Structure-Preserving and Re-randomizable RCCA-secure Public Key Encryption and its Applications

Antonio Faonio[1], Dario Fiore[1], Javier Herranz[2], and Carla Ràfols[3]

[1] IMDEA Software Institute,
[2] Universitat Politècnica de Catalunya,
[3] Universitat Pompeu Fabra

Abstract. Re-randomizable RCCA-secure public key encryption (Rand-RCCA PKE) schemes reconcile the property of re-randomizability of the ciphertexts with the need of security against chosen-ciphertexts attacks.

In this paper we give a new construction of a Rand-RCCA PKE scheme that is perfectly re-randomizable. Our construction is structure-preserving, can be instantiated over Type-3 pairing groups, and achieves better computation and communication efficiency than the state of the art perfectly re-randomizable schemes (e.g., Prabhakaran and Rosulek, CRYPTO'07).

Next, we revive the Rand-RCCA notion showing new applications where our Rand-RCCA PKE scheme plays a fundamental part: (1) We show how to turn our scheme into a *publicly-verifiable* Rand-RCCA scheme; (2) We construct a malleable NIZK with a (variant of) simulation soundness that allows for re-randomizability; (3) We propose a new UC-secure Verifiable Mix-Net protocol that is secure in the common reference string model. Thanks to the structure-preserving property, all these applications are efficient. Notably, our Mix-Net protocol is the most efficient universally verifiable Mix-Net (without random oracle) where the CRS is an uniformly random string of size independent of the number of senders. The property is of the essence when such protocols are used in large scale.

Table of Contents

1 Introduction

Security against chosen ciphertext attacks (CCA) is considered by many the gold standard for public key encryption (PKE). Since the seminal paper of Micali, Rackoff and Sloan [39], the research community has spent a great effort on this fundamental topic by both interconnecting different security notions and producing a large body of efficient public encryption schemes.

Challenging the overwhelming agreement that CCA security is **the** right notion of security for PKE, a paper of Canetti, Krawczyk and Nielsen [7] showed that for many use cases a weaker security notion than CCA security is already sufficient. More in details, the paper introduced the notion of Replayable CCA (RCCA) and showed that the notion is sufficient to realize a variant of the public key encryption functionality in the universal composability (UC) model of Canetti [4] where only replay attacks, namely attacks in which the data could be maliciously repeated, can be mounted by the adversary.

In a nutshell, the main fundamental difference between RCCA security and CCA security is that, in a RCCA secure scheme (which is not CCA secure) an adversary is able to maul the challenge ciphertext to obtain new decryptable ciphertexts, the only limitation is that the adversary still cannot break the integrity of the underlying plaintext. To explain this with an example, in a RCCA secure PKE scheme an adversary might append an extra 0 at the end of the ciphertext and still be able to obtain a valid decryption of the mauled ciphertext (to the same plaintext), on the other hand, for a CCA secure PKE, this attack should by definition result into an invalid decryption.

Later, Groth [27] showed that the capability to maul a ciphertext to obtain a new ciphertext which decrypts to the same plaintext should be seen as a feature and not a weakness. In his paper, he introduced the notion of re-randomizable RCCA (Rand-RCCA) PKE, namely a

RCCA-secure PKE which comes with an algorithm that re-randomizes the ciphertexts in a way that cannot be linked.

PKE schemes that are both re-randomizable and RCCA-secure have been shown to have several applications, such as: anonymous and secure message transmissions (see Prabhakaran and Rosulek [45]), Mix-Nets (see Pereira and Rivest [43]), Controlled Functional Encryption (see Naveed *et al.* [42]), and one-round message-transmission protocols with reverse firewalls (see Dodis, Mironov, and Stephens-Davidowitz [14]).

When it comes to constructing these objects, if we look at the literature it is striking to observe that there are extremely efficient constructions of schemes that are only RCCA-secure but not re-randomizable (e.g., Cramer-Shoup [11] or Phan-Pointcheval [44]), or are re-randomizable but only CPA-secure (e.g., ElGamal [15]). In contrast, when the two properties are considered in conjunction, a considerable gap in the efficiency of the schemes seems to arise. More in concrete, the most efficient Rand-RCCA scheme in the standard model of [45] has ciphertexts of 20 groups elements,[4] while, for example, the celebrated Cramer-Shoup PKE [11] has ciphertexts of only 4 groups elements.

In the following paragraphs we state the main contributions of our work.

Rand-RCCA PKE. Our first contribution is a new structure-preserving[5] Rand-RCCA PKE scheme which significantly narrows the efficiency gap described above. The scheme is secure under the Matrix Diffie-Hellman Assumption (MDDH) in bilinear groups, and for its strongest instantiation, namely, under the Symmetric External Diffie-Hellman Assumption (SXDH), has ciphertexts of 6 groups elements (3 elements in $\mathbb{G}_1$, 2 elements in $\mathbb{G}_2$ and 1 element in $\mathbb{G}_T$).

From a practical perspective, the advantage of a re-randomizable PKE over a standard (non-re-randomizable) PKE strikes when the re-randomizable PKE scheme is part of a larger protocol. To this end, we notice that the structure-preserving property is indeed vital as it allows for modularity and easy integration, which are basic principles for protocol design. However, we can substantiate further our assertion by giving three applications where structure-preserving Rand-RCCA PKE schemes are essential.

Publicly-verifiable Rand-RCCA PKE. Our first application is a publicly-verifiable (pv) Rand-RCCA PKE scheme. A PKE scheme is publicly verifiable when the validity of a ciphertext can be checked without the secret key. This property is for example convenient in the setting of threshold decryption with CCA security [47,5], as the task, roughly speaking, reduces to first publicly check the validity of the ciphertext and then CPA-threshold-decrypt it. Very roughly speaking, we can obtain our pv-Rand-RCCA PKE scheme by appending a Groth-Sahai (GS) NIZK proof [29] of the validity of the ciphertext. We notice that the ciphertext of our Rand-PKE scheme contains[6] an element in $\mathbb{G}_T$. The verification equation does not admit a GS NIZK proof, but only NIWI. We overcome this problem by constructing an additional commitment type for elements in $\mathbb{G}_T$. This gives us a *new* general technique that extends the class of pairing product equations which admit GS NIZK proofs, enlarging therefore the notion of structure preserving. The latter is a contribution of independent interest which might have applications in the field of structure-preserving cryptography in general.

Controlled-Malleable NIZKs. Our second application is a general framework for true-simulation extractable (tSE) and re-randomizable (more generally, controlled-malleable) NIZK systems. The notion of tSE-NIZK was introduced by Dodis *et al.* [13] and found a long series

[4] A recent work of Faonio and Fiore [18], originally appeared in [17] takes this down to 11 group elements at the price of achieving a strictly weaker notion of re-randomizability, in the random oracle model.

[5] A scheme is structure preserving if all its public materials, such as messages, public keys, etc. are group elements and the correctness can be verified via pairing-product equations.

[6] In the lingo of structure-preserving cryptography, the scheme is not *strongly* structure preserving.

of applications (see for example [22,12,20]). Briefly, the notion assures soundness of the NIZK proofs even when the adversary gets to see simulated NIZK proofs for *true* statements of its choice. In comparison with simulation-extractable (SE) NIZKs (see [46,28]), tSE-NIZKs are considerably more efficient and keep many of the benefits which motivated the introduction of SE-NIZKs[7]. However, if one would like a *controlled malleable tSE-NIZK*, the only available scheme is an SE-NIZK obtained through the general result of Chase *et al.* [8], which is not very efficient. As main result, we scale down the framework of Chase *et al.* to true-simulation extractability, and by using our new Rand-RCCA PKE we construct a new re-randomizable tSE-NIZK scheme. Compared to [8], our scheme can handle a more restricted class of relations and transformations,[8] but our proofs are significantly more efficient. For example, for simple re-randomizable NIZK proofs our tSE NIZKs have an overhead of the order of *tens* more pairing operations for verification, opposed to an overhead of the order of *hundreds* more pairing operations for verification of the simulation-extractable with controlled malleability NIZK systems of [8]. The overhead is computed as the difference with the adaptive sound Groth-Sahai NIZK proof for the same statement.

Mix-Net. Our third application is a universally verifiable and UC-secure Mix-Net based on our pv-Rand-RCCA PKE scheme. Recently, Faonio and Fiore [18] gave a new paradigm to obtain UC-secure verifiable Mix-Net protocols based on Rand-RCCA PKE scheme. Their construction makes use of a non-publicly verifiable Rand-RCCA PKE scheme and obtains a weaker notion of security called *optimistic* (*àla* Golle *et al.* [26]). More in details, the mixing paradigm of [18] is conceptually simple: a mixer receives a list of Rand-RCCA ciphertexts and outputs a randomly permuted list of re-randomized ciphertexts together with a simple NIZK proof that they informally dub "loose shuffling". Such "loose shuffling" proof guarantees that if all the ciphertexts correctly decrypt then the output list is a shuffle of the input one. Hence, in their scheme, cheating can be caught at decryption time, that is after the last mixer returned its list. The problem is that, cheating might be caught too late, thus, their scheme is only optimistic secure. Namely, the scheme is an universal verifiable mix-net optimized to quickly produce a correct output when all the mixers run the protocol correctly. If instead one or more mixers cheat, then no privacy is guaranteed but one can "back up" to a different, slow, mix-net execution.

In this paper, we show that by leveraging the public verifiability of the Rand-RCCA PKE scheme we can obtain a simple design for Mix-Net protocols. In fact, since it is possible to publicly check that a mixer did not invalidate any ciphertext, the proof of loose shuffling turns out to be, indeed, a proof of shuffle.

Interestingly, our use of publicly verifiable ciphertexts come with additional benefits. As mentioned in the paragraph above, our pv-RCCA-PKE scheme can support threshold decryption very easily, and more efficiently than Faonio and Fiore [18]. Finally, our protocol can be fully instantiated in the standard model, whereas the one in [18] rely on non-programmable random oracles.

Most notably, our protocol is the *first efficient universally verifiable Mix-Net in the common random string model*, namely where the common reference string is a (small) uniformly random string. In fact, a popular approach to achieve a universally verifiable Mix-Net is to use a NIZK proof of shuffle. However, the most efficient protocols for this task either rely on random oracles

[7] As an example, tSE-NIZKs are sufficient for the CCA2-secure Naor-Yung PKE of Sahai [46], simulation-sound (SS) NIZKs were introduced in the same paper with exactly this application in mind.

[8] Yet, our framework is powerful enough for the application of controlled-malleable CCA security of Chase *et al.* Interestingly, we can obtain another pv-Rand-RCCA PKE through their paradigm, although less efficient than our construction. We believe that analyzing what other kinds of CM-CCA notions are supported by our scheme is interesting future work.

PKE	Group Setting	Assumption	Model	Struc. Pres.	Pub. Ver.	Re-Rand
[27] Groth	–	DDH	GGM			perfect
[45] PR07	Cunn.	DDH	std			perfect
[8,37] CKLM12, LPQ17	Bilin.	SXDH	std	✓	✓	perfect
[18] FF18	–	DDH	NPRO			weak
$\mathcal{PKE}_1$	Bilin.	$\mathcal{D}_k$-MDDH	std	✓*		perfect
$\mathcal{PKE}_2$	Bilin.	$\mathcal{D}_k$-MDDH	std	✓*	✓	perfect

Table 1. Comparison of the properties of a selection of Rand-RCCA-secure PKE schemes. For group setting, – means any group where the assumption holds; Cunn. refers to a pair of groups whose prime orders form a Cunningham chain (see [45]); Bil. stands for bilinear groups. For model, GGM refers to generic group and NPRO refers to non-programmable random oracle. * the structure-preserving property of the two schemes in this paper is not strict, since ciphertexts contain some elements in $\mathbb{G}_T$.

to become non-interactive (such as the protocol of Bayer and Groth [1] or Verificatum [50]), or need a structured common reference string (as is the case for the most efficient state-of-the-art NIZK proof of shuffle of Fauzi *et al.* [21]). Furthermore, the common reference string of [21] has size that depends on the number of senders (which in practical scenarios can be huge), whereas our common reference string is made by a number of group elements that is linear in the number of mixers.

Our Mix-Net protocol is proved secure based only on general properties of the pv-Rand-RCCA PKE scheme, and can be instantiated with other schemes in literature (for example with the schemes in [37,8]).

Controlled-Malleable Smooth Projective Hash Functions. At the core of our Rand-RCCA PKE scheme is a new technique that can be seen as a re-randomizable version of smooth projective hash functions (SPHFs) [11]. Given the pervasive use of SPHFs in cryptographic constructions, we believe that our technique may find more applications in the realm of re-randomizable cryptographic primitives. For this reason, we formalize our technique as a primitive called *controlled-malleable SPHF*. Briefly, we define it as an SPHF with tags that allows to re-randomize both instances and tags (inside appropriate spaces), and for which soundness (i.e., smoothness) holds even if the adversary can see a hash value for an invalid instance. We elaborate on this notion in Appendix D.

Comparison with Related Work. If we consider the state of the art of Rand-RCCA PKE schemes, the most relevant works are the work of Groth, which introduced the notion of Rand-RCCA PKE scheme [27], the aforementioned scheme of Prabhakaran and Rosulek [45], the Rand-RCCA PKE scheme of Chase *et al.* derived from their malleable NIZK systems [8], and two recent works of Libert, Peters and Qian [37] and of Faonio and Fiore [18]. In Table 1 we offer a comparison, in terms of security and functionality properties, of our schemes of Sec. 3 ($\mathcal{PKE}_1$) and Sec. 4 ($\mathcal{PKE}_2$) against previous schemes.

From a technical point of view, the scheme of [45] and our scheme $\mathcal{PKE}_1$, although both based on the Cramer-Shoup paradigm, have little in common. The main differences are: (1) a different design to handle the tags (see next section); (2) a different approach for the re-randomization of the ciphertext. In particular, the Rand-PKE scheme of [45] uses the double-strand technique of Golle *et al.* [25] to re-randomize the ciphertext, while our re-randomization technique, as far as we know, is novel. Furthermore, the scheme of [45] works in two special groups, $\hat{\mathbb{G}}$ and $\tilde{\mathbb{G}}$ that are the subgroups of quadratic residues of $\mathbb{Z}^*_{2q+1}$ and $\mathbb{Z}^*_{4q+3}$ respectively, for a prime q such that $(q, 2q+1, 4q+3)$ is a sequence of primes (a Cunningham Chain of the first kind of length 3).

PKE	Enc ≈ Rand	Dec	|C|	|pk|
PR07	22 $\tilde{E}$	32 $\tilde{E}$	20$\tilde{\mathbb{G}}$	11$\tilde{\mathbb{G}}$
FF18	16 E	18 E	11$\mathbb{G}$	11$\mathbb{G}$
$\mathcal{PKE}_1$	$4E_1$+$5E_2$+$2E_T$+5P	$8E_1$+$4E_2$+4P	3$\mathbb{G}_1$+2$\mathbb{G}_2$+$\mathbb{G}_T$	7$\mathbb{G}_1$+7$\mathbb{G}_2$+2$\mathbb{G}_T$
LPQ17	$79E_1$+$64E_2$	$1E_1$+142P	42$\mathbb{G}_1$+20$\mathbb{G}_2$	11$\mathbb{G}_1$+16$\mathbb{G}_2$
$\mathcal{PKE}_2$	$36E_1$+$45E_2$+$6E_T$+5P	$2E_1$+50P	14$\mathbb{G}_1$+15$\mathbb{G}_2$+4$\mathbb{G}_T$	8$\mathbb{G}_1$+8$\mathbb{G}_2$

Table 2. Efficiency comparison among the best Rand-RCCA-secure PKE schemes; only the last two rows include schemes with public verifiability. For our schemes we consider $k = 1$, so based on SXDH assumption. We use $\tilde{\mathbb{G}}$ for the special groups used in [45], $\mathbb{G}$ for standard DDH groups as considered in [**?**], and then groups in asymmetric bilinear pairings $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ as considered both in [37] and in this work. Similarly, we denote as $E, \tilde{E}, E_1, E_2, E_T$ the cost of an exponentiation in groups $\mathbb{G}, \tilde{\mathbb{G}}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$, respectively. Finally, P denotes the cost of computing a bilinear pairing.

In Table 2 we compare the efficiency of our new schemes (in the most efficient instantiation with $k = 1$) with the most efficient ones among the Rand-RCCA schemes: the ones in [45] and [18] for the case of secret verifiability, and the scheme in [37] for publicly verifiable Rand-RCCA encryption. We do not consider the scheme of Groth [27], that suffers having ciphertexts with as many group elements as the bitlength of the plaintext, and the one of [8] that is superseded by [37]. First we stress that the generic $\mathcal{D}_k$-MDDH Assumption, in the case $k = 1$, can be instantiated with the SXDH Assumption. Therefore, the security guarantees of our schemes $\mathcal{PKE}_1$ and $\mathcal{PKE}_2$ in Table 2 are exactly the same as in the pairing-based schemes in [8,37]. Among the schemes with private verifiability, the most efficient one is that in [18], but its re-randomizability property is weak and the security is in the random oracle model. Among the other two, our scheme $\mathcal{PKE}_1$ is more efficient than that in [45], because the special groups $\tilde{\mathbb{G}}$ required in [45] are large, at least 3072 bits for a security level of 128 bits. Turning to comparing with publicly verifiable schemes, the computational costs for the scheme in [37], in the table, are roughly approximate, because not all the exact computations in the algorithms of the scheme (involving Groth-Sahai proofs) are explicitly described. The size of the ciphertexts reported in [37] is $34|\mathbb{G}_1| + 18|\mathbb{G}_2|$. After personal communication with the authors, we realized that this number is not correct; the correct one is $42|\mathbb{G}_1| + 20|\mathbb{G}_2|$. Our scheme $\mathcal{PKE}_2$ is the most efficient Rand-RCCA scheme with public verifiability up to date: ciphertext size is comparable to that in [37] whereas the computational costs are significantly lower. Even for ciphertext size, ours is comparable to [37] only due to the size of the 4 $\mathbb{G}_T$ elements in our scheme. Besides that, our ciphertexts have many fewer group elements, which is conceptually simpler and, we believe, leaves hope for further improvements. For the two publicly verifiable schemes, the number of pairings required for decryption can be decreased, at the cost of increasing the number of exponentiations, by applying the batching techniques in [31]. The resulting number would be 22P for $\mathcal{PKE}_2$ and something between 40P and 50 P for the scheme in [37].

Technical Overview. We recall that the main technical contributions of this paper are: (1) a new technique for Rand-RCCA PKE scheme (which we also formalize in terms of SPHFs), (2) a new general technique that extends significantly the class of pairing product equations which admits GS NIZK proofs, and (3) a new technique for standard-model UC-secure verifiable Mix-Nets. For space reason, in this technical overview we concentrate on (1).

A common technique of many CCA-secure PKE schemes in the standard model consists in explicitly labeling each ciphertext produced by the encryption algorithm with a unique tag. This technique is usually useful to simulate the decryption oracle and to asses that the decryption oracle would not reveal any information about the challenge ciphertext. Some notable examples

of CCA-secure PKE schemes that use tags are the Cramer-Shoup PKE [11], the tag-based PKE of Kiltz [35], and IBE-to-CCA transform of Canetti, Halevi and Katz [6].

Unfortunately, unique tags are not a viable option when designing a re-randomizable PKE scheme. In fact, a ciphertext and its re-randomization would share the same tag, and so they could be trivially linked by an attacker. The main consequence is that many well-known techniques in CCA security cannot be easily exported in the context of Rand-RCCA security. A remarkable exception is the work on Rand-RCCA PKE of Prabhakaran and Rosulek [45]. In this work, the authors managed to reconcile tags and re-randomizability with an ingenious technique: the tag for a new ciphertext is computed as a re-randomizable encoding of the plaintext itself, the tag is then encrypted and attached to the rest of the ciphertext. The decryptor first decrypts the tag and then uses it to check the validity of the payload ciphertext. More in details, the PKE scheme follows the Cramer-Shoup paradigm, therefore their tag (more accurately, a part of their tag) is a $\mathbb{Z}_q$ element (for a properly chosen q). Unfortunately, the restriction on the type of the tags implies that the scheme can be instantiated only in special groups $\mathbb{G}$ of prime order q where the DDH assumption simultaneously holds for both $\mathbb{Z}_q$ and $\mathbb{G}$. Conclusively, the main drawback is a quite large ciphertext size.

We use bilinear-pairing cryptography to overcome the problem of the tags in $\mathbb{Z}_q$. Our starting point is the structure-preserving CCA-PKE of Camenisch *et al.* [3]. Briefly, their PKE scheme is based on the Cramer-Shoup paradigm, with the main twist of performing the validity check in $\mathbb{G}_T$. This trick allows to move the tags from $\mathbb{Z}_q$ to the source group. We give a brief description of the ideas underlying our PKE scheme. We use the implicit notation of Escala *et al.* [16], that uses additive notation for groups and where elements in $\mathbb{G}_i$, are denoted as $[a]_i := a\mathcal{P}_i$ where $\mathcal{P}_i$ is a generator for $\mathbb{G}_i$. The PKE scheme of [3] uses Type-1 pairing groups (where $\mathbb{G}_1 = \mathbb{G}_2$) which are less efficient and secure than Type-3 pairing groups (where no efficient isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$ is known to exist). As a first step, we convert their scheme to Type-3 pairing groups; however, for simplicity, in this overview we present the Type-1 version.

Following the blue print of Cramer and Shoup, a ciphertext of the PKE scheme of Camenisch *et al.* consists of three elements: a vector $[\mathbf{c}]_1 \in \mathbb{G}_1^3$ which we call the *instance* (for the DLIN problem described by a matrix $[\mathbf{D}]_1 \in \mathbb{G}_1^{3\times 2}$), an element $[p]_1$ which we call the *payload*, and an element $[\pi]_T$ which we call the *hash*. Together, the instance and the payload form the *tag*, that we denote as $[\mathbf{x}]_1 = [(\mathbf{c}^\top, p)^\top]_1$. The hash is, briefly speaking, a tag-based designated-verifier zero-knowledge proof of the randomness of $[\mathbf{c}]_1$ (namely, that $[\mathbf{c}]_1 = [\mathbf{D}]_1 \cdot \mathbf{r}$). The main difference is that in Cramer-Shoup PKE the tag is computed as a collision-resistant hash of $[\mathbf{x}]_1$, while in our scheme the is the value $[\mathbf{x}]_1$ itself. More in details, the public key material consists of $[\mathbf{D}^*]_1 = [(\mathbf{D}^\top, (\mathbf{a}^\top\mathbf{D})^\top)^\top]_1$, $[\mathbf{f}^\top\mathbf{D}]_T$, and $[\mathbf{F}^\top\mathbf{D}]_1$, where $\mathbf{a}, \mathbf{f} \in \mathbb{Z}_q^3$ and $\mathbf{F} \in \mathbb{Z}_q^{3\times 4}$ are uniformly random, and the encryption algorithm on message $[m]_1$ computes the tag as $[\mathbf{x}]_1 = [\mathbf{D}^*]_1 \cdot \mathbf{r} + [(\mathbf{0}^\top, m)^\top]_1$, and the proof of consistency as $([\mathbf{f}^\top\mathbf{D}]_T + [(\mathbf{F}^\top\mathbf{D})^\top \cdot \mathbf{x}]_T) \cdot \mathbf{r}$, where the addend $[(\mathbf{F}^\top\mathbf{D})^\top \cdot \mathbf{x}]_T$ can be efficiently computed using the pairing. Using the terminology of SPHFs, the hash of the instance $[\mathbf{c}]_1$ and tag $[\mathbf{x}]_1$ is produced using the projective hash algorithm which takes as input the witness $\mathbf{r}$ for $[\mathbf{c}]_1 \in \mathit{span}([\mathbf{D}])$, the tag $[\mathbf{x}]_1$ and the projection key $([\mathbf{f}^\top\mathbf{D}]_T, [\mathbf{F}^\top\mathbf{D}]_1)$. The decryption procedure can re-compute the hash as $e(\mathbf{f}^\top[\mathbf{c}]_1, [1]_1) + e([\mathbf{x}]_1, \mathbf{F}^\top[\mathbf{c}]_1)$, without the knowledge of the witness $\mathbf{r}$ but only using the hash key $(\mathbf{f}, \mathbf{F})$.

To validly re-randomize a ciphertext, the goal would be to compute, using only public information, a new ciphertext where the tag is of the form $[\mathbf{x}'] = [\mathbf{D}^*](\mathbf{r} + \hat{\mathbf{r}}) + [(\mathbf{0}^\top, m)^\top]_1$ (and therefore the instance is of the form $[\mathbf{c}'] = [\mathbf{D}](\mathbf{r} + \hat{\mathbf{r}})$) and the hash is of the form $([\mathbf{f}^\top\mathbf{D}]_T + [(\mathbf{F}^\top\mathbf{D})^\top\mathbf{x}']_T)(\mathbf{r} + \hat{\mathbf{r}})$. However, computing such a re-randomization of the hash is actually infeasible since the scheme is CCA secure.

To overcome this problem, our idea is to reveal enough information about the secret key so as to allow re-randomizability while keeping the scheme secure. To this end, our first observation

is to rewrite the equation defining the re-randomized hash considering what we know about $\mathbf{x}'$. Specifically, we use the fact that $(\mathbf{F}^\top\mathbf{D})^\top\mathbf{x}' = (\mathbf{F}^\top\mathbf{D})^\top(\mathbf{x}+\mathbf{D}^*\hat{\mathbf{r}}) = (\mathbf{F}^\top\mathbf{D})^\top\mathbf{x} + (\mathbf{F}^\top\mathbf{D})^\top\ \mathbf{D}^*\hat{\mathbf{r}}$. So the re-randomized hash can be decomposed in three addends as:

$$[\mathbf{f}^\top\mathbf{D} + (\mathbf{F}^\top\mathbf{D})^\top\mathbf{x}]_T(\mathbf{r}+\hat{\mathbf{r}}) \ + \ [(\mathbf{F}^\top\mathbf{D})^\top(\mathbf{D}^*\hat{\mathbf{r}})]_T\hat{\mathbf{r}} \ + \ [(\mathbf{F}^\top\mathbf{D})^\top(\mathbf{D}^*\hat{\mathbf{r}})]_T\mathbf{r}$$

Notice that the first and the second addends can be easily computed knowing the randomizer $\hat{\mathbf{r}}$, the hash $[\pi]_T$ and thanks to the pairing function. So only the third addend is missing.

