorem follows by inspection of the proof of Theorem 2 in [FFHR19]. 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 by the proof system 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 [FFHR19]. In more detail, [FFHR19] 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 [EHK⁺13] and, given their nice algebraic properties they possess malleability property [CKLM12]. The verification equation uses a special projecting bilinear map $\tilde{e} : \mathbb{G}_1^{k+1} \times \mathbb{G}_2^{k+1} \rightarrow \mathbb{G}_T^m$. In general, the map \tilde{e} with the optimal m depends on \mathcal{D}_k (not only on k), as was proven [HHH⁺14]. Groth-Sahai Proofs are perfect adaptive sound, however we also make use of a QA-NIZK for linear spaces [KW15] which is computationally adaptive sound. Fortunately, the language to be proved is witness samplable, thus the tight-security of $\mathcal{PK}\mathcal{E}_2$ holds. Recall that we need a NIZK for the following relationship:

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

We follow a similar approach to [FFHR19], specifically the proof includes:

1. Commitments to $[y]_T$ and commitments to $[\mathbf{u} \otimes \mathbf{u}]_T$.
2. A commitment $[\mathbf{c}_0]_1 \in \mathbb{G}_1^3$ to $[\mathbf{f}^\top \mathbf{D} \mathbf{r}]_1$ and commitments $[\mathbf{c}_1]_1, \dots, [\mathbf{c}_{n+1}]_1$ to the components of the vector $[\mathbf{F}^\top \mathbf{D}^* \mathbf{r}]_1$.
3. Commitments $[\mathbf{c}_{n+2}]_1, \dots, [\mathbf{c}_{n+2+nd}]_1$ to the components of the vector $[\mathbf{u} \otimes \mathbf{r}]_1$.
4. An (extended) GS proof of the equation $[y]_T = [1]_1 \cdot [\mathbf{f}^\top \mathbf{D} \mathbf{r}]_1 + [\mathbf{x}^\top]_1 \cdot [\mathbf{F}^\top \mathbf{D} \mathbf{r}]_1$.
5. An (extended) GS proof of the equation $[\mathbf{k}^\top \mathbf{D} \otimes \mathbf{I}]_1 \cdot [\mathbf{u} \otimes \mathbf{r}]_1 = \pi$.
6. n^2 (extended) GS proofs of the equations $[\mathbf{I} \otimes \mathbf{D}]_1 \cdot [\mathbf{u} \otimes \mathbf{r}]_1 = [\mathbf{u} \otimes \mathbf{u}]_T$.
7. A proof that the commitments $[\mathbf{c}_0]_1, \dots, [\mathbf{c}_{n+1}]_1$ are well formed. This can be proven with one proof of membership in a linear spaces in each group [JR14, KW15]. With more detail, the proof shows that $[\mathbf{u}]_1 = [\mathbf{D}]_1 \mathbf{r}$ and that the commitments commit to \mathbf{r} using the basis $[\mathbf{f}^\top \mathbf{D}]_1$ and $[\mathbf{F}^\top \mathbf{D}]_1$ to commit.

As the reader might have noticed, the proof is quite expensive, as it contains more than mn^2 elements in \mathbb{G}_T . In the next paragraph we show how to reduce the size of the proof relying on a stronger cryptographic assumption.

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 $[\mathbf{A} \otimes \mathbf{A}]$ in order to distinguish an element from $\text{span}([\mathbf{A}])$ and random.

Definition 12 (Tensor Matrix Diffie-Hellman assumption in \mathbb{G}_γ). *The $\mathcal{D}_{\ell,k}$ -Tensor-Matrix-Decisional-Diffie-Hellman (TMDDH) assumption in group \mathbb{G}_γ holds if for all non-uniform PPT adversaries \mathcal{A} ,*

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

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

The TMDDH assumption can be seen as a generalization of the “square-Diffie-Hellman” assumption [BDS98, MW96], and as a special case of the “Uber assumption family” [Boy08]. 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} .

To explain what we mean by “holds generically”, we give a brief explanation of the generic group model (in the formulation of Maurer [Mau05]). A *generic adversary* on assumptions in a group only interacts through an oracle with the investigated group. Concretely, this adversary initially gets only the group order and so-called *handles* (i.e., running numbers) to the challenge group elements. The adversary additionally gets access to oracles that allow to test group elements for zero (which also allows testing for equality), perform the group operation on two handles (which yields a new handle to the resulting group element), and potentially to other operations involving group elements (such as a pairing).

It is possible to show that relative to generic adversaries, the Diffie-Hellman and discrete logarithm assumptions hold [Nec94, Sho97]. In fact, the $\mathcal{U}_{k+1,k}$ -MDDH assumption even holds against generic adversaries with a symmetric k -linear map [EHK⁺13]. We remark that this last fact implies that also the $\mathcal{U}_{k+1,k}$ -TMDDH assumption holds generically, even in the presence of a (symmetric) pairing.