The second key observation is that we can include the value $[\mathbf{F}\mathbf{D}^*]_1$ in the public key. It is easy to check that, due to the bilinearity of the pairing function, we can compute the missing part as a function of tag $\mathbf{x}$, the randomizer $\hat{\mathbf{r}}$ and this extra piece of information. The third addend can be rewritten as:

$$[(\mathbf{F}^\top\mathbf{D})^\top(\mathbf{D}^*\hat{\mathbf{r}})]_T\mathbf{r} = [\mathbf{D}^\top\mathbf{F}\mathbf{D}^*\hat{\mathbf{r}}]_T\mathbf{r} = [(\mathbf{r}^\top\mathbf{D}^\top)(\mathbf{F}\mathbf{D}^*)\hat{\mathbf{r}}]_T = [\mathbf{x}^\top(\mathbf{F}\mathbf{D}^*\hat{\mathbf{r}})]_T$$

(The last equation can be computed using the pairing $e\left([\mathbf{x}]_1,\ [\mathbf{F}\mathbf{D}^*]\hat{\mathbf{r}}\right)$.) However, at first look, it is not clear why the scheme should still be secure. To understand it, let us strip away all the computational pieces of the scheme, keeping only the information-theoretic core. In a nutshell, the (one-time simulation) soundness property of the hash boils down to the fact that the function $f(\mathbf{x}) = \mathbf{f} + \mathbf{F}\cdot\mathbf{x}$ is pair-wise independent, meaning that, with knowledge of $f(\mathbf{x})$ one cannot predict $f(\mathbf{x}')$ for $\mathbf{x} \neq \mathbf{x}'$ better than guessing it. However, once we publish the value $\mathbf{F}\mathbf{D}^*$ we lose this property. Indeed, given $f(\mathbf{x})$ and $\mathbf{F}\mathbf{D}^*$, now we can easily compute the function f over all the points in the affine space $\{\mathbf{x}' \mid \mathbf{x}' = \mathbf{x} + \mathbf{D}^*\mathbf{r}, \mathbf{r} \in \mathbb{Z}_q^2\}$. On one hand, this is good as it allows us to re-randomize. On the other hand, we should prove that one cannot do more than this honest manipulation. Our main technical lemma shows that for any $\mathbf{x}'$ outside this affine space we still have pair-wise independence, i.e., the value $f(\mathbf{x}')$ is unpredictable.

In Appendix D, we abstract the re-randomization procedure in the framework of malleable SPHF [9] with tags. Specifically, we define three re-randomization algorithms: the first re-randomizes the instance $[\mathbf{c}]$, computing $[\mathbf{c}]+[\mathbf{D}]\cdot\hat{\mathbf{r}}$; the second re-randomizes the tag computing $[\mathbf{x}] + [\mathbf{D}^*]\cdot\hat{\mathbf{r}}$ (notice that, in the specific case of our Type-1 pairing construction, the instance and the tag "overlap", but in general, they could be unrelated); the third, taking as input the randomness used by the first two and the projection key, can re-randomize the hash value to be correct with respect to the re-randomized instance and tag.

2 Preliminaries and Definitions

A function is negligible in λ if it vanishes faster than the inverse of any polynomial in λ, we write $f(\lambda) \in \mathtt{negl}(\lambda)$ when f is negligible in λ. An asymmetric bilinear group is a tuple $\mathcal{G}$ is a tuple $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \mathcal{P}_1, \mathcal{P}_2)$, where $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are groups of prime order q, the elements $\mathcal{P}_1, \mathcal{P}_2$ are generators of $\mathbb{G}_1, \mathbb{G}_2$ respectively, $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently computable, non-degenerate bilinear map, and there is no efficiently computable isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$. Let GGen be some probabilistic polynomial time algorithm which on input 1^λ, where λ is the security parameter returns a description of an asymmetric bilinear group $\mathcal{G}$. Elements in $\mathbb{G}_i$, are denoted in implicit notation as $[a]_i := a\mathcal{P}_i$, where $i \in \{1, 2, T\}$ and $\mathcal{P}_T := e(\mathcal{P}_1, \mathcal{P}_2)$. 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 $[a]_1 \cdot [b]_2 = [a \cdot b]_T$, namely the execution of a pairing between $[a]_1$ and $[b]_2$. Vectors and matrices are denoted in boldface. We extend the pairing operation to vectors and matrices as $e([\mathbf{A}]_1, [\mathbf{B}]_2) = [\mathbf{A}^\top \cdot \mathbf{B}]_T$. $span(\mathbf{A})$ denotes the

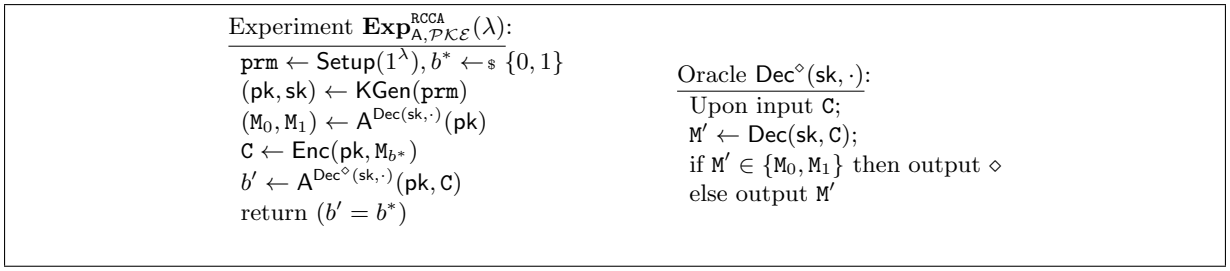

Fig. 1: The RCCA Security Experiment.

linear span of the columns of $\mathbf{A}$. Given a set of vectors V in some vector space over $\mathbb{Z}_q$, $span(V)$ denotes its linear span.

Let ℓ, k be positive integers. We call $\mathcal{D}_{\ell,k}$ a matrix distribution if it outputs (in PPT time, with overwhelming probability) matrices in $\mathbb{Z}_q^{\ell\times k}$. We define $\mathcal{D}_k := \mathcal{D}_{k+1,k}$. Our results will be proven secure under the following decisional assumption in $\mathbb{G}_\gamma$, for some $\gamma \in \{1,2\}$.

Definition 1 (Matrix Decisional Diffie-Hellman Assumption in $\mathbb{G}_\gamma$, [16]). *The $\mathcal{D}_{\ell,k}$-MDDH assumption holds if for all non-uniform PPT adversaries A,*

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

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

2.1 Re-randomizable RCCA PKE

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

$\mathsf{Setup}(1^\lambda)$ upon input the security parameter λ produces public parameters $\mathtt{prm}$, which include the description of the message and ciphertext space $\mathcal{M}, \mathcal{C}$.
$\mathsf{KGen}(\mathtt{prm})$ upon input the parameters $\mathtt{prm}$, outputs a key pair $(\mathsf{pk},\mathsf{sk})$.
$\mathsf{Enc}(\mathsf{pk},\mathtt{M})$ upon inputs a public key pk and a message $\mathtt{M} \in \mathcal{M}$, outputs a ciphertext $\mathtt{C} \in \mathcal{C}$.
$\mathsf{Dec}(\mathsf{pk},\mathsf{sk},\mathtt{C})$ upon input the secret key sk and a ciphertext $\mathtt{C}$, outputs a message $\mathtt{M} \in \mathcal{M}$ or an error symbol $\perp$.
$\mathsf{Rand}(\mathsf{pk},\mathtt{C})$ upon inputs a public key pk and a ciphertext $\mathtt{C}$, outputs another ciphertext $\mathtt{C}'$.

The RCCA security notion is formalized with a security experiment similar to the CCA security one except that in RCCA the decryption oracle (called the guarded decryption oracle) can be queried on any ciphertext and, when decryption leads to one of the challenge messages $\mathtt{M}_0, \mathtt{M}_1$, it answers with a special symbol $\diamond$ (meaning "same").

Definition 2 (Replayable CCA Security, [7]). *Consider the experiment $\mathbf{Exp}^{\mathtt{RCCA}}$ in Fig. 1, with parameters λ, an adversary A, and a PKE scheme $\mathcal{PKE}$. We say that $\mathcal{PKE}$ is* indistinguishable secure under replayable chosen-ciphertext attacks *(RCCA-secure) for any PPT adversary A:*

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

Definition 3 (Perfect Re-randomizability). *We say that $\mathcal{PKE}$ is* perfectly re-randomizable *(Re-Rand, for short) if the following three conditions are met:*

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

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

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

$$\mathsf{Dec}(\mathsf{sk}, \mathtt{C}') = \mathsf{Dec}(\mathsf{sk}, \mathtt{C}).$$

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

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

The last condition, coupled with the first one, implies that for any (possibly malicious) ciphertext that decrypts correctly the distribution of the re-randomized ciphertext and a fresh ciphertext are statistically close. This stronger property is particularly useful in applications where we need to re-randomize adversarially chosen ciphertexts.

The condition 3 is similar to the notion of *"tidiness"* of Namprempre, Rogaway and Shrimpton [41]. There are some minor differences related to the fact that we model public key (probabilistic) encryption and we leave the possibility that there may exist a negligible fraction of non tight ciphertexts. The last condition, coupled with the first one, implies that for any (possibly malicious) ciphertext that decrypts correctly the distribution of the re-randomized ciphertext and a fresh ciphertext are statistically close. This stronger property is particularly useful in applications, like our Mix-Net of Sec. 6, where we need to re-randomize adversarially chosen ciphertexts.

Definition 4 (Public Verifiability). $\mathcal{PKE} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Rand})$ *is a public key scheme with publicly verifiable ciphertexts if there is a deterministic algorithm* Ver *which, on input* $(\mathsf{pk}, \mathtt{C})$ *outputs an error symbol* $\perp$ *whenever* $\mathsf{Dec}(\mathsf{pk}, \mathsf{sk}, \mathtt{C}) = \perp$, *else it outputs valid.*

2.2 Malleable NIZKs

Recall that a non-interactive zero-knowledge proof system (NIZK) is a tuple $(\mathsf{Init}, \mathsf{P}, \mathsf{V})$ of PPT algorithms. Briefly, the algorithm Init upon input group parameters outputs a common reference string and, possibly, trapdoor information (we will consider algorithms that outputs a trapdoor tp_e for extraction and a trapdoor tp_s for simulation). We use the definitional framework of Chase *et al.* [8] for malleable proof systems. For simplicity of the exposition we consider only the unary case for transformations (see the aforementioned paper for more details). Let $T = (T_x, T_r)$ be a pair of efficiently computable functions, that we refer as a *transformation.*

Definition 5 (Admissible transformations, [8]). *An efficient relation* $\mathcal{R}$ *is* closed *under a transformation* $T = (T_x, T_w)$ *if for any* $(x, w) \in \mathcal{R}$ *the pair* $(T_x(x), T_w(w)) \in \mathcal{R}$. *If* $\mathcal{R}$ *is closed under* T *then we say that* T *is an* 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* allowable *set of transformations.*

We are ready to define malleable NIZK proof systems.

Definition 6 (Malleable NIZK, [8]). *Let* $\mathcal{NIZK} = (\mathsf{Init}, \mathsf{P}, \mathsf{V})$ *be a NIZK for a relation* $\mathcal{R}$. *Let* $\mathcal{T}$ *be an allowable set of transformations for* $\mathcal{R}$. *The proof system is* malleable with respect to $\mathcal{T}$ *if there exists an PPT algorithm* ZKEval *that on input* $(\mathsf{crs}, T, (x, \pi))$, *where* $T \in \mathcal{T}$ *and* $\mathsf{V}(\mathsf{crs}, x, \pi) = 1$ *outputs a valid proof* π' *for the statement* $x' = T_x(x)$.

We would like the property that two NIZK proofs where one is derived from the other cannot be linked. This is formalized with the notion of *derivation privacy.*

$\underline{\mathbf{Exp}^{\mathtt{der\text{-}priv}}_{\mathsf{A},\mathcal{NIZK}}}$:
$\mathtt{prm}_G \leftarrow\!\!\$\ \mathsf{Setup}_G(1^\lambda)$; $b^* \leftarrow\!\!\$\ \{0,1\}$;
$(\mathsf{crs}, tp_e, tp_s) \leftarrow \mathsf{Init}(\mathtt{prm}_G)$;
$(x, w, \pi, T) \leftarrow \mathsf{A}(\mathsf{crs}, tp_s)$; Assert $\mathsf{V}(\mathsf{crs}, x, \pi) = 1$;
If $b^* = 0$ then $\pi' \leftarrow\!\!\$\ \mathsf{P}(\mathsf{crs}, T_x(x), T_w(w))$;
else $\pi' \leftarrow\!\!\$\ \mathsf{ZKEval}(\mathsf{crs}, \pi, T)$;
$b \leftarrow \mathsf{A}(\pi')$;
Output $b = b^*$.

Fig. 2: The security experiments for the derivation privacy.

Definition 7. *Let $\mathcal{NIZK} = (\mathsf{Init}, \mathsf{P}, \mathsf{V}, \mathsf{ZKEval})$ be a malleable NIZK argument for a relation $\mathcal{R}$ and an allowable set of transformations $\mathcal{T}$. We say that $\mathcal{NIZK}$ is* derivation private *if for any PPT adversary A we have that*

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

where $\mathbf{Exp}^{\mathtt{der\text{-}priv}}$ is the game described in Fig. 2. Moreover we say that $\mathcal{NIZK}$ is perfectly derivation private *(resp.* statistically derivation private*) when for any (possibly unbounded) adversary the advantage above is 0 (resp. negligible).*

Finally, we assume that an adversary cannot find a verifying proof for a valid statement which is not in the support of the proof generated by the proving algorithm (see Def. 19 in Appendix G). We notice that this property is true for both GS proof systems and for quasi-adaptive proof system of Kiltz and Wee [36]. 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. On the other hand, the proofs of Kiltz and Wee are unique, therefore the condition is trivially true.

3 Our Rand-RCCA PKE scheme

We present our scheme in Fig. 3. We refer to the introduction for an informal exposition of our techniques. We notice that the check in the decryption procedure can be efficiently computed using the pairing function and the knowledge of $\mathbf{f}, \mathbf{F}, \mathbf{g}, \mathbf{G}$. In the next paragraphs we first show correctness of the scheme, secondly, we give an information-theoretic lemma which is the basic core of the security of our PKE scheme, then we proceed with perfect re-randomizability and the RCCA-security of the scheme.

Correctness of decryption. For correctness of decryption, it is easy to see that for a honestly generated ciphertext $([\mathbf{x}]_1, [\mathbf{v}]_2, [\pi]_T) \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{pk}, [\mathtt{M}]_1)$, the first line of decryption $[p]_1 - [\mathbf{a}^\top \mathbf{u}]_1$ yields $[\mathtt{M}]_1$. Hence, we are left with showing that the test $[\pi]_T = [(\mathbf{f} + \mathbf{F}\mathbf{v})^\top \mathbf{u}]_T + [(\mathbf{g} + \mathbf{G}\mathbf{x})^\top \mathbf{v}]_T$ is satisfied:

$$\begin{aligned}\pi = \pi_1 + \pi_2 &= (\mathbf{f}^\top \mathbf{D})\mathbf{r} + (\mathbf{F}^\top \mathbf{D}\mathbf{r})^\top \mathbf{v} + (\mathbf{g}^\top \mathbf{E})\mathbf{s} + \mathbf{x}^\top (\mathbf{G}^\top \mathbf{E})\mathbf{s} \\ &= \mathbf{f}^\top \mathbf{u} + \mathbf{u}^\top \mathbf{F}\mathbf{v} + \mathbf{g}^\top \mathbf{v} + \mathbf{x}^\top \mathbf{G}^\top \mathbf{v} \\ &= (\mathbf{f} + \mathbf{F}\mathbf{v})^\top \mathbf{u} + (\mathbf{g} + \mathbf{G}\mathbf{x})^\top \mathbf{v} \end{aligned} \tag{1}$$

Before analyzing the perfect re-randomizability and RCCA security of the scheme we state and prove a powerful information-theoretic lemma. Very informally speaking, the lemma proves that the smooth projective hash proof system at the core of our scheme remains sound even if the adversary gets to see a proof for an instance of its choice. As we want to allow for re-randomization, we relax the notion of soundness by requiring that the instance forged by the adversary does not lie in the set of possible re-randomizations of its query.

$\underline{\mathsf{Setup}(1^\lambda)}$:
$\mathcal{G} \leftarrow_\$ \mathsf{GGen}(1^\lambda)$ where
$\mathcal{G} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \mathcal{P}_1, \mathcal{P}_2)$;
$\mathcal{M} = \mathbb{G}_1$;
$\mathcal{C} = \mathbb{G}_1^{k+2} \times \mathbb{G}_2^{k+1} \times \mathbb{G}_T$;
Output $\mathtt{prm} = (\mathcal{G}, \mathcal{M}, \mathcal{C})$.

$\underline{\mathsf{KGen}(\mathtt{prm})}$:
Sample $\mathbf{D}, \mathbf{E} \leftarrow_\$ \mathcal{D}_k$;
Sample $\mathbf{a}, \mathbf{f}, \mathbf{g} \leftarrow_\$ \mathbb{Z}_q^{k+1}$;
$\mathbf{F} \leftarrow_\$ \mathbb{Z}_q^{k+1\times k+1}$ and $\mathbf{G} \leftarrow_\$ \mathbb{Z}_q^{k+1\times k+2}$;
Set $\mathbf{D}^* = (\mathbf{D}^\top, (\mathbf{a}^\top\mathbf{D})^\top)^\top$;
Set $\mathsf{sk} = (\mathbf{a}, \mathbf{f}, \mathbf{g}, \mathbf{F}, \mathbf{G})$ and
Set $\mathsf{pk} = (\ [\mathbf{D}]_1, [\mathbf{E}]_2, [\mathbf{a}^\top\mathbf{D}]_1, [\mathbf{f}^\top\mathbf{D}]_T,$
$[\mathbf{F}^\top\mathbf{D}]_1, [\mathbf{g}^\top\mathbf{E}]_T, [\mathbf{G}^\top\mathbf{E}]_2, [\mathbf{G}\mathbf{D}^*]_1, [\mathbf{F}\mathbf{E}]_2)$;
Output $(\mathsf{pk}, \mathsf{sk})$.

$\underline{\mathsf{Enc}(\mathsf{pk}, [\mathtt{M}]_1)}$:
Sample $\mathbf{r}, \mathbf{s} \leftarrow_\$ \mathbb{Z}_q^k$;
$[\mathbf{u}]_1 \leftarrow [\mathbf{D}]_1 \cdot \mathbf{r}$, $[p]_1 \leftarrow [\mathbf{a}^\top\mathbf{D}]_1 \cdot \mathbf{r} + [\mathtt{M}]_1$;
$[\mathbf{x}]_1 \leftarrow ([\mathbf{u}^\top]_1, [p]_1)^\top$;
$[\mathbf{v}]_2 \leftarrow [\mathbf{E}]_2 \cdot \mathbf{s}$;
$[\pi_1]_T = [\mathbf{f}^\top\mathbf{D}]_T \cdot \mathbf{r} + e([\mathbf{F}^\top\mathbf{D}]_1 \cdot \mathbf{r}, [\mathbf{v}]_2)$;
$[\pi_2]_T = [\mathbf{g}^\top\mathbf{E}]_T \cdot \mathbf{s} + e([\mathbf{x}]_1, [\mathbf{G}^\top\mathbf{E}]_2 \cdot \mathbf{s})$;
Set $\pi = \pi_1 + \pi_2$;
Output $\mathtt{C} = ([\mathbf{x}]_1, [\mathbf{v}]_2, [\pi]_T)$;

$\underline{\mathsf{Dec}(\mathsf{sk}, \mathtt{C})}$:
Parse $\mathtt{C} = ([\mathbf{x}]_1, [\mathbf{v}]_2, \pi)$;
parse $[\mathbf{x}^\top]_1 = ([\mathbf{u}^\top]_1, [p]_1)$;
set $[\mathtt{M}]_1 \leftarrow [p]_1 - [\mathbf{a}^\top\mathbf{u}]_1$;
set $[\pi_1]_T \leftarrow [(\mathbf{f} + \mathbf{F}\mathbf{v})^\top\mathbf{u}]_T$;
set $[\pi_2]_T \leftarrow [(\mathbf{g} + \mathbf{G}\mathbf{x})^\top\mathbf{v}]_T$;
If $\pi \neq \pi_1 + \pi_2$ then output $\perp$
else output $[\mathtt{M}]_1$.

$\underline{\mathsf{Rand}(\mathsf{pk}, \mathtt{C})}$:
Parse $\mathtt{C} = ([\mathbf{x}]_1, [\mathbf{v}]_2, [\pi]_T)$, $[\mathbf{x}^\top]_1 = ([\mathbf{u}^\top]_1, [p]_1)$;
Sample $\hat{\mathbf{r}}, \hat{\mathbf{s}} \leftarrow_\$ \mathbb{Z}_q^k$
$[\hat{\mathbf{x}}]_1 \leftarrow [\mathbf{x}]_1 + [\mathbf{D}^*]_1 \cdot \hat{\mathbf{r}}$;
$[\hat{\mathbf{v}}]_2 \leftarrow [\mathbf{v}]_2 + [\mathbf{E}]_2 \cdot \hat{\mathbf{s}}$;
$[\hat{\pi}_1]_T = [\mathbf{f}^\top\mathbf{D}]_T \cdot \hat{\mathbf{r}} + e([\mathbf{F}^\top\mathbf{D}]_1 \cdot \hat{\mathbf{r}}, [\hat{\mathbf{v}}]_2) + e([\mathbf{u}]_1, [\mathbf{F}\mathbf{E}]_2 \cdot \hat{\mathbf{s}})$;
$[\hat{\pi}_2]_T = [\mathbf{g}^\top\mathbf{E}]_T \cdot \hat{\mathbf{s}} + e([\hat{\mathbf{x}}]_1, [\mathbf{G}^\top\mathbf{E}]_2 \cdot \hat{\mathbf{s}}) + e([\mathbf{G}\mathbf{D}^*]_1 \cdot \hat{\mathbf{r}}, [\mathbf{v}]_2)$;
Output the ciphertext $\hat{\mathtt{C}} = ([\hat{\mathbf{x}}]_1, [\hat{\mathbf{v}}]_2, [\hat{\pi}]_T)$, with $[\hat{\pi}]_T \leftarrow [\pi]_T + [\hat{\pi}_1]_T + [\hat{\pi}_2]_T$.

Fig. 3: Our Rand-RCCA encryption scheme $\mathcal{PKE}_1$ based on the $\mathcal{D}_k$-MDDH assumption for $k \in \mathbb{N}^*$.

Lemma 1. *Let k be a positive integer. For any matrices $\mathbf{D} \in \mathbb{Z}_q^{k+1\times k}, \mathbf{E} \in \mathbb{Z}_q^{k+1\times k}$ and any (possibly unbounded) adversary A:*

$$\Pr\left[\begin{array}{c} \mathbf{u} \notin \mathit{span}(\mathbf{D}) \\ (\mathbf{v} - \mathbf{v}^*) \notin \mathit{span}(\mathbf{E}) \\ z = (\mathbf{f} + \mathbf{F}\mathbf{v})^\top\mathbf{u} \end{array} \middle| \begin{array}{l} \mathbf{f} \leftarrow_\$ \mathbb{Z}_q^{k+1}, \mathbf{F} \leftarrow_\$ \mathbb{Z}_q^{k+1\times k+1}; \\ (z, \mathbf{u}, \mathbf{v}) \leftarrow_\$ \mathsf{A}^{\mathcal{O}(\cdot)}(\mathbf{D}, \mathbf{E}, \mathbf{D}^\top\mathbf{f}, \mathbf{D}^\top\mathbf{F}, \mathbf{F}\mathbf{E}) \end{array}\right] \leq 1/q,$$

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

Proof. Let $\mathbf{K} = \Big(\mathbf{f}, \mathbf{F}\Big) \in \mathbb{Z}_q^{k+1\times k+2}$. We can rewrite the information that the adversary sees about $\mathbf{f}, \mathbf{F}$ in matrix form:

$$\Big(\mathbf{D}, \mathbf{E}, \mathbf{D}^\top\mathbf{f}, \mathbf{D}^\top\mathbf{F}, \mathbf{F}\mathbf{E}, \mathbf{f} + \mathbf{F} \cdot \mathbf{v}^*\Big) = \left(\mathbf{D}, \mathbf{E}, \mathbf{D}^\top\mathbf{K}, \mathbf{K}\begin{pmatrix}\mathbf{0}\\ \mathbf{E}\end{pmatrix}, \mathbf{K}\begin{pmatrix}1\\ \mathbf{v}^*\end{pmatrix}\right).$$

We now have to argue that $z = \mathbf{u}^\top\mathbf{K}\begin{pmatrix}1\\ \mathbf{v}\end{pmatrix}$ is independent of the adversary's view when $\mathbf{u} \notin \mathit{span}(\mathbf{D})$ and $(\mathbf{v} - \mathbf{v}^*) \notin \mathit{span}(\mathbf{E})$. Without loss of generality we assume the matrices $\mathbf{D}, \mathbf{E}$ to be full rank. Otherwise this means there is a redundancy in the information provided to the adversary and this clearly does not give him more chances of being successful. Define the following matrices:

$$\tilde{\mathbf{D}} = \Big(\mathbf{D}, \mathbf{u}\Big) \in \mathbb{Z}_q^{k+1\times k+1}, \qquad \tilde{\mathbf{E}} = \begin{pmatrix}\mathbf{0}, & 1, & 1\\ \mathbf{E}, & \mathbf{v}^*, & \mathbf{v}\end{pmatrix} \in \mathbb{Z}_q^{k+2\times k+2}.$$

By the condition that $\mathbf{u} \notin span(\mathbf{D})$ and $(\mathbf{v} - \mathbf{v}^*) \notin span(\mathbf{E})$, $\tilde{\mathbf{D}}$ and $\tilde{\mathbf{E}}$ are invertible matrices.

Let us consider the matrix $\mathbf{Z} = \tilde{\mathbf{D}}^\top \mathbf{K} \tilde{\mathbf{E}} \in \mathbb{Z}_q^{k+1 \times k+2}$ and the information that the adversary has on this matrix. Note that for $z_{k+1,k+2}$, namely the term in last row and last column of $\mathbf{Z}$, the following holds:

$$z_{k+1,k+2} = \mathbf{u}^\top \mathbf{K} \begin{pmatrix} 1 \\ \mathbf{v} \end{pmatrix} = z.$$

Since the view of the adversary contains invertible matrix $\tilde{\mathbf{E}}$, knowledge of $\mathbf{D}^\top \mathbf{K}$ (in the view of the adversary) is equivalent to knowledge of $\mathbf{D}^\top \mathbf{K} \tilde{\mathbf{E}}$, which are the first k rows of $\mathbf{Z}$.

Similarly, let $\hat{\mathbf{E}}$ be the first $k+1$ columns of $\tilde{\mathbf{E}}$, since $\tilde{\mathbf{D}}$ is invertible and is known by the adversary, knowledge of $\mathbf{K}\hat{\mathbf{E}}$ (in the view of the adversary) is equivalent to knowledge of $\tilde{\mathbf{D}}^\top \mathbf{K} \hat{\mathbf{E}}$, the first $k+1$ columns of $\mathbf{Z}$. Therefore, the view of the adversary includes all the matrix $\mathbf{Z}$ except for $z_{k+1 \times k+2}$.

On the other hand, since $\tilde{\mathbf{D}}$ and $\tilde{\mathbf{E}}$ are invertible matrices, if we see $\mathbf{Z} = \tilde{\mathbf{D}}^\top \mathbf{K} \tilde{\mathbf{E}} \in \mathbb{Z}_q^{k+1 \times k+2}$ as a system of equations with unknown $\mathbf{K}$, there exists a unique solution $\mathbf{K}$ for any choice of $\mathbf{Z}$, namely, $\mathbf{K} = (\tilde{\mathbf{D}}^\top)^{-1} \mathbf{Z} \tilde{\mathbf{E}}^{-1}$.

Therefore, from the point of view of the adversary, every value of $z_{k+1 \times k+2} \in \mathbb{Z}_q$ is equally likely, since $\mathbf{K} \leftarrow_\$ \mathbb{Z}_q^{k+1 \times k+2}$ is sampled uniformly at random. This concludes the proof.

Corollary 1. *Let k be a positive integer. For any matrices $\mathbf{D} \in \mathbb{Z}_q^{k+1 \times k}, \mathbf{E} \in \mathbb{Z}_q^{k+1 \times k}$ and any (possibly unbounded) adversary A:*

$$\Pr\left[\begin{array}{c|l} \mathbf{u} \notin span(\mathbf{D}) & \mathbf{f} \leftarrow_\$ \mathbb{Z}_q^{k+1}, \mathbf{F} \leftarrow_\$ \mathbb{Z}_q^{k+1 \times k+1}; \\ z = (\mathbf{f} + \mathbf{F}\mathbf{v})^\top \mathbf{u} & (z, \mathbf{u}, \mathbf{v}) \leftarrow_\$ \mathsf{A}(\mathbf{D}, \mathbf{E}, \mathbf{D}^\top \mathbf{f}, \mathbf{D}^\top \mathbf{F}, \mathbf{F}\mathbf{E}) \end{array} \right] \leq 1/q.$$

The proof of the corollary is trivial, for any adversary A that does not make oracle queries, we can consider an adversary A' that first receives the output of A and then makes an oracle query for $\mathbf{v}^*$ such that $\mathbf{v} - \mathbf{v}^* \notin span(\mathbf{E})$.

Perfect re-randomizability.

Next, we prove that the scheme has perfect re-randomizability. First, we focus on property (1) which says that the re-randomization of a honest encryption is identically distributed to a fresh encryption.

Let $\mathtt{C} = ([\mathbf{x}]_1, [\mathbf{v}]_2, [\pi]_T)$ be an encryption of $[\mathtt{M}]_1$ with randomness fixed to $\mathbf{r}, \mathbf{s}$, and let $\hat{\mathtt{C}} = ([\hat{\mathbf{x}}]_1, [\hat{\mathbf{v}}]_2, [\hat{\pi}]_T) \leftarrow_\$ \mathsf{Rand}(\mathsf{pk}, \mathtt{C})$ be its re-randomization with randomness $\hat{\mathbf{r}}, \hat{\mathbf{s}}$. We show that $\hat{\mathtt{C}}$ is identically distributed to a fresh encryption of $[\mathtt{M}]_1$ with randomness $(\mathbf{r} + \hat{\mathbf{r}})$ and $(\mathbf{s} + \hat{\mathbf{s}})$. (Notice that for any $\mathbf{r}$ and $\mathbf{s}$, the random variables $(\mathbf{r} + \hat{\mathbf{r}}), (\mathbf{s} + \hat{\mathbf{s}})$ are uniformly distributed.)

It is straightforward to verify that this holds for $([\hat{\mathbf{x}}]_1, [\hat{\mathbf{v}}]_2)$, i.e., $\hat{\mathbf{x}} = \mathbf{D}^*(\mathbf{r} + \hat{\mathbf{r}}) + (\mathbf{0}^\top, \mathtt{M})^\top$ and $\hat{\mathbf{v}} = \mathbf{E}(\mathbf{s} + \hat{\mathbf{s}})$. We show that also $\hat{\pi}$ is correctly distributed:

$$[\hat{\pi}]_T = [\mathbf{f}^\top \mathbf{D}]_T (\mathbf{r} + \hat{\mathbf{r}}) + e([\mathbf{F}^\top \mathbf{D}]_1 (\mathbf{r} + \hat{\mathbf{r}}), [\hat{\mathbf{v}}]_2) + [\mathbf{g}^\top \mathbf{E}]_T (\mathbf{s} + \hat{\mathbf{s}}) + e([\hat{\mathbf{x}}]_1, [\mathbf{G}^\top \mathbf{E}]_2 (\mathbf{s} + \hat{\mathbf{s}})) \quad (2)$$

Let $\mathbf{u}$ (resp. $\hat{\mathbf{u}}$) be the first $k+1$ elements of $\mathbf{x}$ (resp. $\hat{\mathbf{x}}$).

Notice that by the construction of $\hat{\pi}$ in Rand, of π_1 and π_2 in Enc (with the alternative expression proven in (1)), and of $\hat{\pi}_1$ and $\hat{\pi}_2$ in Rand we have

$$\begin{aligned} \hat{\pi} &= \pi_1 + \pi_2 + \hat{\pi}_1 + \hat{\pi}_2 \\ &= (\mathbf{f} + \mathbf{F}\mathbf{v})^\top \mathbf{u} + (\mathbf{g} + \mathbf{G}\mathbf{x})^\top \mathbf{v} + \hat{\pi}_1 + \hat{\pi}_2 \\ &= (\mathbf{f} + \mathbf{F}\mathbf{v})^\top \mathbf{u} + (\mathbf{g} + \mathbf{G}\mathbf{x})^\top \mathbf{v} + \mathbf{f}^\top \mathbf{D}\hat{\mathbf{r}} + (\mathbf{F}^\top \mathbf{D} \cdot \hat{\mathbf{r}})^\top \hat{\mathbf{v}} + \mathbf{u}^\top \mathbf{F}\mathbf{E}\hat{\mathbf{s}} + \\ &\quad \mathbf{g}^\top \mathbf{E}\hat{\mathbf{s}} + \hat{\mathbf{x}}^\top \mathbf{G}^\top \mathbf{E}\hat{\mathbf{s}} + (\mathbf{G}\mathbf{D}^*\hat{\mathbf{r}})^\top \mathbf{v}. \end{aligned}$$

After grouping common terms and by using the definition of $\hat{\mathbf{x}}, \hat{\mathbf{u}}, \hat{\mathbf{v}}$ in Rand, we can rewrite the above equation as

$$\begin{aligned}
&= \mathbf{f}^\top(\mathbf{u} + \mathbf{D}\hat{\mathbf{r}}) + (\mathbf{F}(\mathbf{v} + \mathbf{F}\mathbf{E}\hat{\mathbf{s}}))^\top \mathbf{u} + (\mathbf{g} + \mathbf{G}(\mathbf{x} + \mathbf{D}^*\hat{\mathbf{r}}))^\top \mathbf{v} + (\mathbf{F}\hat{\mathbf{v}})^\top \mathbf{D}\hat{\mathbf{r}} + (\mathbf{g} + \mathbf{G}\hat{\mathbf{x}})^\top \mathbf{E}\hat{\mathbf{s}} \\
&= \mathbf{f}^\top \hat{\mathbf{u}} + (\mathbf{F}\hat{\mathbf{v}})^\top \mathbf{u} + (\mathbf{g} + \mathbf{G}\hat{\mathbf{x}})^\top \mathbf{v} + (\mathbf{F}\hat{\mathbf{v}})^\top \mathbf{D}\hat{\mathbf{r}} + (\mathbf{g} + \mathbf{G}\hat{\mathbf{x}})^\top \mathbf{E}\hat{\mathbf{s}} \\
&= (\mathbf{f} + \mathbf{F}\hat{\mathbf{v}})^\top \hat{\mathbf{u}} + (\mathbf{g} + \mathbf{G}\hat{\mathbf{x}})^\top \hat{\mathbf{v}}.
\end{aligned}$$

Finally, by an argument analogous to that proven in (1) we obtain that equation (2) holds as desired.

Secondly, we show that our algorithm Rand satisfies the second property of Def. 3, namely that the outcome of decryption is preserved. Let $\mathtt{C} = ([\mathbf{x}]_1, [\mathbf{v}]_2, [\pi]_T)$, and let $\hat{\mathtt{C}} = ([\hat{\mathbf{x}}]_1, [\hat{\mathbf{v}}]_2, [\hat{\pi}]_T) \leftarrow_{\$} \mathsf{Rand}(\mathsf{pk}, \mathtt{C})$ be its re-randomization with randomness $\hat{\mathbf{r}}, \hat{\mathbf{s}}$. First, notice that Rand adds to $\mathbf{x}$ an encryption of $[0]_1$, therefore if the first line of $\mathsf{Dec}(\mathsf{sk}, \mathtt{C})$ computes $[\mathtt{M}]_1$, the same holds in $\mathsf{Dec}(\mathsf{sk}, \hat{\mathtt{C}})$.

Second, if $\mathtt{C}$ is valid, then $\pi = (\mathbf{f} + \mathbf{F}\mathbf{v})^\top \mathbf{u} + (\mathbf{g} + \mathbf{G}\mathbf{x})^\top \mathbf{v}$ and by the same proof given above, we have that $\hat{\pi} = (\mathbf{f} + \mathbf{F}\hat{\mathbf{v}})^\top \hat{\mathbf{u}} + (\mathbf{g} + \mathbf{G}\hat{\mathbf{x}})^\top \hat{\mathbf{v}}$. Hence, $\hat{\pi}$ passes the test and thus $\hat{\mathtt{C}}$ is also valid and $\mathsf{Dec}(\mathsf{sk}, \hat{\mathtt{C}}) = \mathsf{Dec}(\mathsf{sk}, \mathtt{C})$. Therefore, we are left with showing that if $\mathsf{Dec}(\mathsf{sk}, \mathtt{C}) = \perp$ then $\mathsf{Dec}(\mathsf{sk}, \hat{\mathtt{C}})$ outputs $\perp$ as well. Assume by contradiction that $\mathsf{Dec}(\mathsf{sk}, \hat{\mathtt{C}}) \neq \perp$, that is $\hat{\pi} = (\mathbf{f} + \mathbf{F}\hat{\mathbf{v}})^\top \hat{\mathbf{u}} + (\mathbf{g} + \mathbf{G}\hat{\mathbf{x}})^\top \hat{\mathbf{v}}$. For the same proof given above (going backward on the equations) we obtain that $\pi = (\mathbf{f} + \mathbf{F}\mathbf{v})^\top \mathbf{u} + (\mathbf{g} + \mathbf{G}\mathbf{x})^\top \mathbf{v}$, and thus $\mathsf{Dec}(\mathsf{sk}, \mathtt{C}) \neq \perp$, which is a contradiction.

Lastly, we show the third property of Def. 3. We reduce to the experiment in the Corollary 1. Consider an adversary A that on input pk, produces a ciphertext $\mathtt{C}$ such the $\mathsf{Dec}(\mathsf{sk}, \mathtt{C}) \neq \perp$ and $\mathtt{C} \notin \mathcal{C}_{\mathsf{pk}}$. In particular, the second property implies that either (1) it does not exist $\mathbf{r}$ such that $\mathbf{u} = \mathbf{D}\mathbf{r}$ or (2) it does not exists $\mathbf{s}$ such that $\mathbf{v} = \mathbf{E}\mathbf{s}$. Let us suppose that (1) occurs (an analogous argument holds for (2)). Then since the ciphertext decrypts we have that $\pi = (\mathbf{f} + \mathbf{F}\mathbf{v})^\top \mathbf{u} + (\mathbf{g} + \mathbf{G}\mathbf{x})^\top \mathbf{v}$, therefore $(\pi - (\mathbf{g} + \mathbf{G}\mathbf{x})^\top \mathbf{v}, \mathbf{u}, \mathbf{v})$ is a valid answer for the experiment in Corollary 1. Specifically, we can consider an adversary A' for the experiment of the corollary that samples all the secret key material, with exception of $\mathbf{D}, \mathbf{E}, \mathbf{f}^\top \mathbf{D}, \mathbf{F}^\top \mathbf{D}, \mathbf{F}\mathbf{E}$ and set the public key accordingly. In particular, A' has full knowledge of the vector $\mathbf{g}$ and the matrix $\mathbf{G}$. The adversary A' forwards the public key pk to A and then receives $\mathtt{C}$. From such ciphertext and with the knowledge of $\mathbf{g}, \mathbf{G}$ it can compute its guess.

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

Theorem 1. *For any matrix distribution $\mathcal{D}_k$ such that the $\mathcal{D}_k$-MDDH assumption holds for the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ generated by GGen, the Rand-PKE scheme $\mathcal{PKE}_1$ described above is RCCA-secure.*

Proof. We start by describing a sequence of hybrid games. For readability purposes, we underline the main differences between each consecutive hybrid. In hybrids $\mathbf{H}_0$ and from $\mathbf{H}_3$ until $\mathbf{H}_7$ we progressively change the way the decryption procedure works. In the description of the games, the changes correpond to the underlined formulae. We summarize the main changes in Fig. 4.

Hybrid $\mathbf{H}_0$. This hybrid experiment is equivalent to the RCCA experiment described in Fig. 1 but the oracle $\mathsf{Dec}^\diamond$ is instantiated with a slightly different decryption procedure. Decryption proceeds exactly as in the description of the PKE scheme, except that, before setting each variable $\mathtt{M}, \pi_1, \pi_2$ it additionally checks if the variable was not set already. For future reference, we label these commands as the decryption rule (*).

Notice that, in this hybrid, this change is merely syntactical, as at each invocation of the decryption procedure all the three variables are unset. The hybrid $\mathbf{H}_0$ is equivalent to the experiment $\mathbf{Exp}^{\mathtt{RCCA}}_{\mathsf{A},\mathcal{PKE}}(\lambda)$ of Fig. 1.

Procedure $\mathsf{Dec}^*(\mathsf{sk}, \mathtt{C})$:
Parse $\mathtt{C} = ([\mathbf{x}]_1, [\mathbf{v}]_2, [\pi]_T)$ and $[\mathbf{x}^\top]_1 = ([\mathbf{u}^\top]_1, [p]_1)$
(i) If $\mathbf{u} \in span(\mathbf{D})$, let $\mathbf{u} = \mathbf{D}\mathbf{r}$ then
$[\mathtt{M}]_1 \leftarrow [p - \mathbf{a}^\top \mathbf{D}\mathbf{r}]_1$;
$[\pi_1]_T \leftarrow [(\mathbf{f}^\top \mathbf{D} + \mathbf{v}^\top \mathbf{F}^\top \mathbf{D})\mathbf{r}]_T$;
(ii) If $\mathbf{v} \in span(\mathbf{E})$, let $\mathbf{v} = \mathbf{E}\mathbf{s}$ then
$[\pi_2]_T \leftarrow [(\mathbf{g}_0^\top \mathbf{E} + \mathbf{x}^\top \mathbf{G}^\top \mathbf{E})\mathbf{s}]_T$;
(iii) If $\mathbf{u} \notin span(\mathbf{D})$ and $(\mathbf{v} - \mathbf{v}^* \notin span(\mathbf{E})$ or $\mathbf{v}^*$ unset) then output $\perp$.
(iv) If $\mathbf{v} \notin span(\mathbf{E})$ and $(\mathbf{x} - \mathbf{x}^* \notin span(\mathbf{D}^*)$ or $\mathbf{u}^*$ unset) then output $\perp$.
(v) If $\mathbf{x} - \mathbf{x}^* \in span(\mathbf{D}^*)$ and $\mathbf{v} - \mathbf{v}^* \in span(\mathbf{E})$ then
$\mathtt{M} \leftarrow \diamond$;
$[\pi_1]_T \leftarrow [\pi^*]_T + [(\mathbf{f}^\top \mathbf{D} + \tilde{\mathbf{v}}^\top \mathbf{F}^\top \mathbf{D})\tilde{\mathbf{x}}]_T$
$[\pi_2]_T \leftarrow [(\mathbf{g}_0^\top \mathbf{E} + \tilde{\mathbf{x}}^\top \mathbf{G}^\top \mathbf{E})\tilde{\mathbf{x}}]_T$
(*) If $[\mathtt{M}]_1$ is unset set $[\mathtt{M}]_1 \leftarrow [p]_1 - \mathbf{a}^\top [\mathbf{u}]$;
(*) If $[\pi_1]_T$ is unset set $[\pi_1]_T \leftarrow [(\mathbf{f} + \mathbf{F}\mathbf{v})^\top \mathbf{u}]_T$;
(*) If $[\pi_2]_T$ is unset set $[\pi_2]_T \leftarrow [(\mathbf{g}_0 + \mathbf{G}\mathbf{x})^\top \mathbf{v}]_T$;
If $[\pi]_T = [\pi_1]_T + [\pi_2]_T$ output $\mathtt{M}$ else $\perp$.

Fig. 4: The decryption procedure in the hybrids experiment. The decryption procedure of the hybrid $\mathbf{H}_0$ executes only the rules (*) and the last decryption check. The decryption procedure of the hybrid $\mathbf{H}_3$ additionally executes (i) and (ii). The decryption procedure of the hybrid $\mathbf{H}_4$ additionally executes (iii). The decryption procedure of the hybrid $\mathbf{H}_5$ additionally executes (iv). The decryption procedure of the hybrid $\mathbf{H}_6$ additionally executes (v). The decryption procedure of the hybrid $\mathbf{H}_7$ stops to execute the rules (*).

Hybrid $\mathbf{H}_1$. The hybrid $\mathbf{H}_1$ is the same as $\mathbf{H}_0$ but it computes the challenge ciphertext $\mathtt{C}^* = ([\mathbf{x}^*]_1, [\mathbf{v}^*]_2, [\pi^*]_T)$ by using the secret key. Specifically:

$$[\mathbf{u}^*]_1 \leftarrow [\mathbf{D}]_1 \cdot \mathbf{r}^*, \quad [p^*]_1 \leftarrow \underline{\mathbf{a}^\top \cdot [\mathbf{u}^*]_1 + [\mathtt{M}_{b^*}]_1} \qquad \text{where } \mathbf{r}^* \leftarrow\!\$\ \mathbb{Z}_q^k$$
$$[\mathbf{v}^*]_2 \leftarrow [\mathbf{E}]_2 \cdot \mathbf{s}^* \qquad \text{where } \mathbf{s}^* \leftarrow\!\$\ \mathbb{Z}_q^k$$
$$[\pi_1^*]_T \leftarrow \underline{e([\mathbf{u}^*]_1, [\mathbf{f}]_2 + \mathbf{F} \cdot [\mathbf{v}^*]_2)},$$
$$[\pi_2^*]_T \leftarrow \underline{e([\mathbf{g}]_1 + \mathbf{G} \cdot [\mathbf{x}^*]_1, [\mathbf{v}^*]_2)},$$

where $\mathbf{x}^*$ is $((\mathbf{u}^*)^\top, p^*)^\top$ and $\pi^* = \pi_1^* + \pi_2^*$.

Notice that $[\pi_1^*]_T$ and $[\pi_2^*]_T$ can be efficiently computed using the secret key and the pairing function. The only differences introduced are in the way we compute $[p^*]_1$ and $[\pi^*]_T$. However, notice that such differences are only syntactical, as, by the correctness of the scheme, we compute exactly the same values the hybrid $\mathbf{H}_0$ would compute.

Hybrid $\mathbf{H}_2$. The hybrid $\mathbf{H}_2$ is the same as $\mathbf{H}_1$ but the challenger, upon challenge messages $[\mathtt{M}_0]_1, [\mathtt{M}_1]_1 \in \mathbb{G}_1$, computes the challenge ciphertext $\mathtt{C}^* = ([\mathbf{x}^*]_1, [\mathbf{v}^*]_2, [\pi^*]_T)$ where $\mathbf{x}^*$ is $((\mathbf{u}^*)^\top, p^*)^\top$ by sampling :

$$\underline{\mathbf{u}^* \leftarrow\!\$\ \mathbb{Z}_q^{k+1} \setminus span(\mathbf{D})} \qquad \underline{\mathbf{v}^* \leftarrow\!\$\ \mathbb{Z}_q^{k+1} \setminus span(\mathbf{E})}.$$

The hybrids $\mathbf{H}_1$ and $\mathbf{H}_2$ are computationally indistinguishable. This follows by applying the $\mathcal{D}_k$-MDDH Assumption on $[\mathbf{D}, \mathbf{u}^*]_1$ in $\mathbb{G}_1$ and $[\mathbf{E}, \mathbf{v}^*]_2$ in $\mathbb{G}_2$, respectively, and then a standard statistical argument to show that sampling $\mathbf{u}^*$ uniformly at random in $\mathbb{Z}_q^{k+1}$ is statistically close to sampling it at random in $\mathbb{Z}_q^{k+1} \setminus span(\mathbf{D})$. The reduction is straightforward and is omitted.

From now on, we prove that each pair of consecutive hybrids is statistically close. In particular, this means that the hybrids (and in principle also the adversary) are allowed to run in unbounded time.

Hybrid $\mathbf{H}_3$. The hybrid $\mathbf{H}_3$ is the same as $\mathbf{H}_2$ but adds the following decryption rules that upon input a ciphertext $([\mathbf{u}]_1, [p]_1, [\mathbf{v}]_2, [\pi]_T)$:

(i) If $\mathbf{u} = \mathbf{D}\mathbf{r}$ for some $\mathbf{r} \in \mathbb{Z}_q^k$, then compute

$$\underline{[\pi_1]_T \leftarrow [(\mathbf{f}^\top \mathbf{D} + \mathbf{v}^\top \mathbf{F}^\top \mathbf{D})]_T \cdot \mathbf{r}} \qquad \underline{[\mathtt{M}]_1 \leftarrow [p]_1 - [\mathbf{a}^\top \mathbf{D}]_1 \cdot \mathbf{r}}$$

(ii) If $\mathbf{v} = \mathbf{E}\mathbf{s}$ for some $\mathbf{s} \in \mathbb{Z}_q^k$, letting $\mathbf{x} = (\mathbf{u}^\top, p)^\top$, then compute:

$$\underline{[\pi_2]_T \leftarrow [(\mathbf{g}^\top \mathbf{E} + \mathbf{x}^\top \mathbf{G}^\top \mathbf{E})]_T \cdot \mathbf{s}}$$

Specifically, in the first rule the decryption of $\mathtt{M}$ and π_1 are computed using the public key components $[\mathbf{a}^\top \mathbf{D}]_1, [\mathbf{f}^\top \mathbf{D}]_T$ and $[\mathbf{F}^\top \mathbf{D}]_1$ instead of the secret key components $\mathbf{a}, \mathbf{f}, \mathbf{F}$ for all the ciphertexts with $\mathbf{u} \in \mathit{span}(\mathbf{D})$. Recall that this strategy is not efficient, but it is possible because the simulator does not need to run in polynomial time (since we want to argue the games are statistically close). If $\mathbf{v} = \mathbf{E}\mathbf{s}$, then by the second rule, the hybrid computes the proof π_2 using only the components $[\mathbf{g}^\top \mathbf{E}]_T$ and $[\mathbf{G}^\top \mathbf{E}]_2$ of the public key.

We notice that, again by correctness of the PKE scheme, the computation of π_1, π_2 and $\mathtt{M}$ in the hybrids $\mathbf{H}_3$ and $\mathbf{H}_2$ is equivalent. In particular, let π_1' be the proof as computed in $\mathbf{H}_2$, then $[\pi_1']_T = [(\mathbf{f} + \mathbf{F}\mathbf{v})^\top \mathbf{u}]_T = [(\mathbf{f} + \mathbf{F}\mathbf{v})^\top \mathbf{D}\mathbf{r}]_T = [(\mathbf{f}^\top \mathbf{D} + \mathbf{v}^\top \mathbf{F}^\top \mathbf{D})]_T \cdot \mathbf{r} = [\pi_1]_T$. (An equivalent derivation holds for π_2 and $\mathtt{M}$.) The difference is then only syntactical.

Hybrid $\mathbf{H}_4$. The hybrid $\mathbf{H}_4$ is the same as $\mathbf{H}_3$ but adds the following decryption rule, on input a ciphertext $\mathtt{C} = ([\mathbf{u}]_1, [p]_1, [\mathbf{v}]_2, [\pi]_T)$:

(iii) If $\mathbf{u} \notin \mathit{span}(\mathbf{D})$ and ($\mathbf{v} - \mathbf{v}^* \notin \mathit{span}(\mathbf{E})$ or $\mathbf{v}^*$ is unset) then output $\perp$.

Recall that the challenge ciphertext is $\mathtt{C}^* = ([\mathbf{u}^*]_1, [p^*]_1, [\mathbf{v}^*]_2, [\pi]_T)$. Notice that we check either if $\mathbf{v} - \mathbf{v}^* \notin \mathit{span}(\mathbf{E})$ or $\mathbf{v}^*$ is unset. We do so to handle simultaneously the decryption queries before and after the challenge ciphertext is computed. In particular, before the challenge ciphertext is computed the decryption rule simply checks if $\mathbf{u} \notin \mathit{span}(\mathbf{D})$ (as in the classical Cramer-Shoup proof strategy).

We show in Lemma 3 that $\mathbf{H}_4$ is statistically close to $\mathbf{H}_3$. Here we continue describing the hybrid games.

Hybrid $\mathbf{H}_5$. The hybrid $\mathbf{H}_5$ is the same as $\mathbf{H}_4$ but adds the following decryption rule, on input a ciphertext $\mathtt{C} = ([\mathbf{x}]_1, [\mathbf{v}]_2, [\pi]_T)$:

(iv) If $\mathbf{v} \notin \mathit{span}(\mathbf{E})$ and ($\mathbf{x} - \mathbf{x}^* \notin \mathit{span}(\mathbf{D}^*)$ or $\mathbf{x}^*$ is unset) then output $\perp$.