Lemma 7 (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.*

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

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

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

- It does not include the commitments to $[\mathbf{u} \otimes \mathbf{r}]$ but instead it includes commitments $[\mathbf{c}_{n+2}], \dots, [\mathbf{c}_{n+2+d^2}]$ to the components of the vector $\mathbf{r} \otimes \mathbf{r}$.
- It does not contain the (extended) GS proofs of step 5 and 6 but instead it contains a (standard) GS proof for the equations $\mathbf{k}^\top [\mathbf{D} \otimes \mathbf{D}] (\mathbf{r} \otimes \mathbf{r}) \cdot [1] = \pi$.

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

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

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 \mathbb{G}_3 to \mathbb{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 [FFHR19] because their protocol works for any pv-Rand RCCA scheme that has the property of being *linear* and a property that holds for both $\mathcal{PK}\mathcal{E}_2$ and $\mathcal{PK}\mathcal{E}_3$.

6.1 Linear pv-Rand PKE

Definition 13 (Linear pv-Rand-RCCA PKE, [FFHR19]). *We say that a pv-Rand-RCCA PKE scheme is linear if there exist a group \mathbb{G} (for example $\mathbb{G} = \mathbb{G}_1$) and parameters $\ell, \ell', \ell'' \in \mathbb{N}$ such that (1) every key pair (pk, sk) we can parse $\text{pk} = ([\mathbf{P}], \hat{\text{pk}})$ and $\text{sk} = (\mathbf{S}, \hat{\text{sk}})$, where $[\mathbf{P}] \in \mathbb{G}^{\ell \times \ell''}$ and $\mathbf{S} \in \mathbb{Z}_q^{\ell' \times \ell}$, (2) any ciphertext $\mathbf{C} \in \mathcal{C}$ can be parsed as $([\mathbf{y}], \hat{\mathbf{C}})$ where $[\mathbf{y}] \in \mathbb{G}^\ell$, (3) for any ciphertext \mathbf{C} such that $\text{Ver}(\text{pk}, \mathbf{C}) = 1$ the decryption procedure is linear, i.e., we have $\text{Dec}(\text{sk}, \mathbf{C}) = \mathbf{S} \cdot [\mathbf{y}]$ (4) let $\mathbf{C}' = \text{Rand}(\text{pk}, \mathbf{C}; \mathbf{r}, r)$ where $\mathbf{C}' = ([\mathbf{y}'], \hat{\mathbf{C}}')$ be a re-randomization of $\mathbf{C} = ([\mathbf{y}], \hat{\mathbf{C}})$ and $\mathbf{r} \in \mathbb{Z}_q^{\ell''}$ then $([\mathbf{y}] - [\mathbf{y}']) = [\mathbf{P}]\mathbf{r}$.*

Lemma 8. *The pv-Rand-RCCA PKE schemes $\mathcal{PK}\mathcal{E}_2$ and $\mathcal{PK}\mathcal{E}_3$ defined in Section 5.1 are linear schemes.*

Proof. Given the definition of $[\mathbf{P}]_1, \mathbf{S}, \mathbf{y}, \ell, \ell', \ell''$ in Definition 13, we set for both schemes the following:

- $\ell := n + 1, \ell' := 1$ and $\ell'' := d$
- $\mathbf{P} := \mathbf{D}^*$

- $\mathbf{S} := (-\mathbf{a}^\top, 1)$
- $\mathbf{y} := \mathbf{x}$

Property 1 and 2 of Definition 13 easily follow from the above definitions. Property 3 holds because if the verification procedure outputs 1 when given in input the ciphertext \mathbf{C} , the decryption algorithm returns the message $[p]_1 - [\mathbf{a}^\top \mathbf{u}]_1 = (-\mathbf{a}^\top, 1)[\mathbf{x}]_1 = \mathbf{S} \cdot \mathbf{y}$. Property 4 can be easily proved since the difference between $[\hat{\mathbf{x}}]_1$ and $[\mathbf{x}]_1$, where $[\hat{\mathbf{x}}]_1$ is the first component of $\text{Rand}(psk, \mathbf{C}, \hat{\mathbf{r}})$, is equal to $[\mathbf{D}^*]_1 \hat{\mathbf{r}} = [\mathbf{P}]_1 \hat{\mathbf{r}}$.

6.2 The MixNet

The MixNet ideal functionality interacts with n sender parties and m mixer parties. The i -th sender sends the message \mathbf{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.