We show in Lemma 7 (Appendix A) that $\mathbf{H}_5$ is statistically close to $\mathbf{H}_4$. The proof of the lemma is almost identical to the proof of Lemma 3.

Hybrid $\mathbf{H}_6$. The hybrid $\mathbf{H}_6$ is the same as $\mathbf{H}_5$ but adds the following decryption rule, on input a ciphertext $\mathtt{C} = ([\mathbf{x}]_1, [\mathbf{v}]_2, [\pi]_T)$:

(v) If $\mathbf{x} - \mathbf{x}^* \in \mathit{span}(\mathbf{D}^*)$ and $\mathbf{v} - \mathbf{v}^* \in \mathit{span}(\mathbf{E})$ then let $\tilde{\mathbf{r}}, \tilde{\mathbf{s}}$ be such that $\mathbf{x} - \mathbf{x}^* = \tilde{\mathbf{x}} = \mathbf{D}\tilde{\mathbf{r}}$ and $\mathbf{v} - \mathbf{v}^* = \tilde{\mathbf{v}} = \mathbf{E}\tilde{\mathbf{s}}$, and compute $[\pi_1]_T, [\pi_2]_T$ as follows:

$$[\pi_1]_T \underline{\leftarrow [\pi^*]_T + [(\mathbf{f}^\top \mathbf{D} + \tilde{\mathbf{v}}^\top \mathbf{F}^\top \mathbf{D})\tilde{\mathbf{x}}]_T},$$
$$[\pi_2]_T \underline{\leftarrow [(\mathbf{g}\mathbf{E} + \tilde{\mathbf{x}}^\top \mathbf{G}^\top \mathbf{E})\tilde{\mathbf{v}}]_T},$$

This hybrid is equivalent to $\mathbf{H}_5$. The conditions of the decryption rule (v) imply that, if the proof π is correct, then the ciphertext $\mathtt{C}$ is a re-randomization of $\mathtt{C}^*$.

Hybrid $\mathbf{H}_7$. The hybrid $\mathbf{H}_7$ is the same as $\mathbf{H}_6$ but its decryption procedure does not execute the rules (*) introduced in the hybrid $\mathbf{H}_0$.

In Lemma 4 we show that $\mathbf{H}_7$ and $\mathbf{H}_6$ are identically distributed, while in the following we prove that the challenge bit b^* is perfectly hidden

in the adversary's view.

Lemma 2. $\Pr[\mathbf{H}_7 = 1] = \frac{1}{2}$.

Proof. We notice that in $\mathbf{H}_7$ the decryption procedure does not use the secret key $\mathbf{a}$ to perform the decryption; this can be easily confirmed by inspection of the decryption procedure in Fig. 4. Notice also that given the value $\mathbf{a}^\top\mathbf{D}$ the random variable $\mathbf{a}^\top \cdot \mathbf{u}^*$ is uniformly distributed. Thus, both the challenge ciphertext $\mathtt{C}^*$ and the answers of the decryption oracle are independent of the bit b^*.

Lemma 3. *The hybrids* $\mathbf{H}_4$ *and* $\mathbf{H}_3$ *are statistically close.*

Proof. We prove the statement with a hybrid argument over the number of decryption queries of the adversary. Let the hybrid $\mathbf{H}_{3,i}$ be the experiment that answers the first i-th oracle queries as in $\mathbf{H}_4$ (namely, considering the decryption rule (iii)) and answers the remaining queries as in $\mathbf{H}_3$. Let Q_D be the number of decryption queries performed by the adversary A. It is easy to check that $\mathbf{H}_{3,0} \equiv \mathbf{H}_3$ and $\mathbf{H}_{3,Q_D} \equiv \mathbf{H}_4$.

On the other hand $\mathbf{H}_{3,i}$ and $\mathbf{H}_{3,i+1}$ differ when the $(i+1)$-th ciphertext $\mathtt{C} = (([\mathbf{u}]_1, [p]_1), [\mathbf{v}]_2, [\pi]_T)$ is such that "$\mathbf{u} \notin \mathit{span}(\mathbf{D})$ and $((\mathbf{v} - \mathbf{v}^*) \notin \mathit{span}(\mathbf{E})$ or $\mathbf{v}^*$ is unset)", but the decryption oracle (as it would be computed in $\mathbf{H}_3$) outputs a value different from $\perp$. In particular, the latter implies that the proof $[\pi]_T$ verifies correctly. Let $\mathtt{Sound_i}$ be such event. To conclude the proof of the lemma we prove the following proposition. Then a standard union bound gives us that the statistical distance between $\mathbf{H}_4$ and $\mathbf{H}_3$ is at most Q_D/q, which is negligible.

Proposition 1. $\Pr[\mathtt{Sound}_i] \leq 1/q$.

Proof. We reduce an adversary A that causes event $\mathtt{Sound}_i$ to occur into an adversary A' for the game of Lemma 1. Namely, we define an adversary A' for the experiment in the lemma which internally simulates the experiment $\mathbf{H}_{3,i+1}$ running with the adversary A.

Adversary $\mathsf{A}'(\mathbf{D}, \mathbf{E}, \mathbf{f}^\top\mathbf{D}, \mathbf{F}^\top\mathbf{D}, \mathbf{FE})$ with oracle access to $\mathcal{O}$:

1. Sample $\mathbf{a} \leftarrow_\$ \mathbb{Z}_q^{k+1}, \mathbf{g} \leftarrow_\$ \mathbb{Z}_q^{k+1}, \mathbf{G} \leftarrow_\$ \mathbb{Z}_q^{k+1\times k+2}$.
2. Set the public key as:

$$\mathsf{pk} = \begin{pmatrix} [\mathbf{D}]_1, [\mathbf{E}]_2, [\mathbf{a}^\top\mathbf{D}]_1, [\mathbf{f}^\top\mathbf{D}]_T, [\mathbf{F}^\top\mathbf{D}]_1, \\ [\mathbf{g}^\top\mathbf{E}]_T, [\mathbf{G}^\top\mathbf{E}]_2, [\mathbf{GD}^*]_1, [\mathbf{FE}]_2 \end{pmatrix}$$

   as described by the key generation algorithm and set the secret key $\mathsf{sk} = (\mathbf{a}, \cdot, \mathbf{g}, \cdot, \mathbf{G})$.
3. Run the adversary A with input the public key pk. Answer the j-th decryption oracle query with ciphertext $\mathtt{C} = ([\mathbf{u}]_1, [p]_1, [\mathbf{v}]_2, [\pi]_T)$ as follows:
   (a) If $j \leq i$ and $\mathbf{u} \in \mathit{span}(\mathbf{D})$ compute, let $\mathbf{u} = \mathbf{Dr}$:

$$[\mathtt{M}]_1 \leftarrow [p - \mathbf{a}^\top\mathbf{D} \cdot \mathbf{r}]_1, \qquad [\pi_1]_T \leftarrow [(\mathbf{f}^\top\mathbf{D} + \mathbf{v}^\top \cdot \mathbf{F}^\top\mathbf{D})]_T \cdot \mathbf{r},$$
$$[\pi_2]_T \leftarrow [(\mathbf{g} + \mathbf{G} \cdot \mathbf{x})^\top \cdot \mathbf{v}]_T$$

   If $\pi = \pi_1 + \pi_2$ then answer with $[\mathtt{M}]_1$, else anwer $\perp$;
   (b) If $\mathbf{u} \notin \mathit{span}(\mathbf{D})$ answer $\perp$;
   (c) If $j = i + 1$ then stop and return $(\pi - (\mathbf{g} + \mathbf{Gx})^\top\mathbf{v}, \mathbf{u}, \mathbf{v})$.
4. Eventually, A outputs $[\mathtt{M}_0]_1, [\mathtt{M}_1]_1$. Sample $\mathbf{v}^* \leftarrow_\$ \mathbb{Z}_q^{k+1} \setminus \mathit{span}(\mathbf{E})$, and sample $\mathbf{u}^* \leftarrow_\$ \mathbb{Z}_q^{k+1} \setminus \mathit{span}(\mathbf{D})$, query the oracle $\mathcal{O}$ with the element $\mathbf{v}^*$ and receive $\mathit{\Pi} = \mathbf{f} + \mathbf{F} \cdot \mathbf{v}^*$. Set $p^* = \mathbf{a}^\top\mathbf{u}^* + \mathtt{M}_{b^*}$ and $\mathbf{x}^* = ((\mathbf{u}^*)^\top, p^*)^\top$, and:

$$[\pi^*]_T \leftarrow [\mathit{\Pi}^\top \cdot \mathbf{u}^* + (\mathbf{g} + \mathbf{Gx}^*)^\top\mathbf{v}]_T \tag{3}$$

   and send to the adversary the challenge ciphertext $\mathtt{C}^* = ([\mathbf{c}^*]_1, [p^*]_1, [\mathbf{v}]_2, [\pi^*]_T)$.

5. Answer the j-th decryption oracle query with ciphertext $\mathtt{C} = ([\mathbf{u}]_1, [p]_1, [\mathbf{v}]_2, [\pi]_T)$ as follows:
   (a) If $j \leq i$ and $\mathbf{u} \in span(\mathbf{D})$ execute the same as in step 3a.
   (b) If $j \leq i$ and $\mathbf{u} \notin span(\mathbf{D})$ do as follows:
      i. if $(\mathbf{v}^* - \mathbf{v}) \in span(\mathbf{E})$ let $\mathbf{v} = \mathbf{v}^* + \mathbf{E}\boldsymbol{\gamma}$, compute
      $$[\pi_1]_T \leftarrow [(\mathit{\Pi} + \mathbf{F}\mathbf{E}\boldsymbol{\gamma})^\top)\mathbf{u}]_T, \qquad [\pi_2]_T \leftarrow [(\mathbf{g}^\top + \mathbf{G}\mathbf{x})^\top \mathbf{v}]_T$$
      if $\pi = \pi_1 + \pi_2$ then answer $[p - \mathbf{a}^\top \cdot \mathbf{u}]_1$ else answer $\perp$.
      ii. if $(\mathbf{v}^* - \mathbf{v}) \notin span(\mathbf{E})$ then output $\perp$.
   (c) If $j = i + 1$ then stop and return $(\pi - (\mathbf{g} + \mathbf{G}\mathbf{x})^\top \mathbf{v}, \mathbf{u}, \mathbf{v})$.

We show that the adversary perfectly simulates the hybrid $\mathbf{H}_{3,i}$ up to the i-th decryption query. By inspection, it is easy to check that up to step 3, the simulation is perfect[9].

More interestingly, at step 4 the adversary A' uses its oracle to compute $\mathit{\Pi} = \mathbf{f} + \mathbf{F}\mathbf{v}^*$. Thanks to this information the adversary can compute the challenge ciphertext exactly as the hybrid experiment would do as shown in eq. 3. After this step, the adversary A' can easily answer the decryption queries whenever $j \leq i$ and $\mathbf{u} \in span(\mathbf{D})$ or $\mathbf{u} \notin span(\mathbf{D})$ and $(\mathbf{v}^* - \mathbf{v}) \notin span(\mathbf{E})$. We show that the answers for the decryption queries where $j \leq i$, $\mathbf{u} \notin span(\mathbf{D})$ and $(\mathbf{v}^* - \mathbf{v}) \in span(\mathbf{E})$ are distributed exactly as in the hybrid experiment, in fact:

$$\begin{aligned}(\mathit{\Pi} + \mathbf{F}\mathbf{E}\boldsymbol{\gamma})^\top \mathbf{u} &= \mathbf{f}^\top \mathbf{u} + (\mathbf{F}\mathbf{v}^*)^\top \mathbf{u} + (\mathbf{F}\mathbf{E}\boldsymbol{\gamma})^\top \mathbf{u} \\ &= \mathbf{f}^\top \mathbf{u} + (\mathbf{F}(\mathbf{v}^* + \mathbf{E}\boldsymbol{\gamma}))^\top \mathbf{u} \\ &= (\mathbf{f} + \mathbf{F}\mathbf{v})^\top \mathbf{u}.\end{aligned}$$

Finally, by definition of $\mathtt{Sound}_i$, the adversary A at the $(j+1)$-th query outputs a ciphertext that would correctly decrypt in the hybrid experiment and where $\mathbf{u} \notin span(\mathbf{D})$ and $(\mathbf{v}^* - \mathbf{v}) \notin span(\mathbf{E})$ with probability $\Pr[\mathtt{Sound}_i]$. Since the ciphertext correctly decrypts, it means that $\pi = (\mathbf{f} + \mathbf{F}\mathbf{v})^\top \mathbf{u} + (\mathbf{g} + \mathbf{G}\mathbf{x})^\top \mathbf{v}$, therefore the output of A' is a valid guess for the experiment of Lemma 1. However, the adversary A' can win with probability at most $1/q$, and thus the lemma follows.

Lemma 4. *The hybrids $\mathbf{H}_6$ and $\mathbf{H}_7$ are identically distributed.*

Proof. We prove this lemma by showing that in $\mathbf{H}_6$ the decryption procedure never executes the lines with rules (*). To do this, for any ciphertext queried to the decryption oracle we partition over all possible cases and show that the decryption procedure used for the oracle queries either sets the values $\mathtt{M}, \pi_1, \pi_2$ (and thus the rules (*) are not executed) or it stops before reaching those rules as it outputs $\perp$ or $\diamond$. Let $\mathtt{C} = ([\mathbf{x}]_1, [\mathbf{v}]_2, [\pi]_T)$ be the ciphertext queried to the oracle, where $[\mathbf{x}^\top]_1 = ([\mathbf{u}^\top]_1, [p]_1)$. We consider all the possible alternatives:

– $\mathbf{u} \in span(\mathbf{D})$: notice that in this case, by the rule (i), $\mathtt{M}$ and π_1 are set;
  - $\mathbf{v} \in span(\mathbf{E})$: notice that in this case, by rule (ii), π_2 is also set. Therefore, since in this branch $\mathtt{M}, \pi_1, \pi_2$ are set, the rules (*) are not executed.
  - $\mathbf{v} \notin span(\mathbf{E})$: in this case we enter rule (iv) and thus decryption stops and outputs $\perp$. To see why this rule is entered, notice that either $\mathbf{u}^*$ is unset, or, if it is set, then $\mathbf{u}^* \notin span(\mathbf{D})$, and so $\mathbf{x} - \mathbf{x}^* \notin span(\mathbf{D}^*)$.

– $\mathbf{u} \notin span(\mathbf{D})$, in this case the output could be either $\diamond$ or $\perp$, more in details:

[9] The adversary computes π_2 in step 3a as the original decryption procedure would do, but by the modification in $\mathbf{H}_1$ we are assured that this is equivalent.

$\underline{\mathsf{KGen}_2(\mathtt{prm})}$:
$(\mathsf{pk}', \mathsf{sk}') \leftarrow\!\!\$\ \mathsf{KGen}'(\mathtt{prm})$, $\mathsf{crs} \leftarrow \mathsf{Init}(\mathtt{prm})$;
Parse $\mathsf{sk}' = (\mathbf{a}, \mathbf{f}, \mathbf{F}, \mathbf{g}, \mathbf{G})$;
Set $\mathsf{sk} = (\mathbf{a}, \mathsf{crs})$, $\mathsf{pk} = (\mathsf{pk}', \mathsf{crs})$;
Output $(\mathsf{pk}, \mathsf{sk})$.

$\underline{\mathsf{Rand}_2(\mathsf{pk}, \mathtt{C})}$:
Parse $\mathtt{C} = ([\mathbf{x}]_1, [\mathbf{v}]_2, \Pi)$,
$T \leftarrow\!\!\$\ \mathcal{T}$, (with associated $\hat{\mathbf{r}}, \hat{\mathbf{s}} \in \mathbb{Z}_q^k$)
$\hat{\mathbf{x}} = \mathbf{x} + \mathbf{D}^* \cdot \hat{\mathbf{r}}$;
$\hat{\mathbf{v}} = \mathbf{v} + \mathbf{E} \cdot \hat{\mathbf{s}}$;
$\hat{\Pi} = \mathsf{ZKEval}(\mathsf{crs}, T, ([\mathbf{x}]_1, [\mathbf{v}]_2), \Pi)$;
Output $([\hat{\mathbf{x}}]_1, [\hat{\mathbf{v}}]_2, \hat{\Pi})$.

$\underline{\mathsf{Enc}_2(\mathsf{pk}, [\mathtt{M}]_1)}$:
$\mathbf{r}, \mathbf{s} \leftarrow\!\!\$\ \mathbb{Z}_q^k$;
$([\mathbf{x}]_1, [\mathbf{v}]_2, [\pi]_T) \leftarrow \mathsf{Enc}'(\mathsf{pk}, [\mathtt{M}]_1; \mathbf{r}, \mathbf{s})$;
$\Pi \leftarrow\!\!\$\ \mathsf{P}(\mathsf{crs}, ([\mathbf{x}]_1, [\mathbf{v}]_2), ([\pi]_T, \mathbf{r}, \mathbf{s}))$;
Output $\mathtt{C} = ([\mathbf{x}]_1, [\mathbf{v}]_2, \Pi)$.

$\underline{\mathsf{Dec}_2(\mathsf{sk}, \mathtt{C})}$:
Parse $\mathtt{C} = ([\mathbf{x}]_1, [\mathbf{v}]_2, \Pi)$;
if $\mathsf{V}(\mathsf{crs}, ([\mathbf{x}]_1, [\mathbf{v}]_2), \Pi) = 1$
  output $(-\mathbf{a}^\top, 1) \cdot [\mathbf{x}]_1$;
else output $\perp$.

$\underline{\mathsf{Ver}(\mathsf{pk}, \mathtt{C})}$:
Parse $\mathtt{C} = ([\mathbf{x}]_1, [\mathbf{v}]_2, \Pi)$;
Output $\mathsf{V}(\mathsf{crs}, ([\mathbf{x}]_1, [\mathbf{v}]_2), \Pi)$.

Fig. 5: Our publicly-verifiable re-randomizable RCCA encryption scheme $\mathcal{PKE}_2$. The NIZK is for the relation $\mathcal{R}_{\mathcal{PKE}_1}$ and transformation $\mathcal{T}_{\mathcal{PKE}_1}$.

- $\mathbf{v}^*$ is unset: by rule (iii) decryption stops and outputs $\perp$.
- $\mathbf{v}^*$ is set and $(\mathbf{v} - \mathbf{v}^*) \notin span(\mathbf{E})$: by rule (iii) decryption outputs $\perp$.
- $\mathbf{v}^*$ is set and $(\mathbf{v} - \mathbf{v}^*) \in span(\mathbf{E})$:
  - $(\mathbf{x} - \mathbf{x}^*) \notin span(\mathbf{D}^*)$: notice that since $\mathbf{v}^* \notin span(\mathbf{E})$ then it must be that $\mathbf{v} \notin span(\mathbf{E})$. Hence, rule (iv) is entered and decryption outputs $\perp$.
  - $(\mathbf{x} - \mathbf{x}^*) \in span(\mathbf{D}^*)$: rule (v) is entered, decryption outputs $\diamond$, so $\mathtt{M}, \pi_1, \pi_2$ are set, and thus the rules (*) are not executed.

4 Our Publicly-Verifiable Rand-RCCA PKE

Here we show that our RCCA scheme from the previous section can be turned into a publicly verifiable one. Very informally, the idea is to append a malleable proof (essentially a GS proof) that $[\pi]_T$ is well formed. The decryption procedure of the publicly verifiable scheme can simply check the validity of the proof and then CPA-decrypt the ciphertext $[\mathbf{x}]_1$. Let $\mathcal{PKE}_1 = (\mathsf{KGen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1, \mathsf{Rand}_1)$ be the scheme of Sec. 3 and let $\mathcal{NIZK} = (\mathsf{Init}, \mathsf{P}, \mathsf{V}, \mathsf{ZKEval})$ be a malleable NIZK system for membership in the relation defined below:

$$\mathcal{R}_{\mathcal{PKE}_1} = \left\{ ([\mathbf{x}]_1, [\mathbf{v}]_2), ([\pi]_T, \mathbf{r}, \mathbf{s}) : [\pi]_T = [(\mathbf{f} + \mathbf{F}\mathbf{v})^\top \mathbf{u} + (\mathbf{g} + \mathbf{G}\mathbf{x})^\top \mathbf{v}]_T \right\},$$

and with allowable set of transformations:

$$\mathcal{T}_{\mathcal{PKE}_1} = \left\{ T : \begin{array}{c} \exists \hat{\mathbf{r}}, \hat{\mathbf{s}} \in \mathbb{Z}_q^k : \\ \\ \end{array} \begin{array}{l} T_x([\mathbf{x}]_1, [\mathbf{v}]_2) = ([\hat{\mathbf{x}}]_1, [\hat{\mathbf{v}}]_2) \\ T_w([\pi]_T, \mathbf{r}, \mathbf{s}) = ([\hat{\pi}]_T, \mathbf{r} + \hat{\mathbf{r}}, \mathbf{s} + \hat{\mathbf{s}}) \\ ([\hat{\mathbf{x}}]_1, [\hat{\mathbf{v}}]_2, [\hat{\pi}]_T) = \mathsf{Rand}_1(\mathsf{pk}, ([\mathbf{x}]_1, [\mathbf{v}]_2, [\pi]_T); \hat{\mathbf{r}}, \hat{\mathbf{s}}) \end{array} \right\}.$$

We write $T \leftarrow\!\!\$\ \mathcal{T}_{\mathcal{PKE}_1}$ for the operation that samples the uniquely defined $\hat{\mathbf{r}}, \hat{\mathbf{s}}$ associated to the transformation T. The pv-Rand-PKE scheme $\mathcal{PKE}_2 = (\mathsf{Init}, \mathsf{KGen}_2, \mathsf{Enc}_2, \mathsf{Dec}_2, \mathsf{Rand}_2, \mathsf{Ver})$ is described in Fig. 5. We defer the proof of the following theorem in Appendix B.

Theorem 2. *If the $\mathcal{NIZK}$ is adaptive sound and perfect derivation private then the pv-Rand-PKE scheme $\mathcal{PKE}_2$ described in Fig. 5 is publicly verifiable, perfect re-randomizable and RCCA-secure.*

Malleable NIZK. The equations we would like to prove do not admit Groth-Sahai NIZK proofs [29], but only NIWI. We overcome this problem by developing a new technique that

$\underline{\mathbf{Exp}^{\mathtt{tse\text{-}cm}}_{\mathsf{A},\mathsf{Ext},\mathcal{NIZK}}}$:
$\mathtt{prm}_G \leftarrow\!\!\$\ \mathsf{Setup}_G(1^\lambda)$; Set $\mathcal{Q}_w \leftarrow \emptyset$;
$(\mathsf{crs}, tp_e, tp_s) \leftarrow \mathsf{Init}(\mathtt{prm}_G)$;
$(x, \pi) \leftarrow \mathsf{A}(\mathsf{crs}, \mathcal{R})^{\mathcal{SIM}()}$; $z \leftarrow \mathsf{Ext}(tp_e, x, \pi, \mathcal{R})$;
Output 1 if $\mathsf{V}(\mathsf{crs}, x, \pi) = 1$ and either:
(a) $z \neq \circ$ and $\forall w$ s.t. $z = f(w)$ we have $(x, w) \notin \mathcal{R}$ or
(b) $z = \circ$ and $\forall x' \in \mathcal{Q}_x, \forall T \in \mathcal{T}$ we have $T_x(x) \neq x$.

$\underline{\mathcal{SIM}(x, w)}$:
if $(x, w) \in \mathcal{R}$ then
$\pi \leftarrow \mathsf{Sim}(tp_s, x)$;
$\mathcal{Q}_x \leftarrow \mathcal{Q}_x \cup \{x\}$;

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

extends the class of pairing product equations which admit GS NIZK proofs. This technique is *per se* a result of independent interest.

More in detail, we produce an additional commitment to $[\pi]_T$, using a new commitment type defined over $\mathbb{G}_T$ with good bilinear properties. This allows us to construct a NIZK proof that the ciphertext is valid with perfect completeness and soundness and composable zero-knowledge. The latter notion refers to the fact that if the common reference string is defined in a "witness indistinguishable mode", the proof system is perfect zero-knowledge. By replacing $[\pi_T]$ in the ciphertext by its commitment, in the witness indistinguishable mode we can simulate a proof of validity of the ciphertext by setting $\pi = 0$ and in an undetectable manner. The proof will be correctly distributed because of the perfect zero-knowledge property in these modes.

All the details on how to compute the proof are given in Appendix B.1. Beyond GS Proofs, it also makes use of the QANIZK proof of membership in linear spaces [32,33,36]. The size of the ciphertexts for the SXDH instantiation of the publicly verifiable scheme is $12|\mathbb{G}_1| + 11|\mathbb{G}_2| + 4|\mathbb{G}_T|$. The number of pairings for verification is 32 for the GS proof and 14 for the argument of linear spaces, which can be reduced to $8 + 14$ by batch verifying the GS equation using the techniques of [31].

5 Malleable and True-Simulation Extractable NIZK

In this section we show an application of our Rand-RCCA scheme to build a malleable and true-simulation extractable NIZK. We start by recalling the notion of true-simulation extractability and then move to describing our construction.

True-Simulation Extractability. We recall the notion of true-simulation f-extractability (f-tSE-NIZK, for short) of Dodis *et al.*[13]. The notion is a weakening of the concept of simulation extractability where the extractor can compute a function of the witness and the adversary sees simulated proofs only for true statements. Here, we give a variation of the notion that allows for re-randomizability (and malleability). Consider the experiment described in Fig. 6, the main difference respect to the notion of [13], is that the winning condition (b) allows the extractor to give up and output a special symbol $\circ$. The restriction is that the extractor can safely do this without losing the game only when the proof π produced by the adversary is derived from a simulated proof.

Definition 8. *Let f be an efficiently computable function, let $\mathcal{NIZK} = (\mathsf{Init}, \mathsf{P}, \mathsf{V})$ be a NIZK argument for a relation $\mathcal{R}$, and consider the experiment $\mathbf{Exp}^{\mathtt{tse\text{-}cm}}$ described in Fig. 6. We say that $\mathcal{NIZK}$ is* true-simulation controlled-malleable f-extractable *(f-tSE-cm) iff there exists a PPT algorithm* Ext *such that for all PPT* A *we have that*

$$\mathbf{Adv}^{\mathtt{tse-cm}}_{\mathsf{A},\mathsf{Ext},\mathcal{NIZK}}(\lambda) := \Pr\left[\mathbf{Exp}^{\mathtt{tse\text{-}cm}}_{\mathsf{A},\mathsf{Ext},\mathcal{NIZK}}(1^\lambda) = 1\right] \in \mathtt{negl}(\lambda).$$

Construction. The construction follows the blueprint of Dodis *et al.* [13] with the twist that we use a Rand RCCA-PKE scheme instead of a CCA-PKE scheme. Our compiler works for a

$\underline{\mathsf{Init}(\mathtt{prm})}$:
$(\mathsf{crs}', tp'_s) \leftarrow \mathsf{Init}'(\mathtt{prm})$;
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\mathtt{prm})$;
$\mathsf{crs} \leftarrow (\mathsf{crs}', \mathsf{pk})$, $tp_e \leftarrow \mathsf{sk}$,$tp_s \leftarrow (\mathsf{pk}, tp'_s)$
Output $(\mathsf{crs}, tp_e, tp_s)$.

$\underline{\mathsf{P}(\mathsf{crs}, x, w)}$:
$\mathtt{C} \leftarrow \mathsf{Enc}(\mathsf{pk}, f(w); r)$;
$\pi' \leftarrow \mathsf{P}'(\mathsf{crs}', (\mathsf{pk}, \mathtt{C}, x), (w, r))$;
Output $\pi = (\mathtt{C}, \pi')$.

$\underline{\mathsf{V}(\mathsf{crs}, x, \pi)}$:
Output $\mathsf{V}'(\mathsf{crs}', (\mathsf{pk}, \mathtt{C}, x), \pi')$

$\underline{\mathsf{ZKEval}(\mathsf{crs}, T, (x, \pi))}$:
Let $\pi = (\mathtt{C}, \pi')$, $\rho \leftarrow_\$ \mathbb{Z}_q^\ell$;
Let $T' = (\rho, T)$;
$\hat{\mathtt{C}} \leftarrow \mathsf{Rand}(\mathsf{pk}, \mathtt{C}; \rho)$;
$\hat{\pi}' \leftarrow_\$ \mathsf{ZKEval}'(\mathsf{crs}', T', (x, \pi'))$;
Output $(\hat{\mathtt{C}}, \hat{\pi}')$.

Fig. 7: Our f-tSE-cm $\mathcal{NIZK}$ compiler.

special class of tuples, consisting of a function f, an NP relation $\mathcal{R}$ and a transformation $\mathcal{T}$, that we define below:

Definition 9. *A tuple $(f, \mathcal{R}, \mathcal{T})$, where f is efficiently computable, $\mathcal{R}$ is an NP-relation and $\mathcal{T}$ is an admissible transformation for $\mathcal{R}$, is* suitable *if:*

1. *there exists an efficiently computable decision procedure g such that for any (x, w) the function $g(x, f(w)) = 1$ if and only if $(x, w) \in \mathcal{R}$;*
2. *For any $T \in \mathcal{T}$ and any $(x, w) \in \mathcal{R}$ the transformation of the witness is invariant respect to the function f, namely $f(w) = f(T_w(w))$.*

The restrictions above still allow for many interesting malleabilities. For example, the condition (2) clearly applies to re-randomizable NIZKs, as in this case $T_w(\cdot)$ is the identity function. Condition (1) holds in all those cases where the relation $\mathcal{R}$ can be sampled together with a trapdoor information that allows to compute w from x. The condition (1) applies also to the NIZKs of [13]. More importantly, the conjunction of (1) and (2) allows to efficiently check the condition (b) of the security experiment, which makes the tSE-cm NIZK primitive easier to use.

Let $\mathcal{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Rand})$ be a Rand-RCCA PKE scheme, we additionally assume there exists an integer $\ell \in \mathbb{N}$ such that the random coins of both the encryption procedure and the re-randomization procedure are in $\mathbb{Z}_q^\ell$ and that, for any $\mathsf{pk}, \mathtt{M}$, given $\mathsf{Rand}(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, \mathtt{M}; \rho_0); \rho_1) = \mathsf{Enc}(\mathsf{pk}, \mathtt{M}; \rho_0 + \rho_1)$ where $\rho_0, \rho_1 \in \mathbb{Z}_q^\ell$. Notice that the schemes in Sec. 3 and Sec 4 have this property. Let $\mathcal{R}$ be a NP relation and $\mathcal{T}$ be a set of allowable transformations for the relation $\mathcal{R}$. Let $\mathcal{NIZK}' = (\mathsf{Init}', \mathsf{P}', \mathsf{V}', \mathsf{ZKEval}')$ be a malleable NIZK argument for $\mathcal{R}'$ with the allowable set of transformations $\mathcal{T}'$ as described below:

$$\mathcal{R}' = \{((\mathsf{pk}, c, x), (w, \rho)) : \ (x, w) \in \mathcal{R} \ \wedge \ c = \mathsf{Enc}(\mathsf{pk}, f(w); \rho)\}$$

$$\mathcal{T}' = \left\{ T' : \ \exists \hat{\rho}, T : \begin{array}{c} T'_x(\mathsf{pk}, c, x) = (\mathsf{pk}, \mathsf{Rand}(\mathsf{pk}, c; \hat{\rho}), T_x(x)), \\ T'_w(w, \rho) = (T_w(w), \rho + \hat{\rho}), \quad T \in \mathcal{T} \end{array} \right\}$$

We also assume that any transformation $T' \in \mathcal{T}'$ can be efficiently parsed as a tuple $(\hat{\rho}, T)$ and viceversa. We define a malleable NIZK argument $\mathcal{NIZK} = (\mathsf{Init}, \mathsf{P}, \mathsf{V}, \mathsf{ZKEval})$ for the relation $\mathcal{R}$ with allowable set of transformations $\mathcal{T}$ in Fig 7. Notice that the co-domain of the function f for which we can prove f-tSE soundness is the message space of the underlying Rand-RCCA PKE scheme. We remark that, although our scheme is presented with a message space $\mathcal{M} = \mathbb{G}_1$, we could easily extend our construction to encrypt vectors in $\mathbb{G}_1^{\ell_0} \times \mathbb{G}_2^{\ell_1}$.

Theorem 3. *For any suitable $(f, \mathcal{R}, \mathcal{T})$ the proof system $\mathcal{NIZK}$ is a malleable NIZK for $\mathcal{R}$ with allowable transformations $\mathcal{T}$, and if $\mathcal{NIZK}'$ is perfectly (resp. statistically) derivation private (Def. 7) and $\mathcal{PKE}$ is perfectly re-randomizable (Def. 3) then $\mathcal{NIZK}$ is perfectly (resp. statistically) derivation private.*

Proof. First we notice that ZKEval is complete. In fact, $(f, \mathcal{R}, \mathcal{T})$ is suitable we have that for any $T \in \mathcal{T}$ the transformation $T_w(\cdot)$ is invariant respect to the f and therefore the transformed statement-witness is in the relation $\mathcal{R}'$.

Recall that the adversary A for the derivation privacy of $\mathcal{NIZK}$ outputs (x, w, π, T) where $\pi = (\mathtt{C}, \pi')$ is a valid proof of x and T is a allowable transformation for $\mathcal{T}$

We first notice that we can assume that exists $\mathtt{M}, r$ such that $\mathtt{C} \in \mathsf{Enc}(\mathsf{pk}, \mathtt{M}; r)$ by condition 3 of perfect re-randomizability, therefore we can compute, although inefficiently, from $\mathtt{C}$ the randomness r.

Consider an hybrid experiment $\mathbf{H}_1$ where once received (x, w, π, T) from the adversary A we compute r from $\mathtt{C}$ and we compute a new proof $\pi' \leftarrow \mathsf{P}'(\mathsf{crs}', (\mathsf{pk}, \hat{\mathtt{C}}, x), (w, r + \hat{r}))$ where $\hat{\mathtt{C}} = \mathsf{Rand}(\mathsf{pk}, \mathtt{C}; \hat{r})$.

It is easy to see that $\mathbf{Exp}^{\mathtt{der\text{-}priv}}_{\mathsf{A}, \mathcal{NIZK}}$ conditioned on the challenge bit equal to 1 (namely, when ZKEval is used) and $\mathbf{H}_1$ are statiscally close by the statistical derivation privacy of $\mathcal{NIZK}'$. the reduction B would simply emulate the hybrid outputing $((\mathsf{pk}, \hat{\mathtt{C}}, x), (w, r), T' = (T, \hat{r})$.

Moreover, we can prove that $\mathbf{H}_1$ is distributed equivalently to $\mathbf{Exp}^{\mathtt{der\text{-}priv}}_{\mathsf{A}, \mathcal{NIZK}}$ conditioned on the challenge bit equal to 0. In fact, the only difference between the two distribution is that in one case the ciphertext is fresh while in the other case is a re-randomization.

Theorem 4. *For any suitable $(f, \mathcal{R}, \mathcal{T})$ the proof system $\mathcal{NIZK}$ described above is true-simulation controlled-malleable f-extractable.*

We give an intuition for the proof of Theorem 4, which proceeds with a two-steps hybrid argument. We start with the true-simulation extractability experiment, we can switch to an experiment where each simulated proof for $\mathcal{NIZK}$ contains an encryption of the $f(w)$. This step can be easily argued using the RCCA security of the scheme. In particular, the guarded decryption oracle and the suitability of $(f, \mathcal{R}, \mathcal{T})$ are necessary to check the winning condition of the tSE experiment. In the second step, we switch to valid proofs for $\mathcal{NIZK}'$, instead of simulated proofs, the indistinguishability follows trivially by the zero-knowledge of $\mathcal{NIZK}'$. At this point we are in an experiment where the proofs provided by the $\mathcal{SIM}$ are not simulated, so the standard adaptive soundness of $\mathcal{NIZK}'$ is sufficient to bound the winning probability of the adversary.

Proof. We describe both a simulator and an extractor for the tse-cm-NIZK $\mathcal{NIZK}$.

- Let $\mathsf{Sim}(tp_s, x)$ be the simulator that parses $tp_s = (\mathsf{pk}, tp'_s)$ and computes $\mathtt{C} \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{pk}, \circ)$ and $\pi \leftarrow \mathsf{Sim}'(tp'_s, (\mathsf{pk}, \mathtt{C}, x))$ where Sim' is the simulator of $\mathcal{NIZK}'$.
- Let $\mathsf{Ext}(tp_e, x, \pi)$ be the extractor that parses tp_e ans sk and $\pi = (\mathtt{C}, \pi')$ and outputs $\mathsf{Dec}(\mathsf{sk}, \mathtt{C})$.

We consider a sequence of hybrid experiments.

- The first experiment $\mathbf{H}_0$ is the $\mathbf{Exp}^{\mathtt{tse\text{-}cm}}_{\mathsf{A}, \mathsf{Ext}, \mathcal{NIZK}}$, namely, the oracle $\mathcal{SIM}$ upon the i-th query (x_i, w_i) first checks that $(x_i, w_i) \in \mathcal{R}$ and if so it adds x_i in $\mathcal{Q}_x$ and w_i in $\mathcal{Q}_w$ and outputs $\mathtt{C}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, \circ)$ and $\pi'_i \leftarrow \mathsf{Sim}'(tp'_s, (\mathsf{pk}, \mathtt{C}_i, x_i))$.
- Let $\mathbf{H}_{1,j}$ be the same as $\mathbf{H}_0$ but where the first j ciphertexts are valid encryption of $f(w)$. Specifically, the oracle $\mathcal{SIM}$ upon the i-th query (x_i, w_i) if $i > j$ then it behaves as in $\mathbf{H}_0$ otherwise it first checks that $(x_i, w_i) \in \mathcal{R}$ and if so adds w_i in $\mathcal{Q}_w$ and outputs $\mathtt{C}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, f(w_i))$ and $\pi'_i \leftarrow \mathsf{Sim}'(tp'_s, (\mathsf{pk}, \mathtt{C}_i, x_i))$.
- Let $\mathbf{H}_2$ be the same as $\mathbf{H}_{1,q}$, where q is the number of queries made by A, but where the proofs for $\mathcal{NIZK}'$ are not simulated. Specifically, the oracle $\mathcal{SIM}$ upon the i-th query (x_i, w_i) first checks that $(x_i, w_i) \in \mathcal{R}$ and if so adds w_i in $\mathcal{Q}_w$ and outputs $\mathtt{C}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, f(w_i); r_i)$ where $r_i \leftarrow \{0,1\}^\lambda$ and $\pi'_i \leftarrow \mathsf{P}'(\mathsf{crs}', (\mathsf{pk}, \mathtt{C}_i, x_i), (w_i, r_i))$.

Lemma 5. *For any* $j \in \mathbb{N}$, $|\Pr[\mathbf{H}_{1,j} = 1] - \Pr[\mathbf{H}_{1,j+1} = 1]| \in \mathtt{negl}(\lambda)$.

Proof. We show a reduction to the RCCA security of the PKE-scheme. Consider an adversary B for the RCCA-security experiment. The adversary B upon input pk generates the parameter $\mathsf{crs}', tp'_s \leftarrow \mathsf{Init}(\mathtt{prm}_G)$ and runs $\mathsf{A}(\mathsf{crs})$ where $\mathsf{crs} = (\mathsf{pk}, \mathsf{crs}')$. At the i-th query (x_i, w_i) made by A:

- if $i < j$, the adversary B returns to the adversary A the values $\mathtt{C}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, f(w_i))$ and $\pi'_i \leftarrow \mathsf{Sim}'(tp'_s, (\mathsf{pk}, \mathtt{C}_i, x_i))$;
- if $i = j$ the adversary B sends the challenge messages $(\circ, f(w_j))$ to its own challenger and receives $\mathtt{C}^*$, it returns to A the values $\mathtt{C}^*$ and $\pi'_j \leftarrow \mathsf{Sim}'(tp'_s, (\mathsf{pk}, \mathtt{C}^*, x_i))$;
- If $i > j$, the adversary B returns to the adversary A the values $\mathtt{C}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, \circ)$ and $\pi'_i \leftarrow \mathsf{Sim}'(tp'_s, (\mathsf{pk}, \mathtt{C}_i, x_i))$;

Eventually, the adversary A outputs a tuple x, π where $\pi = (\mathtt{C}, \pi')$, the adversary B forwards $\mathtt{C}$ to its own decryption oracle, let z be the answer from the decryption oracle. First the adversary B checks that the proof π' verifies and if not output 0. Secondly, the adversary B if $z \neq \circ$ then output 1 if and only if $g(x, z, \omega) = 0$, else it outputs 1 if and only if for any $w \in \mathcal{Q}_w$ we have $g(x, f(w), \omega) = 0$.

We notice that B runs in polynomial time in λ. We check that B perfectly simulates the $\mathbf{H}_{1,i+b}$ experiment when the challenge bit of the RCCA experiment is equal to b. It is easy to check that the adversary B perfectly simulates the hybrid experiments until the adversary A outputs x, π. The hybrids output 1 iff the π' verifies and either $z \neq \circ$ and for all w such that $f(w) = z$ we have that $(x, w) \notin \mathcal{R}$ (which can be efficiently computable by condition (1) of Def. 9) or $z = \circ$ and for all w such that $w \in \mathcal{Q}_w$ we have $(x, w) \notin \mathcal{R}$ (again, it can be efficiently computable). Notice that if for any (x_i, w_i) queried to $\mathcal{SIM}$ if $g(x, T_w(w_i), \omega) = 0$, then for any $T \in \mathcal{T}$, $T_x(x_i) \neq x$. In fact, $1 = g(T_x(x_i), f(T_w(w_i)), \omega) = g(T(x_i), f(w_i), \omega) \neq g(x, f(w_i), \omega)$, and thus $T(x_i) \neq x$. This concludes the proof of the lemma.

Instantiation. For any suitable $(f, \mathcal{R}, \mathcal{T})$ where the co-domain of f is $\mathbb{G}_1$, we can instantiate the tSE-cm NIZK scheme with the pv-Rand-RCCA Scheme $\mathcal{PKE}_2$. The public verifiability enables for a simpler malleable NIZK proof for the associated $\mathcal{R}'$. In fact, we can subdivide the proof in: (1) a malleable GS proof Π_1 for $\mathcal{R}$ with transformations $\mathcal{T}$, in particular Π_1 contains GS commitments $[\mathbf{c}_w]_1$ of the witness; (2) a malleable GS proof Π_2 to prove that commitments $[\mathbf{c}_w]_1$ and $[\mathbf{c}_{w'}]_1$ open to w, w' an $w' = f(w)$; (3) a malleable proof Π_3 to prove $w' = (-\mathbf{a}^T, 1) \cdot [\mathbf{x}]$, in particular, from the linearity of GS commitments the relation for the last proof is a linear subspace relationship. The verification checks the proofs Π_1, Π_2, Π_3 and verifies the validity of the ciphertext $\mathtt{C}$.

For the case where f is the identity function, namely, re-randomizable NIZK, the proof Π_2 is trivial as we can set $[\mathbf{c}_w]_1 = [\mathbf{c}_{w'}]_1$. The overhead in proof size between a adaptive sound re-randomizable GS proof for $\mathcal{R}$ based on SXDH and an tSE-cm NIZK based on SXDH is equal to $13|\mathbb{G}_1| + 11|\mathbb{G}_2| + 4|\mathbb{G}_T|$.

6 An UC-Secure Mix-Net

In this section we propose an application of pv-Rand-PKE schemes with RCCA security to Mix-Net protocols. Our starting point is a recent work of Faonio and Fiore [**?**] who build an UC-secure Optimistic Mix-Net using a new paradigm that relies on a specific re-randomizable and RCCA-secure PKE scheme. Here we extend the main idea of [**?**] and use the power of public verifiability in order to obtain a full fledged Mix-Net protocol (not only optimistic secure). As mentioned in

the introduction, our new protocol enjoys an interesting set of features that compare well with state of the art solutions.

The Universal Composability model. We review some basic notions of the Universal Composability model and an extension to *auditable protocols*. In a nutshell, a protocol Π UC-realizes an ideal functionality $\mathcal{F}$ with setup assumption $\mathcal{G}$ if there exists a PPT simulator S such that no PPT environment $\mathcal{Z}$ can distinguish an execution of the protocols Π which can interact with the setup assumption $\mathcal{G}$ from a joint execution of the simulator S with the ideal functionality $\mathcal{F}$. The environment $\mathcal{Z}$ provides the inputs to all the parties of the protocols, decides which party to corrupt (we consider static corruption, where the environment decides the corrupted parties before the protocol starts), and schedules the order of the messages in the networks. When specifying an ideal functionality, we use the "delayed outputs" terminology of Canetti [4]. Namely, when a functionality $\mathcal{F}$ *sends a public delayed output* M *to party* $\mathcal{P}_{P_i}$ we mean that M is first sent to the simulator and then forwarded to $\mathcal{P}_{P_i}$ only after acknowledgement by the simulator. We consider a variation of the UC model where, roughly speaking, a bulletin board functionality **BB** acts as global setup assumption. In more detail, the bulletin board is present in both the ideal world and the real world, so that the simulator does not have any advantage over the real-world adversary and all the parties of the protocol can register their message on the board. An *auditable protocol* is a tuple (Π, Audit) where Π is a protocol and Audit is a PPT algorithm. The model additionally includes an external off-line party, the auditor. The auditor is an incorruptible party which, whenever is called on an input y', runs the audit algorithm Audit on this input and the transcript written in the bulletin boards and forwards its output to the environment. In the ideal world, the auditor always replies according to the output of the ideal functionality, for example, if the ideal functionality has output y and the auditor is called on input y', the auditor replies with `valid` if and only if $y = y'$. We elaborate on the notion of auditable protocols in Appendix **??**. In particular, we give a definition of auditable protocols, formalize the notion of UC composition for auditable protocols and prove a composition theorem.

Defining Mix-Net Protocols. Our protocol UC-realizes the ideal functionality $\mathcal{F}_{\mathsf{Mix}}$ described in Fig. 8 with setup assumptions: the ideal functionality $\mathcal{F}_{\mathsf{TDec}}$ for threshold decryption of our PKE scheme and the ideal functionality for a common-reference string $\mathcal{F}_{\mathsf{CRS}}$ (and the bulletin board of the auditable framework of Faonio and Fiore). The functionality $\mathcal{F}_{\mathsf{Mix}}$ (similarly to [**?**]) is slightly weaker than the one considered by Wikström in [48,49]. The difference is that the corrupted senders can replace their inputs, however, they loose this ability when the first honest mixer sends its message `mix`. On the other hand, in the ideal functionality of Wikström, the senders can cast their messages only during the inputs submission phase.

Functionality $\mathcal{F}_{\mathsf{Mix}}$:

The functionality has n sender parties $\mathcal{P}_{S_i}$ and m mixer parties $\mathcal{P}_{M_i}$:

Input: On message $(\mathtt{input}, \mathtt{M}_i)$ from $\mathcal{P}_{S_i}$ (or the adversary if $\mathcal{P}_{S_i}$ is corrupted) register the index i in the list of the senders and register the entry $(i, \mathtt{M}_i)$ in the database of the inputs. Notify the adversary that the sender $\mathcal{P}_{S_i}$ has sent its input.

Mix: On message `mix` from $\mathcal{P}_{M_i}$ (or the adversary if $\mathcal{P}_{M_i}$ is corrupted), register the index i in the list of the mixers and notify the adversary.

Delivery: If all the senders are in the list of the senders and at least one honest mixer is in the list of the mixers send a public delayed output $\mathcal{O} \leftarrow \mathsf{Sort}(\langle \mathtt{M}_j \rangle_{j \in [n]})$ to all the mixers.

Fig. 8: Ideal Functionality for Mixing.

Building blocks. The main building blocks of our mix-net construction are:

(i) An *linear* pv-Rand-RCCA PKE scheme $\mathcal{PKE}$. 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 $(\mathsf{pk}, \mathsf{sk})$ we can parse $\mathsf{pk} = ([\mathbf{P}], \hat{\mathsf{pk}})$ and $\mathsf{sk} = (\mathbf{S}, \hat{\mathsf{sk}})$, where $[\mathbf{P}] \in \mathbb{G}^{\ell \times \ell''}$ and $\mathbf{S} \in \mathbb{Z}_q^{\ell' \times \ell}$, (2) any ciphertext $\mathtt{C} \in \mathcal{C}$ can be parsed as $([\mathbf{y}], \hat{\mathtt{C}})$ where $[\mathbf{y}] \in \mathbb{G}^{\ell}$, (3) for any ciphertext $\mathtt{C}$ such that $\mathsf{Ver}(\mathsf{pk}, \mathtt{C}) = 1$ the decryption procedure is linear, i.e., we have $\mathsf{Dec}(\mathsf{sk}, \mathtt{C}) = \mathbf{S} \cdot [\mathbf{y}]$ (4) let $\mathtt{C}' = \mathsf{Rand}(\mathsf{pk}, \mathtt{C}; \mathbf{r}, r)$ where $\mathtt{C}' = ([\mathbf{y}'], \hat{\mathtt{C}}')$ be a re-randomization of $\mathtt{C} = ([\mathbf{y}], \hat{\mathtt{C}})$ and $\mathbf{r} \in \mathbb{Z}_q^{\ell''}$ then $([\mathbf{y}] - [\mathbf{y}']) = [\mathbf{P}]\mathbf{r}$. We notice that both the scheme $\mathcal{PKE}_2$ in Sec. 4 and the pv-Rand-RCCA PKE scheme of [37,8] are linear. Indeed, our abstraction is made to include the three schemes under the same template.

(ii) An All-but-One label-based NIZK. An ABO label-based $\mathcal{NIZK}_{\mathtt{sd}} = (\mathsf{Init}_{\mathtt{sd}}, \mathsf{P}_{\mathtt{sd}}, \mathsf{V}_{\mathtt{sd}})$ for knowledge of the plaintext of the linear PKE. In more detail a ABO label-based $\mathcal{NIZK}$ is a NIZK system with labels where there exists an algorithm $\mathsf{ABOInit}(\mathtt{prm}, \tau)$ which creates a common reference string crs together with a trapdoor tp_s such that for any label $\tau' \neq \tau$ the trapdoor allows for zero-knowledge while for τ the proof system is adaptive sound. A ABO label-based $\mathcal{NIZK}$ in the random-string model can be easily obtained from GS NIZK proof system (c.f. Appendix **??**).

(iii) An adaptive sound NIZK. $\mathcal{NIZK}_{\mathtt{mx}} = (\mathsf{Init}_{\mathtt{mx}}, \mathsf{P}_{\mathtt{mx}}, \mathsf{V}_{\mathtt{mx}})$ for proving membership in the relation $\mathcal{R}_{\mathtt{mx}} = \{([\mathbf{P}], [\mathbf{y}]) : [\mathbf{y}] \in span([\mathbf{P}])\}$. We recall that GS proof system is in the random-string model.

(iv) An ideal functionality $\mathcal{F}_{\mathsf{TDec}}$ for threshold decryption of the pv-Rand-RCCA PKE $\mathcal{PKE}$ scheme. In more detail, $\mathcal{F}_{\mathsf{TDec}}$ (formally defined in Appendix F, Fig 11) takes as parameters the definition of the PKE scheme and group parameters $\mathtt{prm}$ for the key generation. The functionality initializes a fresh key pair and accepts input of the form $(\mathtt{dec}, \mathtt{C})$ from the mixers: when a mixer sends a message of this kind, we say that the mixer *asks for the decryption of* $\mathtt{C}$. When all all the mixers have sent a message of the form $(\mathtt{dec}, \mathtt{C})$ the functionality sends a public delayed output $\mathsf{Dec}(\mathsf{sk}, \mathtt{C})$: in this case we say that the mixers *agreed on the decryption of* $\mathtt{C}$. In Appendix F we show a protocol for the functionality $\mathcal{F}_{\mathsf{TDec}}$ in the $\mathcal{F}_{\mathsf{CRS}}$-hybrid world.

(v) An ideal functionality for the common reference string of the above NIZKs. The functionality initializes m different CRS $\{\mathsf{crs}^i_{\mathtt{mx}}\}_{i=1,\ldots,m}$, one for each mixer,[10] for $\mathcal{NIZK}_{\mathtt{mx}}$ and a CRS $\mathsf{crs}_{\mathtt{sd}}$ for $\mathcal{NIZK}_{\mathtt{sd}}$. We stress that all the CRSs can be sampled as uniformly random strings in the real protocol.