In the full version of Faonio *et al.* [FFHR19], it is introduced a variant of the UC model for protocols that are auditable. This denotes protocols that additionally include an external off-line party, the auditor. The auditor uses only the transcript of the protocol execution and public parameters to verify the validity of the public outputs of a protocol execution. For simplicity, we omit this refinement of the UC model and we present our result in the standard UC model¹⁰.

The protocol is divided into 3 phases: (i) at input phase, the *sender parties* send pv-Rand-RCCA ciphertexts of their messages and a simulation-extractable¹¹ NIZK of knowledge; (ii) at 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 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 tight secure. Let $\text{Adv}_{\mathcal{A}, \mathbf{PS}}^{\text{sim-ext}}(\lambda)$ 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.

Theorem 6. *Let $\mathcal{PK}\mathcal{E}$ be a linear pv-Rand RCCA PKE, \mathbf{PS} be a simulation-extractable NIZK, and let Π be the MixNet protocol from [FFHR19] instantiated with $\mathcal{PK}\mathcal{E}$ and \mathbf{PS} . The protocol Π realizes \mathcal{F}_{Mix} with setup assumptions a threshold decryption functionality $\mathcal{F}_{\text{Dec}}[\mathcal{PK}\mathcal{E}]$ and a common-reference string*

¹⁰ It is easy to see that every protocol secure in their model is also secure in the standard UC model.

¹¹ Actually, they need a weaker form of soundness called all-but-one soundness, however simulation extractability is sufficient.

functionality \mathcal{F}_{CRS} . More in detail, there exist a simulator S and negligible function $\text{negl}(\lambda, m)$ 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:

$$\begin{aligned} & \left| \Pr[\text{REAL}_{\mathcal{Z}, \Pi}(\lambda) = 1] - \Pr[\{\mathcal{F}_{\text{CRS}}, \mathcal{F}_{\text{TDec}}\}\text{-HYBRID}_{\mathcal{Z}, S}^{\mathcal{F}_{\text{Mix}}}(\lambda) = 1] \right| \\ & \leq 3\text{Adv}_{\mathcal{B}, \mathcal{PK}\mathcal{E}}^{\text{mRCCA}}(\lambda) + \text{Adv}_{\mathcal{B}', \text{PS}'}^{\text{sim-ext}}(\lambda) + \text{negl}(\lambda, m) \end{aligned}$$

We stress that the negligible function $\text{negl}(\lambda, m)$ in the statement of Theorem 6 is independent of the running time of the environment $T_{\mathcal{Z}}$, and only depends on the number of mixers of the protocol (which we can think as a small number). The proof of the theorem follows by inspection of the proof of Theorem 5 in [FFHR19] and observing that the three steps of the proof that reduce to the p-Rand-RCCA security of $\mathcal{PK}\mathcal{E}$ can be performed tightly by relying on the multiphertext RCCA security definition (cf. Definition 9). In Appendix C.1 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.* [AJOR18]. Thus, instantiating the protocol with $\mathcal{PK}\mathcal{E}_2$ (resp. $\mathcal{PK}\mathcal{E}_3$) we obtain a MixNet protocol that reduces almost-tightly in the number of mixed messages to the MDDH (resp. TMDDH) Assumption.

Acknowledgements

This work has been partially supported by the MESRI-BMBF French-German joint project named PROPOLIS (ANR-20-CYAL-0004-01).

References

- ABP15. Michel Abdalla, Fabrice Benhamouda, and David Pointcheval. Disjunctions for hash proof systems: New constructions and applications. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 69–100. Springer, Heidelberg, April 2015.
- AJOR18. Masayuki Abe, Charanjit S. Jutla, Miyako Ohkubo, and Arnab Roy. Improved (almost) tightly-secure simulation-sound QA-NIZK with applications. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part I*, volume 11272 of *LNCS*, pages 627–656. Springer, Heidelberg, December 2018.
- BBM00. Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274. Springer, Heidelberg, May 2000.
- BDS98. Mike Burmester, Yvo Desmedt, and Jennifer Seberry. Equitable key escrow with limited time span (or, how to enforce time expiration cryptographically). In Kazuo Ohta and Dingyi Pei, editors, *ASIACRYPT'98*, volume 1514 of *LNCS*, pages 380–391. Springer, Heidelberg, October 1998.

- BG12. Stephanie Bayer and Jens Groth. Efficient zero-knowledge argument for correctness of a shuffle. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 263–280. Springer, Heidelberg, April 2012.
- Boy08. Xavier Boyen. The uber-assumption family (invited talk). In Steven D. Galbraith and Kenneth G. Paterson, editors, *PAIRING 2008*, volume 5209 of *LNCS*, pages 39–56. Springer, Heidelberg, September 2008.
- CKLM12. Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Sarah Meiklejohn. Malleable proof systems and applications. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 281–300. Springer, Heidelberg, April 2012.
- CKN03. Ran Canetti, Hugo Krawczyk, and Jesper Buus Nielsen. Relaxing chosen-ciphertext security. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 565–582. Springer, Heidelberg, August 2003.
- CS98. Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *CRYPTO’98*, volume 1462 of *LNCS*, pages 13–25. Springer, Heidelberg, August 1998.
- DMS16. Yevgeniy Dodis, Ilya Mironov, and Noah Stephens-Davidowitz. Message transmission with reverse firewalls—secure communication on corrupted machines. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 341–372. Springer, Heidelberg, August 2016.
- EHK⁺13. Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Heidelberg, August 2013.
- FF20. Antonio Faonio and Dario Fiore. Improving the efficiency of re-randomizable and replayable CCA secure public key encryption. In Mauro Conti, Jianying Zhou, Emiliano Casalicchio, and Angelo Spognardi, editors, *ACNS 20, Part I*, volume 12146 of *LNCS*, pages 271–291. Springer, Heidelberg, October 2020.
- FFHR19. Antonio Faonio, Dario Fiore, Javier Herranz, and Carla Ràfols. Structure-preserving and re-randomizable RCCA-secure public key encryption and its applications. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 159–190. Springer, Heidelberg, December 2019.
- FLSZ17. Prastudy Fauzi, Helger Lipmaa, Janno Siim, and Michal Zajac. An efficient pairing-based shuffle argument. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, pages 97–127. Springer, Heidelberg, December 2017.
- FR22. Antonio Faonio and Luigi Russo. Mix-nets from re-randomizable and replayable cca-secure public-key encryption. In Clemente Galdi and Stanislaw Jarecki, editors, *Security and Cryptography for Networks*, pages 172–196. Springer International Publishing, 2022.
- GHK17. Romain Gay, Dennis Hofheinz, and Lisa Kohl. Kurosawa-desmedt meets tight security. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 133–160. Springer, Heidelberg, August 2017.

- GHKP18. Romain Gay, Dennis Hofheinz, Lisa Kohl, and Jiaxin Pan. More efficient (almost) tightly secure structure-preserving signatures. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 230–258. Springer, Heidelberg, April / May 2018.
- GHKW16. Romain Gay, Dennis Hofheinz, Eike Kiltz, and Hoeteck Wee. Tightly CCA-secure encryption without pairings. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 1–27. Springer, Heidelberg, May 2016.
- Gro04. Jens Groth. Rerandomizable and replayable adaptive chosen ciphertext attack secure cryptosystems. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 152–170. Springer, Heidelberg, February 2004.
- HHH⁺14. Gottfried Herold, Julia Hesse, Dennis Hofheinz, Carla Ràfols, and Andy Rupp. Polynomial spaces: A new framework for composite-to-prime-order transformations. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 261–279. Springer, Heidelberg, August 2014.
- HLLG19. Shuai Han, Shengli Liu, Lin Lyu, and Dawu Gu. Tight leakage-resilient CCA-security from quasi-adaptive hash proof system. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 417–447. Springer, Heidelberg, August 2019.
- Hof17. Dennis Hofheinz. Adaptive partitioning. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part III*, volume 10212 of *LNCS*, pages 489–518. Springer, Heidelberg, April / May 2017.
- JR13. Charanjit S. Jutla and Arnab Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 1–20. Springer, Heidelberg, December 2013.
- JR14. Charanjit S. Jutla and Arnab Roy. Switching lemma for bilinear tests and constant-size NIZK proofs for linear subspaces. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 295–312. Springer, Heidelberg, August 2014.
- KD04. Kaoru Kurosawa and Yvo Desmedt. A new paradigm of hybrid encryption scheme. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 426–442. Springer, Heidelberg, August 2004.
- KW15. Eike Kiltz and Hoeteck Wee. Quasi-adaptive NIZK for linear subspaces revisited. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 101–128. Springer, Heidelberg, April 2015.
- LJYP14. Benoit Libert, Marc Joye, Moti Yung, and Thomas Peters. Concise multi-challenge CCA-secure encryption and signatures with almost tight security. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 1–21. Springer, Heidelberg, December 2014.
- LPJY15. Benoit Libert, Thomas Peters, Marc Joye, and Moti Yung. Compactly hiding linear spans - tightly secure constant-size simulation-sound QA-NIZK proofs and applications. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 681–707. Springer, Heidelberg, November / December 2015.
- LPQ17. Benoit Libert, Thomas Peters, and Chen Qian. Structure-preserving chosen-ciphertext security with shorter verifiable ciphertexts. In Serge Fehr, editor, *PKC 2017, Part I*, volume 10174 of *LNCS*, pages 247–276. Springer, Heidelberg, March 2017.