Also we recall that our auditable protocol uses a Bulletin Board functionality. We do not mention it as a "building block" because every auditable protocol, as defined by [**?**], necessarily needs a bulletin board as setup assumption.

Our Mix-Net Protocol. Following the design rationale of Faonio and Fiore, given two lists of ciphertexts $\mathcal{L} = \langle \mathtt{C}_1, \ldots, \mathtt{C}_n \rangle$ and $\mathcal{L}' = \langle \mathtt{C}'_1, \ldots, \mathtt{C}'_n \rangle$, we define the *checksum* of these lists as the output of the following procedure:

Procedure $\mathbf{CkSum}(\mathcal{L}, \mathcal{L}')$:

1. For all $j \in [n]$ parse $\mathtt{C}_j = ([\mathbf{y}_j], \hat{\mathtt{C}}_j)$ and $\mathtt{C}'_j = ([\mathbf{y}'_j], \hat{\mathtt{C}}'_j)$;
2. Output $\sum_j [\mathbf{y}_j] - [\mathbf{y}'_j]$.

We describe our mix-net protocol Π between n sender parties $\mathcal{P}_{S_i}$ and m mixer parties $\mathcal{P}_{M_i}$ and with resources the ideal functionalities $\mathcal{F}_{\mathsf{TDec}}$ and $\mathcal{F}_{\mathsf{CRS}}$:

[10] We could modify our protocol to let the mixers share the same CRS, at the price of requiring $\mathcal{NIZK}_{\mathtt{mx}}$ be simulation sound. Since in most applications the number of mixers is small, we go for the simpler option of one crs per mixer.

Inputs Submission. Every sender $\mathcal{P}_{S_j}$, with $j \in [n]$, encrypts its message $\mathtt{M}_j$ by computing $\mathtt{C}_j \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathtt{M}_j; r)$, and creates a NIZK proof of knowledge $\pi_j^{\mathtt{sd}} \leftarrow \mathsf{P}_{\mathtt{sd}}(\mathsf{crs}_{\mathtt{sd}}, j, (\mathsf{pk}, \mathtt{C}), (\mathtt{M}_j, r))$ (the label for the proof is j). The party $\mathcal{P}_{S_j}$ posts $(\mathtt{C}_j, \pi_j^{\mathtt{sd}})$ on the bulletin board.

Mix. Once all the senders are done with the previous phase, let $\mathcal{L}_0 = \langle \mathtt{C}_{0,j} \rangle_{j \in [n]}$ be the list of ciphertexts they posted on the bulletin board. To simplify the exposition of the result, we assume that all the NIZK proofs $\{\pi_j^{\mathtt{sd}}\}_{j \in [n]}$ and all the ciphertexts in $\mathcal{L}_0$ verify.

For $i = 1$ to m, the mixer $\mathcal{P}_{M_i}$ waits for the $\mathcal{P}_{M_{i-1}}$ to complete and does:

1. Sample a permutation $\tau_i \leftarrow_{\$} \mathcal{S}_n$;
2. Read from the BB the message $(\mathcal{L}_{i-1}, \pi_{i-1}^{\mathtt{mx}})$ posted by $\mathcal{P}_{M_{i-1}}$ (or read $\mathcal{L}_0$ if this is the first mixer), and parse $\mathcal{L}_{i-1} = \langle \mathtt{C}_{i-1,j} \rangle_{j \in [n]}$;
3. Build the list $\mathcal{L}_i \leftarrow \langle \mathtt{C}_{i,j} \rangle_{j \in [n]}$ of shuffled and re-randomized ciphertexts by sampling randomness $\mathbf{r}_j, r_j$ and computing

$$\mathtt{C}_{i,\tau_i(j)} \leftarrow \mathsf{Rand}(\mathsf{pk}, \mathtt{C}_{i-1,j}; \mathbf{r}_j, r_j).$$

4. Compute a NIZK proof $\pi_i^{\mathtt{mx}} \leftarrow_{\$} \mathsf{P}_{\mathtt{mx}}(\mathsf{crs}_{\mathtt{mx}}^i, ([\mathbf{P}], \mathbf{CkSum}(\mathcal{L}_{i-1}, \mathcal{L}_i)), \sum_j \mathbf{r}_j)$,
5. Post in the BB the tuple $(\mathcal{L}_i, \pi_i^{\mathtt{mx}})$

Verification. Once all mixers are done, with the previous phase,every mixer $\mathcal{P}_{M_i}$ executes:

1. Read the messages $(\mathcal{L}_i, \pi_i^{\mathtt{mx}})$ posted by every mixer on the BB, as well as the messages $(\mathtt{C}_{0,j}, \pi_j^{\mathtt{sd}})$ posted by the senders;
2. For all $i \in [m]$ and for all $j \in [n]$ check that $\mathsf{Ver}(\mathsf{pk}, \mathtt{C}_{i,j}) = 1$;
3. For all $i \in [m]$, check $\mathsf{V}_{\mathtt{mx}}(\mathsf{crs}_{\mathtt{mx}}^i, ([\mathbf{P}], \mathbf{CkSum}(\mathcal{L}_{i-1}, \mathcal{L}_i)), \pi_i^{\mathtt{mx}}) = 1$;
4. If one of the checks does not verify abort and write `invalid` in the BB.

Decrypt. All the mixers $\mathcal{P}_{M_i}$ execute the following in parallel (using the ideal functionality $\mathcal{F}_{\mathsf{TDec}}$ to compute decryptions):

1. let $\mathcal{L}_m = \langle \mathtt{C}_j^* \rangle_{j \in [n]}$ be the list of ciphertexts returned by the last mixer. For $j = 1$ to n, ask $\mathcal{F}_{\mathsf{TDec}}$ for the decryption of $\mathtt{C}_j^*$. Once all the mixers agreed on the decryption, receive $\mathtt{M}_j \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathtt{C}_j^*)$ from the functionality;
2. Post $\mathsf{Sort}(\langle \mathtt{M}_j \rangle_{j \in [n]})$ on the BB.

Audit Message. The mixers $\mathcal{P}_{M_i}$ post the message `valid` on the BB.

Algorithm Audit: the algorithm reads from the BB and computes the verification step of the protocol above (notice that this only relies on public information). The algorithm outputs 1 either if the verification succeeds and `valid` is posted in the BB or if the verification fails and `invalid` is posted in the BB.

Theorem 5. *The auditable protocol* (Π, Audit) *described above* $\mathsf{UC\text{-}auditable}$ *realizes* $\mathcal{F}_{\mathsf{Mix}}$ *with setup assumptions* $\mathcal{F}_{\mathsf{TDec}}$ *and* $\mathcal{F}_{\mathsf{CRS}}$.

Proof. We show a simulator S and we argue that no PPT environment $\mathcal{Z}$ can distinguish an interaction with the real protocol (the real world) from an interaction with the simulator S and the ideal functionality $\mathcal{F}_{\mathsf{Mix}}$ (the ideal world).

To show the indistinguishability of the ideal and real worlds we give a sequence of hybrid experiments in which the real world is progressively modified until reaching an experiment that is identically distributed to the ideal world. Finally, we give a simulator that emulates the last hybrid.

In the proof, we let h^* be the index of the first honest mixer. Also, we consider two sets Ψ_{in} and Ψ_{hide}, both consisting of tuples $(X, Y) \in \mathbb{G}_1^2$. For Ψ_{in} (resp. Ψ_{hide}) we define a corresponding

map $\psi_{\mathsf{in}} : \mathbb{G}_1 \to \mathbb{G}_1$ (resp. ψ_{hide}) such that

$$\psi_{\mathsf{in}}(X) = \begin{cases} Y & \text{if } (X, Y) \in \Psi_{\mathsf{in}} \\ X & \text{else} \end{cases}$$

and analogously for ψ_{hide}. We assume that all the NIZK proofs verify and that all the ciphertexts verify (as otherwise the protocol would abort without producing any output).

Hybrid $\mathbf{H}_1$: We generate the CRS $\mathsf{crs}_{\mathtt{mx}}^{h^*}$ of the NIZK $\mathcal{NIZK}_{\mathtt{mx}}$ using the trapdoor mode and we simulate the NIZK proof of the first honest mixer $\mathcal{P}_{M_{h^*}}$. We generate the CRS $\mathsf{crs}_{\mathtt{mx}}^{j}$ for $j \neq h^*$ in perfect sound mode. Moreover, we pick an uniformly random index τ^* and we sample the $\mathsf{crs}_{\mathsf{sd}}$ using $\mathsf{ABOInit}(\mathtt{prm}, \tau^*)$. By the composable zero-knowledge property this hybrid is computationally indistinguishable from the previous one. The label τ^* is computationally hidden.

Hybrid $\mathbf{H}_2$: Let $\langle \mathtt{C}_{h^*-1,j} \rangle_{j\in[n]}$ be the list of ciphertexts received by the first honest mixer $\mathcal{P}_{M_{h^*}}$. Instead of re-randomizing all ciphertexts, here $\mathcal{P}_{M_{h^*}}$ decrypts and re-encrypts all the ciphertexts. By perfect re-randomizability, this hybrid and the precedent have are statistically close. More in details, by public-verifiability (Def. 4), we have that all the ciphertexts are valid, by perfect-rerandomizability all the ciphertext are in the support of the encryption scheme with overwhelming probability, so the decrypt and re-encrypt is well defined.

Hybrid $\mathbf{H}_3$: Here, instead of re-encrypting the same messages, we re-encrypt new fresh (and uncorrelated) messages. Namely, instead of creating $\mathtt{C}_{h^*,\tau_{h^*}(j)}$ as a re-encryption of $\mathtt{M}_{h^*-1,j}$, this ciphertext is set as an encryption of a random an independent message $\mathtt{H}_j$. Moreover, we populate the set Ψ_{hide} with the pairs $(\mathtt{M}_{h^*-1,j}, \mathtt{H}_j)_{j\in[n]}$ to associate $\mathtt{H}_j$ with $\mathtt{M}_{h^*-1,j}$, and then we simulate the ideal functionality $\mathcal{F}_{\mathsf{TDec}}$ to output $\Psi_{\mathsf{hide}}(\mathtt{M})$ instead of $\mathtt{M}$. This way the modification is not visible by looking at the decrypted ciphertexts. The indistinguishability of $\mathbf{H}_2$ and $\mathbf{H}_3$ can be reduced to the RCCA security of the PKE.

Hybrid $\mathbf{H}_4$: Let $\mathcal{V}_m$ (resp. $\mathcal{V}_{h^*}$) be the decryption of the list of ciphertexts output by the last mixer $\mathcal{P}_{M_m}$ (resp. by the first honest mixer $\mathcal{P}_{M_{h^*}}$). The hybrid $\mathbf{H}_4$ aborts if $\mathcal{V}_m \neq \mathcal{V}_{h^*}$. Using the perfect adaptive soundness of $\mathcal{NIZK}_{\mathtt{mx}}$ and the RCCA security and the public-verifiability of our PKE, we can show that this abort can happen only with negligible probability. We adapt the security argument of Faonio and Fiore [**?**] to our pv-Rand-PKE and our NIZK proof of "checksum".

Lemma 6. *Hybrids* $\mathbf{H}_3$ *and* $\mathbf{H}_4$ *are computationally indistinguishable.*

Proof (Sketch). We notice that the two hybrids diverge when $\mathbf{H}_4$ aborts but $\mathbf{H}_3$ does not. When they diverge necessarily there exists an index j^* such that $\mathtt{H}_{j^*} \in \mathcal{V}_{h^*} \setminus \mathcal{V}_m$. Consider the reduction B which receives input a public key pk and a ciphertext $\mathtt{C}^*$, the latter encrypts $\mathtt{H}_{j^*}$ (uniformly chosen and unknown to B). We show how B can compute the message $\mathtt{H}_{j^*}$ therefore breaking the RCCA-security. Briefly, the reduction B runs the hybrid $\mathbf{H}_4$ and inserts $\mathtt{C}^*$ in the place of $\mathtt{C}_{h^*,j^*}$ in the list $\mathcal{L}_{h^*}$. When the mixing phase terminates, the reduction B queries the ciphertexts in the lists $\mathcal{L}_{h^*}$ obtaining $\mathsf{Dec}(\mathsf{sk}, \mathtt{C}_{h^*,j})$ for $j \neq j^*$ and $j \in [n]$ and queries the ciphertexts in the list $\mathcal{L}_m$ obtaining $\mathsf{Dec}(\mathsf{sk}, \mathtt{C}_{m,j})$ for $j \in [n]$. Notice that all the ciphertexts would be successfully decrypted by a RCCA-decryption oracle, as the Mix-Net protocol (publicly) checks the validity of the ciphertexts at each stage. At this point, B checks that $\mathtt{H}_{j^*} \notin \mathcal{V}_m$, it can do so by looking if the guarded decryption oracle never answered with $\diamond$, and if so it outputs:

$$\sum_j \mathsf{Dec}(\mathsf{sk}, \mathtt{C}_{m,j}) - \sum_{j\neq j^*} \mathsf{Dec}(\mathsf{sk}, \mathtt{C}_{h^*,j}).$$

We analyze the winning probability of B. Recall that, for any $i \in [m]$ the list $\mathcal{L}_i = \langle \mathtt{C}_{i,j} \rangle_{j \in [n]}$ we can parse $\mathtt{C}_{i,j}$ as $([\mathbf{y}_{i,j}], \hat{\mathtt{C}}_{i,j})$. Moreover, the proofs $\pi_i^{\mathtt{mx}}$ for $i \neq h^*$ show that $\sum_j [\mathbf{y}_{i-1,j}] - [\mathbf{y}_{i,j}] \in span([(1, \mathbf{S})^\top]_1)$. By taking the conjunction of the statements proved by the proofs $\pi_{h^*+1}^{\mathtt{mx}}, \ldots, \pi_m^{\mathtt{mx}}$ we can infer that $\sum_j [\mathbf{y}_{h^*,j}] - [\mathbf{y}_{m,j}] \in span([(1, \mathbf{S})^\top)$, moreover, as all the ciphertexts in the two the lists are valid, by applying the linearity of the decryption, we can infer that $\sum_j \mathsf{Dec}(\mathsf{sk}, \mathtt{C}_{h^*,j}) - \sum_j \mathsf{Dec}(\mathsf{sk}, \mathtt{C}_{m,j}) = 0$. By the derivation above, the output of the reduction B is correct. The winning probability of B is then the probability of the event $\mathtt{H}_{j^*} \in \mathcal{V}_{h^*} \setminus \mathcal{V}_m$, the event that makes $\mathbf{H}_3$ and $\mathbf{H}_4$ diverge.

Hybrid $\mathbf{H}_5$: Simulate the ideal functionality $\mathcal{F}_{\mathsf{TDec}}$ in different way. Whenever the mixers agree on the decryption of a ciphertext $\mathtt{C} \in \mathcal{L}_m$, simulate the functionality $\mathcal{F}_{\mathsf{TDec}}$ by outputting a message chosen uniformly at random (without re-introduction) from the list $\mathcal{V}_{h^*-1}$. Notice, we don't need to compile the list Ψ_{hide} anymore as the mixers would only agree to decrypt ciphertexts from the last list $\mathcal{L}_m$ and $\mathcal{V}_m = \mathcal{V}_{h^*} = \Psi_{\mathsf{hide}}(\mathcal{V}_{h^*-1})$.

We can prove that $\mathbf{H}_5$ and $\mathbf{H}_4$ are identically distributed. In fact in $\mathbf{H}_4$, after the first honest mixer outputs $\mathcal{L}_{h^*}$, an unbounded environment $\mathcal{Z}$ knows that in Ψ_{hide} the element $\mathtt{H}_j$ for $j \in [n]$ is mapped to some other value in $\mathcal{V}_{h^*-1}$ but, from its view, it cannot know to which value. Such information is revealed only during decryption time. In other words, we could sample the permutation τ_{h^*} (uniformly at random) at decryption time.

It is easy to check that, at this point of the hybrid argument, the list of ciphertexts received by the first honest mixers is (a permutation of) the output of the protocol. Moreover, the ordering of the ciphertexts in the former list and in the latter list are uncorrelated.

With the next hybrids we make sure that the inputs of the honest senders are not discarded along the way from the first mixer to first honest mixer.

Hybrid $\mathbf{H}_6$: We introduce the set $\mathcal{M}_H$ of honest simulated messages and an initially empty list Ψ_{in}. Every message in $\mathcal{M}_H$ is randomly chosen in $\mathbb{G}$. The list Ψ_{in} is populated to map each simulated honest input in $\mathcal{M}_H$ to a corresponding real honest input, and we simulate the functionality $\mathcal{F}_{\mathsf{TDec}}$ by picking a message $\mathtt{M}$ chosen uniformly at random (without re-introduction) from the list $\mathcal{V}_{h^*-1}$ and outputting $\psi_{\mathsf{in}}(\mathtt{M})$ instead of $\mathtt{M}$. This hybrid is distributed the same as the previous one except if at decryption time a message in $\mathcal{M}_H$ is hit, namely if $\mathcal{V}_{h^*-1} \cap \mathcal{M}_H \neq \emptyset$ (in this case the map ψ_{in} would modify the returned value). However, since messages in $\mathcal{M}_H$ are randomly chosen and are not in the environment's view, this bad event happens only with negligible probability.

Hybrid $\mathbf{H}_7$: We encrypt the simulated sender inputs $\tilde{\mathtt{M}}_j$ instead of the the honest sender inputs. This hybrid can be shown indistinguishable from the previous one based on the RCCA security of the PKE scheme, and the zero-knowledge of $\mathcal{NIZK}_{\mathsf{sd}}$. Notice that after this change, the map ψ_{in} takes care of avoiding trivial differences in the output of $\mathcal{F}_{\mathsf{TDec}}$. The goal of the changes done in the last two hybrids is that we can keep track of the every honest ciphertext via its underlying message $\tilde{\mathtt{M}}_j \in \mathcal{M}_H$, which acts as a unique handle for it.

Hybrid $\mathbf{H}_8$: Let $\mathcal{V}_0$ be the decryption of the list of ciphertexts received by the first mixer. If a message $\tilde{\mathtt{M}}_j \in \mathcal{M}_H$ appears more than once in the list $\mathcal{V}_0$ then the hybrid aborts. By the soundness of $\mathcal{NIZK}_{\mathsf{sd}}$ and the RCCA security, we show that this abort happens only with negligible probability.

More in details, recall that we introduced in $\mathbf{H}_0$ an index τ^* such that the proofs for this label are perfectly sound. Let suppose that for an index $j \in [n]$ the message $\tilde{\mathtt{M}}_j$ appears twice in the list $\mathcal{V}_0$. Of course, this means that it appears once as decryption of the ciphertext $\mathtt{C}_{0,j}$ and once as a decryption of a ciphertext $\mathtt{C}_{0,j'}$ for $j' \in [n]$ and $j' \neq j$. Now we could break

the RCCA-security of the PKE scheme in the following way: we set $\mathtt{C}_{0,j}$ to be the challenge ciphertext of the RCCA experiment. As we don't know the randomness of $\mathtt{C}_{0,j}$, to create the NIZK proof $\pi_j^{\mathtt{sd}}$ we use the simulator of the $\mathcal{NIZK}_{\mathtt{sd}}$. The malicious sender $\mathcal{P}_{S_{j'}}$ produces a ciphertext $\mathtt{C}_{0,j'}$ and a NIZK proof $\pi_{j'}^{\mathtt{sd}}$ with label j'. One (unfruitful) idea could be to ask the RCCA decryption oracle for the decryption of $\mathtt{C}_{0,j'}$, but this won't be very effective, as the oracle would answer with $\diamond$. Our idea is instead to hope that $\tau^* = j'$ and if so extract the message $[\mathtt{M}]$ from the proof $\pi_{j'}^{\mathtt{sd}}$ and break the RCCA-security of the PKE scheme. The strategy works with probability $1/n - \mathtt{negl}(\lambda)$, as the index τ^* is only computationally hidden. In other words the $\mathcal{NIZK}_{\mathtt{sd}}$ proofs prevent the adversary of sending valid ciphertexts whose messages are correlated with honest ones.

Hybrid $\mathbf{H}_9$: Recall that $\mathcal{V}_{h^*-1}$ is the decryption of the list of ciphertexts input to the first honest mixer $\mathcal{P}_{M_{h^*}}$. If there exist an index j^* such that the message $\tilde{\mathtt{M}}_{j^*} \in \mathcal{M}_H$ does not appear or it appears more than once in the list $\mathcal{V}_{h^*-1}$ then the hybrid $\mathbf{H}_9$ aborts. This check essentially ensures that none of the inputs of the honest senders has been discarded or duplicated by the mixers. Using the public verifiability and RCCA security, we can show that abort can happen only with negligible probability. The argument is almost the same as in $\mathbf{H}_4$, but now a message $\tilde{\mathtt{M}}_{j^*}$ might either not appear or, additionally, appear more than once in the list $\mathcal{V}_{h^*-1}$. Let $J = \{j_1, \ldots, j_v\}$ where $v \neq 1$ such that $j \in J$ if and only if $\mathtt{C}_{h^*-1,j}$ decrypts to $\tilde{\mathtt{M}}_{j^*}$. We can compute $\tilde{\mathtt{M}}_{j^*}$ as:

$$\tilde{\mathtt{M}}_{j^*}(1-v) = -\sum_{j \neq j^*} \mathsf{Dec}(\mathsf{sk}, \mathtt{C}_{0,j}) + \sum_{j \notin J} \mathsf{Dec}(\mathsf{sk}, \mathtt{C}_{h^*-1,j})$$

We are ready to present a simulator such that the execution of the simulator with the ideal mixing functionality is indistinguishable from $\mathbf{H}_9$.

Simulator S:

Initialization: Simulate the ideal functionality $\mathcal{F}_{\mathsf{CRS}}$ and $\mathcal{F}_{\mathsf{TDec}}$ by sampling the CRSs for the NIZK system $\mathcal{NIZK}_{\mathtt{sd}}$ in ABO-mode on random label τ^*, sampling the $\mathsf{crs}_{\mathtt{mx}}^{h^*}$ of $\mathcal{NIZK}_{\mathtt{mx}}$ in trapdoor mode, and by a sampling key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{KGen}(\mathtt{prm})$.

Honest Senders: On activation of the honest sender $\mathcal{P}_{S_i}$ where $i \in [n]$, simulate it by executing the code of the honest sender on input the simulated message $\tilde{\mathtt{M}}_j$ chosen uniformly at random from the message space.

Extraction of the Inputs: Let $\mathcal{L}_{h^*-1}$ be the list produced by the malicious mixer $\mathcal{P}_{M_{h^*-1}}$. For any j, decrypt $\hat{\mathtt{M}}_j \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathtt{C}_{h^*-1,j})$ and if $\hat{\mathtt{M}}_j \notin \mathcal{M}_H$ then submit it as input to the ideal functionality $\mathcal{F}_{\mathsf{Mix}}$.

First Honest Mixer: Simulate the first honest mixer by computing $\mathcal{L}_{h^*}$ as a list of encryption of random messages $\mathtt{H}_j$ and simulating the proof $\pi_{h^*}^{\mathtt{mx}}$.

Decryption Phase: Receive from the ideal functionality $\mathcal{F}_{\mathsf{Mix}}$ the sorted output $\langle \mathtt{M}_1^o, \ldots, \mathtt{M}_n^o \rangle$. Whenever the mixers agree on the decryption of a ciphertext, simulate the ideal functionality $\mathcal{F}_{\mathsf{TDec}}$ by outputting a message from the sorted output randomly chosen (without reinsertion).

There are few differences between $\mathbf{H}_9$ and the execution of the simulator S with the ideal functionality $\mathcal{F}_{\mathsf{Mix}}$. The first is that the hybrid compiles the map ψ_{in} by setting a correspondence between the inputs of the honest senders and the simulated ones, and, during the decryption phase, uses the map ψ_{in} to revert this correspondence. On the other hand, the simulator does not explicitly set the map, as it does not know the inputs of the honest senders (which are sent directly to the functionality). However, at inputs submission phase the simulator picks a simulated input for any honest sender, and at decryption phase it picks a message from

the ordered list in output, which contains the inputs of the honest senders. By doing so, the simulator is implicitly defining the map ψ_{in}.

The second difference is that the simulator picks the outputs from the list $\langle \mathtt{M}_1^o, \ldots, \mathtt{M}_n^o \rangle$ while the hybrid $\mathbf{H}_9$ uses the list $\psi_{\mathsf{in}}(\mathcal{V}_{h^*-1})$. However, recall that the simulator extracts the corrupted inputs from the same list $\mathcal{V}_{h^*-1}$, and that, by the change introduced in $\mathbf{H}_9$, we are assured that all the inputs of the honest senders will be in the list $\psi_{\mathsf{in}}(\mathcal{V}_{h^*-1})$ (once and only once). Therefore the list of the messages output by the simulator and by the hybrid are the same.

7 Acknowledgements

We would like to thank Patrick Towa Nguenewou for pointing out an error in the counting of group elements required for the publicly verifiable version of our scheme.

First and second authors are supported by the Spanish Government through the projects Datamantium (ref. RTC-2016-4930-7), SCUM (RTI2018-102043-B-I00), and ERC2018-092822, and by the Madrid Regional Government under project BLOQUES (ref. S2018/TCS-4339). The work of the third author is partially supported by Spanish Government through project MTM2016-77213-R. The fourth author was supported by a Marie Curie "UPF Fellows" Postdoctoral Grant and by Project RTI2018-102112-B-I00 (AEI/FEDER,UE).

References

1. S. Bayer and J. Groth. Efficient zero-knowledge argument for correctness of a shuffle. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 263–280. Springer, Heidelberg, Apr. 2012.
2. M. Bellare, K. G. Paterson, and P. Rogaway. Security of symmetric encryption against mass surveillance. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 1–19. Springer, Heidelberg, Aug. 2014.
3. J. Camenisch, K. Haralambiev, M. Kohlweiss, J. Lapon, and V. Naessens. Structure preserving CCA secure encryption and applications. In D. H. Lee and X. Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 89–106. Springer, Heidelberg, Dec. 2011.
4. R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd FOCS*, pages 136–145. IEEE Computer Society Press, Oct. 2001.
5. R. Canetti and S. Goldwasser. An efficient threshold public key cryptosystem secure against adaptive chosen ciphertext attack. In J. Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 90–106. Springer, Heidelberg, May 1999.
6. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222. Springer, Heidelberg, May 2004.
7. R. Canetti, H. Krawczyk, and J. B. Nielsen. Relaxing chosen-ciphertext security. In D. Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 565–582. Springer, Heidelberg, Aug. 2003.
8. M. Chase, M. Kohlweiss, A. Lysyanskaya, and S. Meiklejohn. Malleable proof systems and applications. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 281–300. Springer, Heidelberg, Apr. 2012.
9. R. Chen, Y. Mu, G. Yang, W. Susilo, F. Guo, and M. Zhang. Cryptographic reverse firewall via malleable smooth projective hash functions. In J. H. Cheon and T. Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 844–876. Springer, Heidelberg, Dec. 2016.
10. R. Cramer, I. Damgård, and J. B. Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.
11. R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In L. R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64. Springer, Heidelberg, Apr. / May 2002.
12. I. Damgård, S. Faust, P. Mukherjee, and D. Venturi. Bounded tamper resilience: How to go beyond the algebraic barrier. In K. Sako and P. Sarkar, editors, *ASIACRYPT 2013, Part II*, volume 8270 of *LNCS*, pages 140–160. Springer, Heidelberg, Dec. 2013.