- Mau05. Ueli M. Maurer. Abstract models of computation in cryptography (invited paper). In Nigel P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *LNCS*, pages 1–12. Springer, Heidelberg, December 2005.
- MW96. Ueli M. Maurer and Stefan Wolf. Diffie-Hellman oracles. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 268–282. Springer, Heidelberg, August 1996.
- NAP⁺14. Muhammad Naveed, Shashank Agrawal, Manoj Prabhakaran, XiaoFeng Wang, Erman Ayday, Jean-Pierre Hubaux, and Carl A. Gunter. Controlled functional encryption. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *ACM CCS 2014*, pages 1280–1291. ACM Press, November 2014.
- Nec94. V. I. Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55(2):165–172, 1994.
- PR07. Manoj Prabhakaran and Mike Rosulek. Rerandomizable RCCA encryption. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 517–534. Springer, Heidelberg, August 2007.
- PR17. Olivier Pereira and Ronald L. Rivest. Marked mix-nets. In Michael Brenner, Kurt Rohloff, Joseph Bonneau, Andrew Miller, Peter Y. A. Ryan, Vanessa Teague, Andrea Bracciali, Massimiliano Sala, Federico Pintore, and Markus Jakobsson, editors, *FC 2017 Workshops*, volume 10323 of *LNCS*, pages 353–369. Springer, Heidelberg, April 2017.
- RS92. Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 433–444. Springer, Heidelberg, August 1992.
- Sho97. Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Heidelberg, May 1997.
- Tar75. Robert Endre Tarjan. Efficiency of a good but not linear set union algorithm. In *Journal of the ACM*, 1975.
- WCY⁺21. Yi Wang, Rongmao Chen, Guomin Yang, Xinyi Huang, Baosheng Wang, and Moti Yung. Receiver-anonymity in rerandomizable RCCA-secure cryptosystems resolved. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 270–300, Virtual Event, August 2021. Springer, Heidelberg.

A Missing Parts from Section 3

A.1 More details on our NIDVPS

The scheme we propose in Section 3.1 is inspired by the work of [ABP15]. In this section we clarify which are the major differences between these two schemes. The work of [ABP15] show how to instantiate the disjunction of two SPHF's for two languages based on diverse vector spaces. We don't need such a functionality for our benign proof system, indeed our proof system is for linear space, i.e. the prover can generate proofs for elements that belong to the span of matrix \mathbf{D} . On the other hand, the security property of our benign proof system allows for soundness even in presence of simulated proofs for elements in two (possibly distinct) super-spaces of the prescribed linear space. In other words, while [ABP15] enables for SPHF's for (arbitrary) disjoint linear spaces, our goal is to enable for proof system for linear spaces with enhanced soundness properties (w.r.t. simulated proofs from disjoint super-spaces).

B Missing Parts from Section 5

B.1 Missing Steps of the proof of Rand-mRCCA

From \mathbf{G}_2 to \mathbf{G}_3 We build adversaries \mathcal{B} , \mathcal{B}' such that for a polynomial $p(\lambda)$ independent of $T(\mathcal{A})$, we have $T(\mathcal{B}) \approx T(\mathcal{B}') \approx T(\mathcal{A}) + (Q_{\text{Enc}} + Q_{\text{Dec}}) \cdot p(\lambda)$ and

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

Let $([\mathbf{D}]_1, [\mathbf{V}]_1)$ be the Q_{Enc} -fold $\mathcal{D}_{n,d}$ -MDDH challenge received by \mathcal{B} , with $[\mathbf{D}]_1 \in \mathbb{G}_1^{n \times d}$ and $[\mathbf{V}]_1 := ([\mathbf{v}_1]_1, \dots, [\mathbf{v}_{Q_{\text{Enc}}}]_1) \in \mathbb{G}_1^{n \times Q_{\text{Enc}}}$. Then, \mathcal{B} samples the secret material $\mathbf{a} \leftarrow_{\$} \mathbb{Z}_q^n$, $\mathbf{f} \leftarrow_{\$} \mathbb{Z}_q^n$, $\mathbf{F} \leftarrow_{\$} \mathbb{Z}_q^{n \times n+1}$, generates the keys for the benign proof system, running $\text{PGen}([\mathbf{D}]_1)$, and finally sends the public key to \mathcal{A} . On the j -the query to the encryption oracle, \mathcal{B} sets

$$\begin{aligned} [\mathbf{u}_j]_1 &\leftarrow [\mathbf{v}_j]_1 \\ [y_j]_T &\leftarrow \mathbf{f}^\top [\mathbf{u}_j]_T + [\mathbf{x}_j]_1^\top \mathbf{F}^\top [\mathbf{u}_j]_1 \end{aligned}$$

\mathcal{B} has generated the secret key itself, so it can perfectly simulate the decryption oracle. In case $[\mathbf{V}]_1 = [\mathbf{DR}]_1$, \mathcal{B} perfectly simulates \mathbf{G}_2 . In case $[\mathbf{V}]_1$ is uniformly random over $\mathbb{G}_1^{n \times Q_{\text{Enc}}}$, \mathcal{B} simulates an intermediary game \mathbf{H} . Analogously, we can build an adversary \mathcal{B}' on the Q_{Enc} -fold $\mathcal{D}_{n,2d}$ -MDDH assumption, who simulates \mathbf{H} if $[\mathbf{V}]_1$ is uniformly random over $\mathbb{G}_1^{n \times Q_{\text{Enc}}}$, and game \mathbf{G}_3 if $[\mathbf{V}]_1 = [\mathbf{D}_0\mathbf{R}]_1$. Altogether, this proves the claim stated above: from Lemma 1 and Corollary 1, we obtain an adversary \mathcal{B}'' such that $T(\mathcal{B}'') \approx T(\mathcal{A}) + (Q_{\text{Enc}} + Q_{\text{Dec}}) \cdot \text{poly}(\lambda)$ where poly is independent of $T(\mathcal{A})$ and

$$|\epsilon_3 - \epsilon_2| \leq 2(n-d) \mathbf{Adv}_{\mathbb{G}_1, \mathcal{D}_{n,d}, \mathcal{B}''}^{\text{mddh}}(\lambda) + \frac{2}{q-1}.$$

Game \mathbf{G}_6 We recall that in this last step we encrypt a random message. More formally: for each query, the encryption oracle encrypts the message $M_{j_{b^*}} + R_j$, where R_j is random. Clearly, it holds that $\epsilon_6 = 0$. We first observe switching from \mathbf{a}^\top to $\mathbf{a}^\top + \mathbf{r}^\top \mathbf{D}^\perp$ for a random \mathbf{r} and using the vector $\mathbf{a}^\top + \mathbf{r}^\top \mathbf{D}^\perp$ both for the encryption and decryption oracle queries does not change the view of the adversary. So consider a new game \mathbf{G}'_5 , identical to \mathbf{G}_5 but where we set $\mathbf{a}^\top + \mathbf{r}^\top \mathbf{D}^\perp$ instead of \mathbf{a} .

Let $[\mathbf{u}_j] = [\mathbf{D}_0] \mathbf{r}_j$ be the first component of the j -th challenge ciphertext, computed by the encryption oracle. It holds that $[p_j] = (\mathbf{a}^\top + \mathbf{r}^\top \mathbf{D}^\perp)[\mathbf{u}_j] + [M_{j_{b^*}}]$, which is equal to $\mathbf{a}^\top [\mathbf{u}_j] + [M_{j_{b^*}}] + \mathbf{r}^\top (\mathbf{D}^\perp [\mathbf{D}_0]) \mathbf{r}_j$. We show that the term $\mathbf{r}^\top (\mathbf{D}^\perp [\mathbf{D}_0])$, is statistically close to a random vector $[\mathbf{v}]$. The reason is that for any fixed choice of the columns of \mathbf{D}_0 , the rows of \mathbf{D}^\perp are linearly independent with overwhelming probability (over \mathbf{D}^\perp).

We now reduce to MDDH to show that $|\epsilon_6 - \epsilon_{5'}|$ is negligible in the security parameter for any adversary \mathcal{A} : more formally, we build an adversary \mathcal{B} such that $T(\mathcal{B}) \approx T(\mathcal{A}) + (Q_{\text{Enc}} + Q_{\text{Dec}}) \cdot \text{poly}(\lambda)$ and:

$$|\epsilon_6 - \epsilon_{5'}| \leq \mathbf{Adv}_{\mathbb{G}_1, \mathcal{U}_{2d+1, 2d}, \mathcal{B}}^{\text{mddh}}(\lambda).$$