13. Y. Dodis, K. Haralambiev, A. López-Alt, and D. Wichs. Efficient public-key cryptography in the presence of key leakage. In M. Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 613–631. Springer, Heidelberg, Dec. 2010.
14. Y. Dodis, I. Mironov, and N. Stephens-Davidowitz. Message transmission with reverse firewalls—secure communication on corrupted machines. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 341–372. Springer, Heidelberg, Aug. 2016.
15. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and D. Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 10–18. Springer, Heidelberg, Aug. 1984.
16. A. Escala, G. Herold, E. Kiltz, C. Ràfols, and J. Villar. An algebraic framework for Diffie-Hellman assumptions. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Heidelberg, Aug. 2013.
17. A. Faonio and D. Fiore. Optimistic mixing, revisited. Cryptology ePrint Archive, Report 2018/864, 2018. https://eprint.iacr.org/2018/864.
18. A. Faonio and D. Fiore. Improving the efficiency of re-randomizable and replayable CCA secure public key encryption. In M. Conti, J. Zhou, E. Casalicchio, and A. Spognardi, editors, *ACNS 20, Part I*, volume 12146 of *LNCS*, pages 271–291. Springer, Heidelberg, Oct. 2020.
19. A. Faonio, D. Fiore, J. Herranz, and C. Ràfols. Structure-preserving and re-randomizable rcca-secure public key encryption and its applications. Cryptology ePrint Archive, Report 2019/955, 2019. https://eprint.iacr.org/2019/955.
20. A. Faonio and D. Venturi. Efficient public-key cryptography with bounded leakage and tamper resilience. In J. H. Cheon and T. Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 877–907. Springer, Heidelberg, Dec. 2016.
21. P. Fauzi, H. Lipmaa, J. Siim, and M. Zajac. An efficient pairing-based shuffle argument. In T. Takagi and T. Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, pages 97–127. Springer, Heidelberg, Dec. 2017.
22. S. Garg, A. Jain, and A. Sahai. Leakage-resilient zero knowledge. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 297–315. Springer, Heidelberg, Aug. 2011.
23. R. Gennaro and Y. Lindell. A framework for password-based authenticated key exchange. In E. Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 524–543. Springer, Heidelberg, May 2003. https://eprint.iacr.org/2003/032.ps.gz.
24. S. Goldwasser and Y. Lindell. Secure multi-party computation without agreement. *Journal of Cryptology*, 18(3):247–287, July 2005.
25. P. Golle, M. Jakobsson, A. Juels, and P. F. Syverson. Universal re-encryption for mixnets. In T. Okamoto, editor, *CT-RSA 2004*, volume 2964 of *LNCS*, pages 163–178. Springer, Heidelberg, Feb. 2004.
26. P. Golle, S. Zhong, D. Boneh, M. Jakobsson, and A. Juels. Optimistic mixing for exit-polls. In Y. Zheng, editor, *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 451–465. Springer, Heidelberg, Dec. 2002.
27. J. Groth. Rerandomizable and replayable adaptive chosen ciphertext attack secure cryptosystems. In M. Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 152–170. Springer, Heidelberg, Feb. 2004.
28. J. Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In X. Lai and K. Chen, editors, *ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 444–459. Springer, Heidelberg, Dec. 2006.
29. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In N. P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Heidelberg, Apr. 2008.
30. G. Herold, J. Hesse, D. Hofheinz, C. Ràfols, and A. Rupp. Polynomial spaces: A new framework for composite-to-prime-order transformations. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 261–279. Springer, Heidelberg, Aug. 2014.
31. G. Herold, M. Hoffmann, M. Klooß, C. Ràfols, and A. Rupp. New techniques for structural batch verification in bilinear groups with applications to groth-sahai proofs. In B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *ACM CCS 2017*, pages 1547–1564. ACM Press, Oct. / Nov. 2017.
32. C. S. Jutla and A. Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In K. Sako and P. Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 1–20. Springer, Heidelberg, Dec. 2013.
33. C. S. Jutla and A. Roy. Switching lemma for bilinear tests and constant-size NIZK proofs for linear subspaces. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 295–312. Springer, Heidelberg, Aug. 2014.
34. J. Katz and V. Vaikuntanathan. Round-optimal password-based authenticated key exchange. In Y. Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 293–310. Springer, Heidelberg, Mar. 2011.
35. E. Kiltz. Chosen-ciphertext security from tag-based encryption. In S. Halevi and T. Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 581–600. Springer, Heidelberg, Mar. 2006.
36. E. Kiltz and H. Wee. Quasi-adaptive NIZK for linear subspaces revisited. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 101–128. Springer, Heidelberg, Apr. 2015.

37. B. Libert, T. Peters, and C. Qian. Structure-preserving chosen-ciphertext security with shorter verifiable ciphertexts. In S. Fehr, editor, *PKC 2017, Part I*, volume 10174 of *LNCS*, pages 247–276. Springer, Heidelberg, Mar. 2017.
38. Y. Lindell, A. Lysyanskaya, and T. Rabin. On the composition of authenticated byzantine agreement. In *34th ACM STOC*, pages 514–523. ACM Press, May 2002.
39. S. Micali, C. Rackoff, and B. Sloan. The notion of security for probabilistic cryptosystems. In A. M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 381–392. Springer, Heidelberg, Aug. 1987.
40. I. Mironov and N. Stephens-Davidowitz. Cryptographic reverse firewalls. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 657–686. Springer, Heidelberg, Apr. 2015.
41. C. Namprempre, P. Rogaway, and T. Shrimpton. Reconsidering generic composition. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 257–274. Springer, Heidelberg, May 2014.
42. M. Naveed, S. Agrawal, M. Prabhakaran, X. Wang, E. Ayday, J.-P. Hubaux, and C. A. Gunter. Controlled functional encryption. In G.-J. Ahn, M. Yung, and N. Li, editors, *ACM CCS 2014*, pages 1280–1291. ACM Press, Nov. 2014.
43. O. Pereira and R. L. Rivest. Marked mix-nets. In M. Brenner, K. Rohloff, J. Bonneau, A. Miller, P. Y. A. Ryan, V. Teague, A. Bracciali, M. Sala, F. Pintore, and M. Jakobsson, editors, *FC 2017 Workshops*, volume 10323 of *LNCS*, pages 353–369. Springer, Heidelberg, Apr. 2017.
44. D. H. Phan and D. Pointcheval. OAEP 3-round: A generic and secure asymmetric encryption padding. In P. J. Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 63–77. Springer, Heidelberg, Dec. 2004.
45. M. Prabhakaran and M. Rosulek. Rerandomizable RCCA encryption. In A. Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 517–534. Springer, Heidelberg, Aug. 2007.
46. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th FOCS*, pages 543–553. IEEE Computer Society Press, Oct. 1999.
47. V. Shoup and R. Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. In K. Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 1–16. Springer, Heidelberg, May / June 1998.
48. D. Wikström. A universally composable mix-net. In M. Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 317–335. Springer, Heidelberg, Feb. 2004.
49. D. Wikström. A sender verifiable mix-net and a new proof of a shuffle. In B. K. Roy, editor, *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 273–292. Springer, Heidelberg, Dec. 2005.
50. D. Wikström. Verificatum, 2010. https://www.verificatum.com.

A Missing proofs from Section 3 (Rand-RCCA PKE)

Lemma 7. *The hybrids* $\mathbf{H}_5$ *and* $\mathbf{H}_4$ *are statistically close.*

The proof of the lemma, as for the proof of Lemma 3, show a reduction to Lemma 1. For readability we restate the Lemma 3 below using more convenient names for the variables:

Let k be a positive integer. For any matrices $\mathbf{E} \in \mathbb{Z}_q^{k+1\times k}, \mathbf{D} \in \mathbb{Z}_q^{k+1\times k}$ and any (possibly unbounded) adversary A:

$$\Pr\left[\begin{array}{c} \mathbf{v} \notin span(\mathbf{E}) \\ (\mathbf{u}-\mathbf{u}^*) \notin span(\mathbf{D}) \\ z = (\mathbf{g}+\mathbf{G}'\mathbf{u})^\top \mathbf{v} \end{array} \middle| \begin{array}{l} \mathbf{g} \leftarrow_\$ \mathbb{Z}_q^{k+1}, \mathbf{G}' \leftarrow_\$ \mathbb{Z}_q^{k+1\times k+1}; \\ (z,\mathbf{v},\mathbf{u}) \leftarrow_\$ \mathsf{A}^{\mathcal{O}(\cdot)}(\mathbf{E},\mathbf{D},\mathbf{E}^\top\mathbf{g},\mathbf{E}^\top\mathbf{G}',\mathbf{G}'\mathbf{D}) \end{array}\right] \leq 1/q,$$

where the adversary outputs a single query $\mathbf{u}^*$ to $\mathcal{O}(\cdot)$ which returns $\mathbf{g}+\mathbf{G}'\cdot\mathbf{u}^*$.

Proof. We prove the statement with a hybrid argument over the number of decryption queries of the adversary. Let the hybrid $\mathbf{H}_{4,i}$ be the experiment that answers the first i-th oracle queries as in $\mathbf{H}_5$ (namely, considering the decryption rule (iii)) and answers the remaining queries as in $\mathbf{H}_4$. Let Q_D be the number of decryption queries performed by the adversary A. It is easy to check that $\mathbf{H}_{4,0} \equiv \mathbf{H}_4$ and $\mathbf{H}_{4,Q_D} \equiv \mathbf{H}_5$.

On the other hand $\mathbf{H}_{4,i}$ and $\mathbf{H}_{4,i+1}$ differ when the $(i+1)$-th ciphertext $\mathtt{C} = (([\mathbf{u}]_1, [p]_1), [\mathbf{v}]_2, [\pi]_T)$ is such that "$\mathbf{v} \notin span(\mathbf{E})$ and $((\mathbf{u}-\mathbf{u}^*) \notin span(\mathbf{E})$ or $\mathbf{u}^*$ is unset)", but the decryption oracle (as it would be computed in $\mathbf{H}_4$) outputs a value different from $\perp$. In particular, the latter

implies that the proof $[\pi]_T$ verifies correctly. Let $\texttt{Sound}_\texttt{i}$ be such event. To conclude the proof of the lemma we prove the following proposition. Then a standard union bound gives us that the statistical distance between $\mathbf{H}_5$ and $\mathbf{H}_4$ is at most Q_D/q, which is negligible.

Proposition 2. $\Pr[\texttt{Sound}_i] \leq 1/q$.

Proof. We reduce an adversary A that causes event $\texttt{Sound}_i$ to occur into an adversary A' for the game of Lemma 1. Namely, we define an adversary A' for the experiment in the lemma which internally simulates the experiment $\mathbf{H}_{4,i+1}$ running with the adversary A.

Adversary $\mathsf{A}'(\mathbf{E}, \mathbf{D}, \mathbf{g}^\top\mathbf{E}, \mathbf{G'}^\top\mathbf{E}, \mathbf{G'D})$ with oracle access to $\mathcal{O}$:

1. Sample $\mathbf{a} \leftarrow_\$ \mathbb{Z}_q^{k+1}, \mathbf{f} \leftarrow_\$ \mathbb{Z}_q^{k+1}, \mathbf{F} \leftarrow_\$ \mathbb{Z}_q^{k+1\times k}$.
2. Sample $\mathbf{g'} \leftarrow_\$ \mathbb{Z}_q^{k+1}$, we set implicitly the matrix $\mathbf{G} = (\mathbf{G'g'}) \in \mathbb{Z}_q^{k+1\times k+1}$ and we compute $\mathbf{G}^\top\mathbf{E}$ and $\mathbf{GD}^*$.
3. Set the public key as:
$$\mathsf{pk} = \begin{pmatrix} [\mathbf{D}]_1, [\mathbf{E}]_2, [\mathbf{a}^\top\mathbf{D}]_1, [\mathbf{f}^\top\mathbf{D}]_T, [\mathbf{F}^\top\mathbf{D}]_1, \\ [\mathbf{g}^\top\mathbf{E}]_T, [\mathbf{G}^\top\mathbf{E}]_2, [\mathbf{GD}^*]_1, [\mathbf{FE}]_2 \end{pmatrix}$$
as described by the key generation algorithm and set the secret key $\mathsf{sk} = (\mathbf{a}, \mathbf{f}, \cdot, \mathbf{F}, \mathbf{g'})$.
4. Run the adversary A with input the public key pk. Answer the j-th decryption oracle query with ciphertext $\mathtt{C} = ([\mathbf{u}]_1, [p]_1, [\mathbf{v}]_2, [\pi]_T)$ as follows:
   (a) If $j \leq i$ and $\mathbf{u} \notin span(\mathbf{D})$ return $\perp$;
   (b) If $j \leq i$ and $\mathbf{v} \in span(\mathbf{E})$ compute, let $\mathbf{v} = \mathbf{Es}$:
$$\begin{aligned} [\mathtt{M}]_1 &\leftarrow [p - \mathbf{a}^\top\mathbf{u}]_1, \\ [\pi_1]_T &\leftarrow [(\mathbf{f} + \mathbf{F}\cdot\mathbf{v})^\top \cdot \mathbf{x}]_T, \\ [\pi_2]_T &\leftarrow [\mathbf{g}^\top\mathbf{E} + \mathbf{v}^\top \cdot \mathbf{G}^\top\mathbf{E}]_T \cdot \mathbf{s}. \end{aligned}$$
   If $\pi = \pi_1 + \pi_2$ then answer with $[\mathtt{M}]_1$, else anwer $\perp$;
   (c) If $\mathbf{v} \notin span(\mathbf{E})$ answer $\perp$;
   (d) If $j = i + 1$ then stop and return $(\pi - (\mathbf{f} + \mathbf{Fv})^\top\mathbf{u} - p(\mathbf{g'}^\top\mathbf{v}), \mathbf{v}, \mathbf{u})$.
5. Eventually, A outputs $[\mathtt{M}_0]_1, [\mathtt{M}_1]_1$. Sample $\mathbf{v}^* \leftarrow_\$ \mathbb{Z}_q^{k+1} \setminus span(\mathbf{E})$, and sample $\mathbf{u}^* \leftarrow_\$ \mathbb{Z}_q^{k+1} \setminus span(\mathbf{D})$, set $p^* = \mathbf{a}^\top\mathbf{u}^* + \mathtt{M}_{b^*}$, query the oracle $\mathcal{O}$ with the element $\mathbf{u}^*$, receive $\mathit{\Pi}' = \mathbf{g} + \mathbf{G'}\cdot\mathbf{u}^*$, and set $\mathit{\Pi} \leftarrow \mathit{\Pi} + p^*\mathbf{g'}$. Set $\mathbf{x}^* = ((\mathbf{u}^*)^\top, p^*)^\top$, and compute:
$$[\pi^*]_T \leftarrow [\mathit{\Pi}^\top \cdot \mathbf{v}^* + (\mathbf{f} + \mathbf{Fv}^*)^\top\mathbf{u}^*]_T \tag{4}$$
and send to the adversary the challenge ciphertext $\mathtt{C}^* = ([\mathbf{c}^*]_1, [p^*]_1, [\mathbf{v}]_2, [\pi^*]_T)$.
6. Answer the j-th decryption oracle query with ciphertext $\mathtt{C} = ([\mathbf{u}]_1, [p]_1, [\mathbf{v}]_2, [\pi]_T)$ as follows:
   (a) If $j \leq i$ and $\mathbf{v} \in span(\mathbf{E})$ execute the same as in step 4b.
   (b) If $j \leq i$ and $\mathbf{v} \notin span(\mathbf{E})$ do as follows:
      i. if $(\mathbf{x}^* - \mathbf{x}) \in span(\mathbf{D}^*)$ let $\mathbf{x} = \mathbf{x}^* + \mathbf{D}\boldsymbol{\gamma}$, compute
$$\begin{aligned} [\pi_1]_T &\leftarrow [(\mathbf{f}^\top + \mathbf{Fv})^\top\mathbf{u}]_T, \\ [\pi_2]_T &\leftarrow [(\mathit{\Pi} + \mathbf{GD}^*\boldsymbol{\gamma})^\top\mathbf{v}]_T \end{aligned}$$
   if $\pi = \pi_1 + \pi_2$ then set $[\mathtt{M}'] := [p - \mathbf{a}^\top\cdot\mathbf{u}]_1$ and if $[\mathtt{M}']_1 \in \{[\mathtt{M}_0]_1, [\mathtt{M}_1]_1\}$ answer $\diamond$ else with $[\mathtt{M}]_1$.
   if $\pi \neq \pi_1 + \pi_2$ answer $\perp$.

ii. if $(\mathbf{v}^* - \mathbf{v}) \notin span(\mathbf{E})$ then output $\perp$.

(c) If $j = i + 1$ then stop and return $(\pi - (\mathbf{f} + \mathbf{F}\mathbf{v})^\top \mathbf{u} - p(\mathbf{g'}^\top \mathbf{v}), \mathbf{v}, \mathbf{u}))$.

We show that the adversary perfectly simulates the hybrid $\mathbf{H}_{4,i}$ up to the i-th decryption query. By inspection, it is easy to check that up to step 3, the simulation is perfect. More interestingly, at step 4 the adversary A' uses its oracle to compute $\Pi = \mathbf{g} + \mathbf{G}'\mathbf{u}^* + p\mathbf{g}'$. Thanks to this information the adversary can compute the challenge ciphertext exactly as the hybrid experiment would do as shown in eq. 4. After this step, the adversary A' can easily answer the decryption queries whenever $j \leq i$ and $\mathbf{v} \in span(\mathbf{E})$ or $\mathbf{v} \notin span(\mathbf{E})$ and $(\mathbf{x}^* - \mathbf{x}) \notin span(\mathbf{D}^*)$. We show that the answers for the decryption queries where $j \leq i$, $\mathbf{v} \notin span(\mathbf{E})$ and $(\mathbf{x}^* - \mathbf{x}) \in span(\mathbf{D}^*)$ are distributed exactly as in the hybrid experiment, in fact:

$$\begin{aligned}(\Pi + \mathbf{G}\mathbf{D}^*\boldsymbol{\gamma})^\top \mathbf{v} &= \mathbf{g}^\top \mathbf{v} + (\mathbf{G}'\mathbf{x}^*)^\top \mathbf{v} + (p^*\mathbf{g}')^\top \mathbf{v} + (\mathbf{G}\mathbf{D}^*\boldsymbol{\gamma})^\top \mathbf{v} \\ &= \mathbf{g}^\top \mathbf{v} + (\mathbf{G}\mathbf{x}^*)^\top \mathbf{v} + (\mathbf{G}\mathbf{D}^*\boldsymbol{\gamma})^\top \mathbf{v} \\ &= \mathbf{g}^\top \mathbf{v} + (\mathbf{G}(\mathbf{x}^* + \mathbf{D}^*\boldsymbol{\gamma}))^\top \mathbf{v} \\ &= (\mathbf{g} + \mathbf{G}\mathbf{x})^\top \mathbf{v}.\end{aligned}$$

Finally, by definition of $\mathtt{Sound}_i$, the adversary A at the $(j+1)$-th query outputs a ciphertext that would correctly decrypt in the hybrid experiment and where $\mathbf{v} \notin span(\mathbf{E})$ and $(\mathbf{u}^* - \mathbf{u}) \notin span(\mathbf{D})$ with probability $\Pr[\mathtt{Sound}_i]$. Since the ciphertext correctly decrypts, it means that $\pi = (\mathbf{f} + \mathbf{F}\mathbf{v})^\top \mathbf{u} + (\mathbf{g} + \mathbf{G}\mathbf{x})^\top \mathbf{v}$, therefore the output of A' is a valid guess for the experiment of Lemma 1. However, the adversary A' can win only with probability $1/q$, and thus the lemma follows.

B Missing proofs from Section 4 (pv-Rand-RCCA PKE)

Proof. We first prove RCCA security. The proof proceeds in three main steps. Define a Rand-RCCA PKE scheme $\mathcal{PKE}^* = (\mathsf{KGen}^*, \mathsf{Enc}^*, \mathsf{Dec}^*)$ as the scheme that is the same as $\mathcal{PKE}_2$ except that: Enc^* outputs $([\mathbf{x}]_1, [\mathbf{v}]_2, [\pi]_T, \Pi)$ (namely it reveals $[\pi]_T$), and Dec^* is the algorithm that upon input the ciphertext $\mathtt{C} = ([\mathbf{x}]_1, [\mathbf{v}]_2, [\pi]_T, \Pi)$ runs $\mathsf{Dec}_1(\mathsf{sk}, ([\mathbf{x}]_1, [\mathbf{v}]_2, [\pi]_T))$, the decryption of the scheme of Sec. 3. We can reduce the security of $\mathcal{PKE}^*$ to the security of $\mathcal{PKE}_1$ as follows. The reduction upon a decryption query $\mathtt{C} = ([\mathbf{x}]_1, [\mathbf{v}]_2, [\pi]_T, \Pi)$ forwards $([\mathbf{x}]_1, [\mathbf{v}]_2, [\pi]_T)$ to its own challenger and answers with the same response; for the challenge, it first forwards the messages to its challenger, and then uses the zero-knowledge simulator to produce the proof Π^* to attach to the challenge ciphertext.

As second step, we define a Rand-RCCA PKE scheme $\mathcal{PKE}^{**}$ which is the same as $\mathcal{PKE}^*$ but where the decryption algorithm first parses $\mathtt{C} = ([\mathbf{x}]_1, [\mathbf{v}]_2, [\pi]_T, \Pi)$ and then decrypt $([\mathbf{x}]_1, [\mathbf{v}]_2, \Pi)$ using the algorithm Dec_2 described in Fig. 5. By adaptive soundness of the malleable proof system $\mathcal{NIZK}$ an interaction in the RCCA-security experiment with $\mathcal{PKE}^*$ and an interaction in the RCCA-experiment with $\mathcal{PKE}^{**}$ are statistically indistinguishable. We notice that $\mathcal{PKE}^{**}$ is essentially equivalent to $\mathcal{PKE}_2$, with the only difference that the encryption algorithm of $\mathcal{PKE}^{**}$ outputs $[\pi]_T$, which, however, is ignored by Dec^{**}. In particular it is straightforward to show that if $\mathcal{PKE}^{**}$ is secure, then so is $\mathcal{PKE}_2$. Combining the three steps we get that $\mathcal{PKE}_2$ is secure assuming so is $\mathcal{PKE}_1$.

We prove the three conditions of perfect re-randomizability.

1. For any $\mathsf{pk}, \mathsf{sk} \in \mathsf{KGen}(\mathtt{prm})$, any message $[\mathtt{M}]_1$, we show that $\mathtt{C}_0 \leftarrow_\$ \mathsf{Enc}(\mathsf{pk}, [\mathtt{M}]_1)$ and $\mathtt{C}_1 \leftarrow_\$ \mathsf{Rand}(\mathsf{pk}, \mathtt{C})$ where $\mathtt{C} = \mathsf{Enc}(\mathsf{pk}, [\mathtt{M}]_1; y, r)$ and are equivalently distributed.
   Consider the hybrid distribution that outputs $\mathtt{C}'$ such that

(a) Parse $\mathtt{C}$ as $([\mathbf{y}]_1, \pi)$;
(b) Sample $T \leftarrow_\$ \mathcal{T}_{\mathcal{PKE}'}$ (with associated randomness $\hat{\mathbf{r}}, \hat{\mathbf{s}}$) and compute $\hat{\Pi} \leftarrow_\$ \mathsf{P}(\mathsf{crs}, T_x([\mathbf{x}]_1, [\mathbf{v}]), T_w([\pi]_T, \mathbf{r},$
(c) Compute $[\hat{\mathbf{x}}]_1, [\hat{\mathbf{v}}] = T_x([\mathbf{x}]_1, [\mathbf{v}])$;
(d) Output $[\hat{\mathbf{x}}]_1, [\hat{\mathbf{v}}], \hat{\Pi}$.

By derivation privacy of the NIZK system we can easily prove that $\mathtt{C}_0$ and $\mathtt{C}'$ are equivalently distributed. Moreover, by perfect re-randomizability of the $[\mathbf{x}]_1, [\mathbf{v}]_2$ components, we have that $\mathtt{C}'$ and $\mathtt{C}_1$ are equivalently distributed.

2. Trivial, as we re-randomize the $[\mathbf{x}]$ by adding a vector $\mathbf{D}^*\mathbf{r}$, we remain in the same affine subspace. Moreover, if the proof Π does not verify, by correctness of ZKEval, neither $\hat{\Pi}$ does.
3. The condition trivially holds by the property of malleable NIZK and the perfect re-randomizability of $[\mathbf{x}]_1, [\mathbf{v}]_2$.

B.1 Details on the Malleable Proof

In this section we give the full details on the modifications we need to do to our RCCA scheme to make it publicly verifiable. The public key the terms $[\mathbf{f}^\top \mathbf{D}]_T$ and $[\mathbf{g}^\top \mathbf{E}]_T$ should be changed to $[\mathbf{f}^\top \mathbf{D}]_1$ and $[\mathbf{g}^\top \mathbf{E}]_2$ and the proof is now evaluated as:

$$[\pi_1]_T = e(([\mathbf{f}^\top \mathbf{D}]_1 \cdot \mathbf{r})^\top, [1]_2) + e([\mathbf{F}^\top \mathbf{D}]_1 \cdot \mathbf{r}, [\mathbf{v}]_2)$$
$$[\pi_2]_T = e([1]_1, [\mathbf{g}^\top \mathbf{E}]_2 \cdot \mathbf{s}) + e([\mathbf{x}]_1, [\mathbf{G}^\top \mathbf{E}]_2 \cdot \mathbf{s}).$$

Then, it is sufficient to prove that $[\pi]_T = [\pi_1]_T + [\pi_2]_T$ satisfies the sum of these pairing product equations. Notice that these changes do not compromise the security proof. In fact, we use statistical properties in Lemma 3 and Lemma 7.