Let $[\mathbf{B}] \in \mathbb{G}_1^{2d+1 \times 2d}$, and let $([\mathbf{h}_1], \dots, [\mathbf{h}_{Q_{\text{Enc}}]}) \in \mathbb{G}_1^{2d+1 \times Q_{\text{Enc}}}$ be the $\mathcal{U}_{2d+1, 2d}$ -MDDH challenge received in input. We indicate the upper values of $[\mathbf{h}_j]$ as $[\overline{\mathbf{h}}_j] \in \mathbb{G}_1^{2d}$, and we indicate its lower values as $[\underline{\mathbf{h}}_j] \in \mathbb{G}_1$. In the reduction, the encryption oracle sets $[\mathbf{u}_j] \leftarrow \mathbf{D}_0 [\overline{\mathbf{h}}_j]$, which implicitly sets $\mathbf{r}_j = \overline{\mathbf{h}}_j$. Also it sets $[p_j] = [M_{j_{b^*}}] + \mathbf{a}^\top [\mathbf{u}_j] + [\underline{\mathbf{h}}_j]$. If $([\mathbf{B}], [\mathbf{h}_1], \dots, [\mathbf{h}_Q])$ is a real MDDH challenge, \mathcal{B} simulates the game $\mathbf{G}_{5'}$. Indeed, it must hold that $\mathbf{h}_j = \mathbf{B} \mathbf{s}_j$, for some $\mathbf{s}_j \in \mathbb{Z}_q^{2d}$. Also, let $[\overline{\mathbf{B}}]$ be the upper square matrix of $[\mathbf{B}]$, and let $[\underline{\mathbf{B}}]$ be the last row of $[\mathbf{B}]$. $[\underline{\mathbf{h}}_j] = [\underline{\mathbf{B}}] \mathbf{s}_j = [\underline{\mathbf{B}}] \overline{\mathbf{B}}^{-1} \mathbf{r}_j$. And we have that the distribution of $[\underline{\mathbf{B}}] \overline{\mathbf{B}}^{-1}$ is statistically close to a random element \mathbf{v} . Otherwise, \mathcal{B} simulates the game \mathbf{G}_6 . Finally, by applying Lemma 1, Lemma 2 and Lemma 3, we can always build a new adversary \mathcal{C} such that $T(\mathcal{C}) \approx T(\mathcal{A}) + (Q_{\text{Enc}} + Q_{\text{Dec}}) \cdot \text{poly}(\lambda)$ and:

$$|\epsilon_6 - \epsilon_5| = |\epsilon_6 - \epsilon_{5'}| \leq (n - d) \mathbf{Adv}_{\mathbb{G}_1, \mathcal{D}_{n,d}, \mathcal{C}}^{\text{mddh}}(\lambda) + \frac{1}{q-1}.$$

C Missing Parts from Section 6

C.1 Quasi-Adaptive NIZK Proof of Knowledge for the Input-Submission Phase

The MixNet protocol, when instantiated with either $\mathcal{PK}\mathcal{E}_2$ or $\mathcal{PK}\mathcal{E}_3$ additionally needs a simulation-extractable NIZK proof system for the relation:

$$\mathcal{R}' = \{(\text{pk}, \text{C}), (\mathbf{r}, \text{M}) : [\mathbf{x}]_1 = [\mathbf{D}^*] \mathbf{r} + \mathbf{eM}\}$$

where \mathbf{e} is the $n + 1$ -th basis. In particular, it is sufficient the notion of simulation f -extractability, where f is the efficiently computable function that maps

$\mathbf{Exp}_{\mathcal{A}, \mathbf{PS}, f, \mathcal{R}}^{\text{sim-ext}}(\lambda)$
 \mathcal{R} sampled from the relations distribution \mathbf{R} .
 $(ppk, psk) \leftarrow \mathbf{PGen}(1^\lambda, \mathcal{R})$; Set $\mathcal{Q} \leftarrow \emptyset$;
 $(x, \pi) \leftarrow \mathcal{A}(ppk)^{\mathcal{SLM}(\cdot)}$; $z \leftarrow \mathbf{Ext}(psk, x, \pi)$;
return $x \notin \mathcal{Q} \wedge \mathbf{PVer}(ppk, x, \pi) = 1 \wedge (\nexists w : f(w) = z \wedge (x, w) \notin \mathcal{R})$

$\mathcal{SLM}(x, w)$:
 $\pi \leftarrow \mathbf{PSim}(psk, x)$;
 $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{x\}$;
return π .

Fig. 6: The security experiments for the NIZK argument system.

tuples (\mathbf{M}, \mathbf{r}) to \mathbf{M} . Moreover, we can focus on quasi-adaptive NIZK that works for distribution of relation indexed by the public key \mathbf{pk} .

We show a simulation f -extractable QA-NIZK based on the simulation-sound QA-NIZK for linear spaces of Abe *et al.*[AJOR18].

Definition 14. Let $\mathbf{PS} = (\mathbf{PGen}, \mathbf{PPrv}, \mathbf{PVer})$ be a NIZK argument for a language \mathcal{L} , we say that \mathbf{PS} is simulation sound if there exist PPT algorithms \mathbf{PGen} and \mathbf{PVer} such that the distribution of the proving key output by \mathbf{PGen} and \mathbf{PGen} are identical and the proof system $(\mathbf{PGen}, \mathbf{PPrv}, \mathbf{PVer})$ is complete and computational $(\mathcal{L}^{\text{sim}}, \mathcal{L}^{\text{ver}}, \mathcal{L}^{\text{snd}})$ -sound as per Definition 3 (namely, the proof system is benign-sound for any PPT adversary) where $\mathcal{L}^{\text{sim}} = \mathcal{L}^{\text{ver}} = \{0, 1\}^*$ and $\mathcal{L}^{\text{snd}} = \mathcal{L}$. We call $\mathbf{Adv}_{\mathcal{A}, \mathbf{PS}}^{\text{sim-sound}}(\lambda)$ the advantage of the adversary \mathcal{A} in the security experiment $\mathbf{Exp}^{\text{snd}}$ described in Fig. 1.

Definition 15. Let f be an efficiently computable function, let $\mathbf{PS} = (\mathbf{PGen}, \mathbf{PPrv}, \mathbf{PVer})$ be a NIZK argument for a relation \mathcal{R} , and consider the experiment $\mathbf{Exp}^{\text{sim-ext}}$ described in Fig. 6. We say that NIZK is simulation extractable iff there exists a PPT algorithm \mathbf{Ext} such that for all PPT \mathcal{A} we have that

$$\mathbf{Adv}_{\mathcal{A}, \mathbf{PS}}^{\text{sim-ext}}(\lambda) := \Pr \left[\mathbf{Exp}_{\mathcal{A}, \mathbf{PS}, f}^{\text{sim-ext}}(\lambda) = 1 \right] \in \text{negl}(\lambda).$$

Let $\mathbf{PS} = (\mathbf{PGen}, \mathbf{PPrv}, \mathbf{PVer})$ be a simulation-sound QA-NIZK for linear spaces. Consider the proof system $\mathbf{PS}' = (\mathbf{PGen}', \mathbf{PPrv}', \mathbf{PVer}')$ for the relation is $\mathcal{R}'_{\mathbf{pk}} = \{[\mathbf{x}], (\mathbf{M}, \mathbf{r}) : [\mathbf{x}] = [\mathbf{D}^*]\mathbf{r} + \mathbf{eM}\}$ of \mathbb{Z}_q^{n+1} and \mathbf{D}^* is sampled as described in the description of the RCCA-PKE of Section 5 and where:

- $\mathbf{PGen}'(1^\lambda)$ samples $\mathbf{U} \leftarrow \mathcal{U}_{n+1,2}$ runs $\text{crs} \leftarrow \mathbf{PGen}(1^\lambda, [\mathbf{P}]_1)$ where \mathbf{P} is the matrix $\mathbf{D}^* \parallel \mathbf{U}$ and outputs $\text{crs}' = (\text{crs}, [\mathbf{U}]_1)$.
- $\mathbf{PPrv}'([\mathbf{x}], (\mathbf{M}, \mathbf{r}))$ samples \mathbf{s} , computes $[\mathbf{c}] = [\mathbf{U}]\mathbf{s} + \mathbf{eM}$ and then it computes a proof π using \mathbf{PPrv} that $([\mathbf{c}] - [\mathbf{x}]) \in \text{span}([\mathbf{P}]_1)$ (whose witness is the vector $(\mathbf{r}^\top, \mathbf{s}^\top)^\top$) and it outputs $\pi, [\mathbf{c}]$.
- $\mathbf{PVer}'([\mathbf{x}], \pi')$ verifies that $[\mathbf{c} - \mathbf{x}] \neq [\mathbf{0}]$ and that π is a valid proof of membership in the linear space spanned by $[\mathbf{P}]_1$ for $[\mathbf{c} - \mathbf{x}]_1$.

Lemma 9. *Let f be the function that maps (M, \mathbf{r}) to M , For any adversary \mathcal{A} exists adversary \mathcal{B} such that*

$$\mathbf{Adv}_{\mathcal{A}, \mathbf{PS}', f}^{\text{sim-ext}}(\lambda) \leq \mathbf{Adv}_{\mathcal{B}, \mathbf{PS}}^{\text{sim-sound}}(\lambda) + O(1/q).$$

The extractor uses the trapdoor \mathbf{U} to extract from $[\mathbf{c}]$ the message M . Assume that the proof verifies but that the extracted message M' is not the same as the message M encrypted in \mathbf{x} , then $\mathbf{x} - \mathbf{c} \in \text{span}(\mathbf{P})$ implies that $\mathbf{e} \in \text{span}(\mathbf{P})$ which can happen with probability smaller than $O(1/q)$ over the choice of \mathbf{U} and \mathbf{D}^* .