Groth-Sahai Proofs can be instantiated under any $\mathcal{D}_k$-MDDH Assumption [16]. The verification equation uses a special projecting bilinear map $\tilde{e} : \mathbb{G}_1^{k+1} \times \mathbb{G}_2^{k+1} \to \mathbb{G}_T^m$. For the SXDH Assumption instantiation, $m = 4$ and $\tilde{e}([\mathbf{a}]_1, [\mathbf{b}]_2) = [\mathbf{a}\mathbf{b}^\top]$. In general, the map $\tilde{e}$ with the optimal m depends on $\mathcal{D}_k$ (not only on k), as was proven [30].

As we said, the main idea to extend the GS proof system so that it allows to prove NIZK of this particular type of equation, is to commit to $[\pi]_T$ instead of giving it in the clear. Our commitment to $[\pi]_T$ is defined over $\mathbb{G}_T^m$. The proof Π includes a commitment $[\mathbf{c}_\pi]_T \in \mathbb{G}_T^m$ and the proof that the value $[\pi]_T$ which is an opening of $[\mathbf{c}_\pi]_T$ is of the right form.

To simplify the exposition, we give the details of the proof only for the SXDH instantiation ($k = 1$) of the RCCA scheme and of GS proofs. The generalization to other matrix distributions is straightforward following the description of the GS proof system for any $\mathcal{D}_k$-MDDH Assumption ([16,30]).

The common reference string of the GS proof system consists of commitment keys $[\mathbf{u}_1]_1, [\mathbf{u}_2]_1 \in \mathbb{G}_1^2$ (resp. $[\mathbf{v}_1]_2, [\mathbf{v}_2]_2 \in \mathbb{G}_2^2$) to group elements in $\mathbb{G}_1$ (resp. $\mathbb{G}_2$). A commitment to an element $[y]_1 \in \mathbb{G}_1$ is defined as $[\mathbf{c}_y]_1 = ([y]_1, [0]_1)^\top + r_1[\mathbf{u}_1]_1 + r_2[\mathbf{u}_2]_1$, $r_1, r_2 \leftarrow \mathbb{Z}_p$ and similarly for elements in $\mathbb{G}_2$. The prover shows that:

$$[\pi]_T = e([\mathbf{f}^\top \mathbf{D}]_1 \mathbf{r}, [1]_2) + e([\mathbf{F}^\top \mathbf{D}]_1 \mathbf{r}, [\mathbf{v}]_2) + e([1]_1, [\mathbf{g}^\top \mathbf{E}]_2 \mathbf{s}) + e([\mathbf{x}]_1, [\mathbf{G}^\top \mathbf{E}]_2 \mathbf{s}) \tag{5}$$

is satisfied, where $[\pi]_T$, $[\mathbf{f}^\top \mathbf{D}]_1 \mathbf{r}$, $[\mathbf{F}^\top \mathbf{D}]_1 \mathbf{r}$, $[\mathbf{g}^\top \mathbf{E}]_2 \mathbf{r}$, $[\mathbf{G}^\top \mathbf{E}]_2 \mathbf{s}$ are values unknown to the verifier and which are committed to as part of the proof. More specifically, the proof includes:

1. A commitment to $[\pi]_T$ defined by sampling $r_{ij} \leftarrow \mathbb{Z}_p$ and:

$$[\mathbf{c}_\pi]_T = \left[\begin{pmatrix} \pi & 0 \\ 0 & 0 \end{pmatrix} \right]_T + \sum_{i,j=1,2} r_{ij} [\mathbf{u}_i \mathbf{v}_j^\top]_T \in \mathbb{G}_T^4,$$

2. A commitment $[\mathbf{c}_0]_1 \in \mathbb{G}_1^2$ to $[\mathbf{f}^\top \mathbf{D}\mathbf{r}]_1$ and commitments $[\mathbf{c}_1]_1, [\mathbf{c}_2]_1$ to the first and second components of the vector $[\mathbf{F}^\top \mathbf{E}\mathbf{r}]_1$ (recall we are in the case where $k = 1$ and $[\mathbf{F}^\top \mathbf{E}\mathbf{r}]_1$ is a vector of two group elements),
3. A commitments $[\mathbf{d}_0]_2$ to $[\mathbf{g}^\top \mathbf{E}\mathbf{s}]_2$ and commitments $[\mathbf{d}_1]_2, [\mathbf{d}_2]_2, [\mathbf{d}_3]_2 \in \mathbb{G}_2^2$ to (respectively) the first, second and third components of $[\mathbf{G}^\top \mathbf{E}\mathbf{s}]_2$,
4. A GS proof that equation (5) is satisfied,
5. A proof that the commitments $[\mathbf{c}_0]_1, [\mathbf{c}_1]_1, [\mathbf{c}_2]_1 \in \mathbb{G}_1^2$ and $[\mathbf{d}_0]_2, [\mathbf{d}_1]_2, [\mathbf{d}_2]_2, [\mathbf{d}_3]_2 \in \mathbb{G}_2^2$ are well formed. This is proven with one proof of membership in linear spaces in each group [33,36]. For more details, the statement one needs to prove in $\mathbb{G}_1$ is:

$$\left[\begin{pmatrix} \mathbf{D}\mathbf{r} \\ \mathbf{c}_0 \\ \mathbf{c}_1 \\ \mathbf{c}_2 \end{pmatrix}\right]_1 \in span\left(\left[\begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ (\mathbf{f}^\top \mathbf{D}, 0)^\top & \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ ((\mathbf{F}^\top \mathbf{D})_1, 0)^\top & \mathbf{0} & \mathbf{0} & \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{0} & \mathbf{0} \\ ((\mathbf{F}^\top \mathbf{D})_2, 0)^\top & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{u}_1 & \mathbf{u}_2 \end{pmatrix}\right]_1\right),$$

where $(\mathbf{F}^\top \mathbf{D})_i$ denotes the ith coordinate of $\mathbf{F}^\top \mathbf{D}$. For the elements of $\mathbb{G}_2$, well-formedness is proved similarly.

For completeness, what is relevant to note is that the commitment in $\mathbb{G}_T$ is such that the commitment keys can be factored out in the terms required for the verification equation. That is, $r_{ij}[\mathbf{u}_i \mathbf{v}_j^\top]_T = \tilde{e}([\tilde{\boldsymbol{\theta}}_{ij}]_1, [\mathbf{v}_j]_2)$, where, $[\tilde{\boldsymbol{\theta}}_{ij}]_1 = [r_{ij}\mathbf{u}_i]_1$. Therefore, if the verification equation is:

$$\tilde{e}([\mathbf{c}_0]_1, ([1]_2, [0]_2)) + \sum_{i=1}^{2} \tilde{e}([\mathbf{c}_i]_1, ([v_i]_2, [0]_2)) + \tilde{e}(([1]_1, [0]_1), [\mathbf{d}_0]_2) + \sum_{i=1}^{3} \tilde{e}(([x_i]_1, [0]_1), [\mathbf{d}_i]_2) - [\mathbf{c}_\pi]_T$$
$$= \sum_{i=1,2} \tilde{e}([\mathbf{u}_i]_1, [\boldsymbol{\pi}_i]_2) + \sum_{i=1,2} \tilde{e}([\boldsymbol{\theta}_i]_1, [\mathbf{v}_i]_2),$$

where v_i (resp. x_i) denotes the ith coordinate of $\mathbf{v}$ (resp. x). Then, if $[\hat{\boldsymbol{\pi}}_i]_2$, $[\hat{\boldsymbol{\theta}}_i]_1$ are proofs computed as in the original GS proof system for equation (5) ignoring the term $[\pi]_T$, then $[\boldsymbol{\theta}_i]_1 = [\hat{\boldsymbol{\theta}}_i] + [\tilde{\boldsymbol{\theta}}_{1i}]_1 + [\tilde{\boldsymbol{\pi}}_{2i}]_1$ and $[\boldsymbol{\pi}_i]_2 = [\hat{\boldsymbol{\pi}}]_2$. We observe that perfect soundness holds because the commitment keys are chosen to define perfectly binding commitments. The argument is exactly as in the original paper [29], because by the properties of the tensor product, in this case the commitment to $[\pi]_T$ is also perfectly binding.

We prove composable zero-knowledge as in [29]. In particular, this means that the zero-knowledge property is proven in a setting in which the commitment keys are changed to an indistinguishable set of perfectly hiding keys. By the properties of the tensor product, in this case the commitment to $[\pi]_T$ is also perfectly hiding. To simulate a proof for equation (5), we simply take as a witness for satisfiability the trivial solution. The proof is perfectly distributed as the commitments are perfectly hiding and for every set of commitments, the proof has the uniform distribution conditioned on satisfying the verification equation. The proof that $[\mathbf{c}_0]_1, [\mathbf{c}_1]_1, [\mathbf{c}_2]_1 \in \mathbb{G}_1^2$ and $[\mathbf{d}_0]_2, [\mathbf{d}_1]_2, [\mathbf{d}_2]_2, [\mathbf{d}_3]_2 \in \mathbb{G}_2^2$ are well formed can be simulated with the simulator of the argument of membership in linear spaces. The proof is perfectly distributed because the argument is perfect zero-knowledge and the commitments are perfectly hiding, so in particular there exists an opening so that the statement is true.

We need to define ZKEval algorithm. We use the homomorphic property of GS proofs. The idea is that we can add to the commitments of $[\mathbf{f}^T \mathbf{D}]_1 \mathbf{r}$ and $[\mathbf{F}^T \mathbf{D}]_1 \mathbf{r}$ the values $[\mathbf{f}^T \mathbf{D}]_1 \cdot \hat{\mathbf{r}}$ and $[\mathbf{F}^T \mathbf{D}]_1 \cdot \hat{\mathbf{r}}$ (similarly for the $\mathbf{s}$ components) and refresh the randomness of the commitments. The commitment $[\mathbf{c}_\pi]$ can also be updated by adding the value $[\pi']_T = [\hat{\pi}_1]_T + [\hat{\pi}_2]_T$ where $[\hat{\pi}_1]_T = [\mathbf{f}^\top \mathbf{D}]_T \cdot \hat{\mathbf{r}} + e([\mathbf{F}^\top \mathbf{D}]_1 \cdot \hat{\mathbf{r}}, [\hat{\mathbf{v}}]_2) + e([\mathbf{u}]_1, [\mathbf{F}\mathbf{E}]_2 \cdot \hat{\mathbf{s}})$ and $[\hat{\pi}_2]_T = [\mathbf{g}^\top \mathbf{E}]_T \cdot \hat{\mathbf{s}} + e([\hat{\mathbf{x}}]_1, [\mathbf{G}^\top \mathbf{E}]_2 \cdot$

$\hat{\mathbf{s}})+e([\mathbf{GD}^*]_1\cdot\hat{\mathbf{r}},[\mathbf{v}]_2)$ (exactly as in the Rand algorithm of Sec. 3) and refreshing the randomness. The corresponding new GS proof can be computed by the homomorphic property of GS proofs. Also, the linear subspace argument in step 5 is instantiated with the adaptive sound scheme of Kiltz and Wee [36] which is homomorphic.

Efficiency. The size of the ciphertexts of the publicly verifiable scheme is $14|\mathbb{G}_1|+15|\mathbb{G}_2|+4|\mathbb{G}_T|$. The number of pairings for verification is 32 for the GS proof and 20 for the argument of linear spaces, which can be reduced to $10+20$ by batch verifying the GS equation using the techniques of [31].

C Missing proofs from Section 5 (tSE-cm NIZK)

Proof. First we notice that ZKEval is complete. In fact, $(f,\mathcal{R},\mathcal{T})$ is suitable we have that for any $T\in\mathcal{T}$ the transformation $T_w(\cdot)$ is invariant respect to the f and therefore the transformed statement-witness is in the relation $\mathcal{R}'$.

Recall that the adversary A for the derivation privacy of $\mathcal{NIZK}$ outputs (x,w,π,T) where $\pi=(\mathtt{C},\pi')$ is a valid proof of x and T is a allowable transformation for $\mathcal{T}$

We first notice that we can assume that exists $\mathtt{M},r$ such that $\mathtt{C}\in\mathsf{Enc}(\mathsf{pk},\mathtt{M};r)$ by condition 3 of perfect re-randomizability, therefore we can compute, although inefficiently, from $\mathtt{C}$ the randomness r.

Consider an hybrid experiment $\mathbf{H}_1$ where once received (x,w,π,T) from the adversary A we compute r from $\mathtt{C}$ and we compute a new proof $\pi'\leftarrow\mathsf{P}'(\mathsf{crs}',(\mathsf{pk},\hat{\mathtt{C}},x),(w,r+\hat{r}))$ where $\hat{\mathtt{C}}=\mathsf{Rand}(\mathsf{pk},\mathtt{C};\hat{r})$.

It is easy to see that $\mathbf{Exp}^{\mathtt{der\text{-}priv}}_{\mathsf{A},\mathcal{NIZK}}$ conditioned on the challenge bit equal to 1 (namely, when ZKEval is used) and $\mathbf{H}_1$ are statiscally close by the statistical derivation privacy of $\mathcal{NIZK}'$. the reduction B would simply emulate the hybrid outputing $((\mathsf{pk},\hat{\mathtt{C}},x),(w,r),T'=(T,\hat{r})$.

Moreover, we can prove that $\mathbf{H}_1$ is distributed equivalently to $\mathbf{Exp}^{\mathtt{der\text{-}priv}}_{\mathsf{A},\mathcal{NIZK}}$ conditioned on the challenge bit equal to 0. In fact, the only difference between the two distribution is that in one case the ciphertext is fresh while in the other case is a re-randomization.

Theorem 4. *For any suitable $(f,\mathcal{R},\mathcal{T})$ the proof system $\mathcal{NIZK}$ described above is true-simulation controlled-malleable f-extractable.*

Proof. We describe both a simulator and an extractor for the tse-cm-NIZK $\mathcal{NIZK}$.

- Let $\mathsf{Sim}(tp_s,x)$ be the simulator that parses $tp_s=(\mathsf{pk},tp'_s)$ and computes $\mathtt{C}\leftarrow_{\$}\mathsf{Enc}(\mathsf{pk},\circ)$ and $\pi\leftarrow\mathsf{Sim}'(tp'_s,(\mathsf{pk},\mathtt{C},x))$ where Sim' is the simulator of $\mathcal{NIZK}'$.
- Let $\mathsf{Ext}(tp_e,x,\pi)$ be the extractor that parses tp_e ans sk and $\pi=(\mathtt{C},\pi')$ and outputs $\mathsf{Dec}(\mathsf{sk},\mathtt{C})$.

We consider a sequence of hybrid experiments.

- The first experiment $\mathbf{H}_0$ is the $\mathbf{Exp}^{\mathtt{tse\text{-}cm}}_{\mathsf{A},\mathsf{Ext},\mathcal{NIZK}}$, namely, the oracle $\mathcal{SIM}$ upon the i-th query (x_i,w_i) first checks that $(x_i,w_i)\in\mathcal{R}$ and if so it adds x_i in $\mathcal{Q}_x$ and w_i in $\mathcal{Q}_w$ and outputs $\mathtt{C}_i\leftarrow\mathsf{Enc}(\mathsf{pk},\circ)$ and $\pi'_i\leftarrow\mathsf{Sim}'(tp'_s,(\mathsf{pk},\mathtt{C}_i,x_i))$.
- Let $\mathbf{H}_{1,j}$ be the same as $\mathbf{H}_0$ but where the first j ciphertexts are valid encryption of $f(w)$. Specifically, the oracle $\mathcal{SIM}$ upon the i-th query (x_i,w_i) if $i>j$ then it behaves as in $\mathbf{H}_0$ otherwise it first checks that $(x_i,w_i)\in\mathcal{R}$ and if so adds w_i in $\mathcal{Q}_w$ and outputs $\mathtt{C}_i\leftarrow\mathsf{Enc}(\mathsf{pk},f(w_i))$ and $\pi'_i\leftarrow\mathsf{Sim}'(tp'_s,(\mathsf{pk},\mathtt{C}_i,x_i))$.
- Let $\mathbf{H}_2$ be the same as $\mathbf{H}_{1,q}$, where q is the number of queries made by A, but where the proofs for $\mathcal{NIZK}'$ are not simulated. Specifically, the oracle $\mathcal{SIM}$ upon the i-th query (x_i,w_i) first checks that $(x_i,w_i)\in\mathcal{R}$ and if so adds w_i in $\mathcal{Q}_w$ and outputs $\mathtt{C}_i\leftarrow\mathsf{Enc}(\mathsf{pk},f(w_i);r_i)$ where $r_i\leftarrow\{0,1\}^\lambda$ and $\pi'_i\leftarrow\mathsf{P}'(\mathsf{crs}',(\mathsf{pk},\mathtt{C}_i,x_i),(w_i,r_i))$.

Lemma 8. *For any* $j \in \mathbb{N}$, $|\Pr[\mathbf{H}_{1,j} = 1] - \Pr[\mathbf{H}_{1,j+1} = 1]| \in \mathtt{negl}(\lambda)$.

Proof. We show a reduction to the RCCA security of the PKE-scheme. Consider an adversary B for the RCCA-security experiment. The adversary B upon input pk generates the parameter $\mathsf{crs}', tp'_s \leftarrow \mathsf{Init}(\mathtt{prm}_G)$ and runs $\mathsf{A}(\mathsf{crs})$ where $\mathsf{crs} = (\mathsf{pk}, \mathsf{crs}')$. At the i-th query (x_i, w_i) made by A:

- if $i < j$, the adversary B returns to the adversary A the values $\mathtt{C}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, f(w_i))$ and $\pi'_i \leftarrow \mathsf{Sim}'(tp'_s, (\mathsf{pk}, \mathtt{C}_i, x_i))$;
- if $i = j$ the adversary B sends the challenge messages $(\circ, f(w_j))$ to its own challenger and receives $\mathtt{C}^*$, it returns to A the values $\mathtt{C}^*$ and $\pi'_j \leftarrow \mathsf{Sim}'(tp'_s, (\mathsf{pk}, \mathtt{C}^*, x_i))$;
- If $i > j$, the adversary B returns to the adversary A the values $\mathtt{C}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, \circ)$ and $\pi'_i \leftarrow \mathsf{Sim}'(tp'_s, (\mathsf{pk}, \mathtt{C}_i, x_i))$;

Eventually, the adversary A outputs a tuple x, π where $\pi = (\mathtt{C}, \pi')$, the adversary B forwards $\mathtt{C}$ to its own decryption oracle, let z be the answer from the decryption oracle. First the adversary B checks that the proof π' verifies and if not output 0. Secondly, the adversary B if $z \neq \circ$ then output 1 if and only if $g(x, z, \omega) = 0$, else it outputs 1 if and only if for any $w \in \mathcal{Q}_w$ we have $g(x, f(w), \omega) = 0$.

We notice that B runs in polynomial time in λ. We check that B perfectly simulates the $\mathbf{H}_{1,i+b}$ experiment when the challenge bit of the RCCA experiment is equal to b. It is easy to check that the adversary B perfectly simulates the hybrid experiments until the adversary A outputs x, π. The hybrids output 1 iff the π' verifies and either $z \neq \circ$ and for all w such that $f(w) = z$ we have that $(x, w) \notin \mathcal{R}$ (which can be efficiently computable by condition (1) of Def. 9) or $z = \circ$ and for all w such that $w \in \mathcal{Q}_w$ we have $(x, w) \notin \mathcal{R}$ (again, it can be efficiently computable). Notice that if for any (x_i, w_i) queried to $\mathcal{SIM}$ if $g(x, T_w(w_i), \omega) = 0$, then for any $T \in \mathcal{T}$, $T_x(x_i) \neq x$. In fact, $1 = g(T_x(x_i), f(T_w(w_i)), \omega) = g(T(x_i), f(w_i), \omega) \neq g(x, f(w_i), \omega)$, and thus $T(x_i) \neq x$. This concludes the proof of the lemma.

D Controlled-Malleable Smooth-Projective Hash Functions

In this section we formalize the technique at the core of our Rand-RCCA PKE scheme, that is a structure-preserving smooth-projective hash function (SPHF) which allows for a controlled malleability of the instances and tags. We call our primitive *Controlled-Malleable Smooth-Projective Hash Functions*.

We formalize our primitive by extending the notion of Malleable Smooth Projective Hash Functions (mSPHFs) introduced by Chen *et al.* in [9]. Their framework additionally has the notion of key-malleability which is not required in our main construction of Sec. 3. However, we show in this section that we can easily add this property to our mSPHF.

This notion found interesting applications in the context of subversion-resilient cryptography[2]; in particular, it allows to generically instantiate Cryptographic Reverse Firewalls (see Mironov and Stephens-Davidowitz [40]) for CPA-secure message-transmission protocols and oblivious-transfer protocols from various assumptions.

Our notion of controlled-malleable SPHFs supports the same functionalities of the mSPHFs in [9] and, additionally, uses a security notion strictly stronger than the classical smoothness property.

A smooth projective hash function (with tags) is a tuple of algorithms $\mathtt{HF} = (\mathsf{Setup}, \mathsf{ProjK}, \mathsf{Hash}, \mathsf{PHash})$ where:

- $\mathsf{Setup}(1^\lambda)$ generates public parameters pp that contains the descriptions of groups $\mathcal{K}, \mathcal{P}, \mathcal{X}, \mathcal{Y}, \mathcal{W}, \mathcal{T}$. Moreover, the pp contains the description of a subgroup $\mathcal{L} \subset \mathcal{X}$. Elements in $\mathcal{L}$ can be efficiently sampled together with a witness $w \in \mathcal{W}$. All the groups are in additive notations.

– $\mathsf{ProjK}(k)$ takes in a hash key $k \in \mathcal{K}$ and produces a projective key $p \in \mathcal{P}$.
– $\mathsf{Hash}(\mathsf{pp}, k, x, t)$ takes as input a hash key $k \in \mathcal{K}$, an instance $x \in \mathcal{X}$, and a tag $t \in \mathcal{T}$, and it produces a hash value $y \in \mathcal{Y}$.
– $\mathsf{PHash}(\mathsf{pp}, k, x, w, t)$ takes as input a hash key $k \in \mathcal{K}$, an instance $x \in \mathcal{X}$, a witness w for x, and a tag $t \in \mathcal{T}$, and it produces a hash value $y \in \mathcal{Y}$.

A SPHF as above is *structure preserving* if $\mathcal{P}, \mathcal{X}, \mathcal{Y}, \mathcal{T}$ are all vector spaces of $\mathbb{G}_1$ or $\mathbb{G}_2$ or $\mathbb{G}_T$, while $\mathcal{K}, \mathcal{W}$ are vector spaces of $\mathbb{Z}_q$ and all the algorithms can be defined via pairing-product equations.

The classical properties of a SPHF are described below:

Definition 10 (Projective). *A* $\mathtt{HF}$ *is projective if for any* $k \in \mathcal{K}$ *and* $p = \mathsf{ProjK}(\mathsf{pp}, k)$*, for any* $x \in \mathcal{L}$ *with witness* w*, and for any tag* $t \in \mathcal{T}$*, we have* $\mathsf{PHash}(\mathsf{pp}, p, x, w, t) = \mathsf{Hash}(\mathsf{pp}, k, x, t)$*.*

Definition 11 (Smoothness). *A* $\mathtt{HF}$ *is smooth if for any* $x \in \mathcal{X} \setminus \mathcal{L}$ *the distributions below are statistically indistinguishable:*

$$(\mathsf{pp}, p, x, t, \mathsf{Hash}(\mathsf{pp}, k, x, t))_{\mathsf{pp} \in \mathsf{Setup}}, \quad (\mathsf{pp}, p, x, t, y)_{\mathsf{pp} \in \mathsf{Setup}}$$

where $k \leftarrow\!\!\$\ \mathcal{K}$*,* $p = \mathsf{ProjK}(\mathsf{pp}, k)$ *and* $y \leftarrow\!\!\$\ \mathcal{Y}$*.*

Notice that the properties above are information theoretic. Finally, we require a computational property on the set $\mathcal{L}$.

Definition 12 (Hard Subset Membership Problem). *We assume that for all PPT adversaries* A*:*

$$|\Pr[\mathsf{A}(\mathsf{pp}, x) = 1 : x \leftarrow\!\!\$\ \mathcal{L}] - \Pr[\mathsf{A}(\mathsf{pp}, x) = 1 : x \leftarrow\!\!\$\ \mathcal{X} \setminus \mathcal{L}]| \in \mathtt{negl}(\lambda),$$

Chen *et al.* specialized the notion of SPHF to malleable SPHF. We import their definitions below. To simplify the exposition, we use a slightly less general syntax which matches more closely our instantiation. In particular, Chen *et al.* consider the Gennaro and Lindell [23] (GL-SPHF) definition for SPHF where the projective hash key can depend on an element of $\mathcal{X}$. Instead, we consider the Katz and Vaikuntanathan [34] (KV-SPHF) definition, and we extend their syntax to support tags.

A malleable SPHF $\mathtt{HF}$ is a SPHF with the following additional algorithms:

– $\mathsf{Setup}'(1^\lambda)$ outputs public parameters pp and a trapdoor parameter td.
– $\mathsf{CheckTag}(\mathsf{td}, t, t')$ where t and t' are tags, outputs a bit b.
– $\mathsf{MaulK}(\mathsf{pp}, p, k')$ outputs a mauled projective key $\tilde{p}$.
– $\mathsf{RandX}(\mathsf{pp}, x, w')$ outputs a re-randomized element $\hat{x}$.
– $\mathsf{RandT}(\mathsf{pp}, t, \tau)$ where τ is a randomness, outputs a re-randomized tag $\hat{t}$.
– $\mathsf{RandH}(\mathsf{pp}, x, t, y, w', \tau)$ outputs a re-randomized hash value $\hat{y}$.

All the algorithms are deterministic PT except for Setup' which is a PPT algorithm. Notice that we could alternatively define $\mathsf{MaulK}, \mathsf{RandX}, \mathsf{RandT}$ and RandH as PPT algorithms, keeping the values k', w', τ implicit. However, we prefer to make them explicit, as it is easier to define correctness. More in details, the algorithm MaulK, indeed, expects to receive as input a fresh random key k', the algorithm RandX, which re-randomizes the instance x, expects to receive a random witness w', and the algorithm RandT, which re-randomizes the tag t, expect to receive a random string τ.

A malleable SPHF is *structure preserving* if the randomness τ is a vector of $\mathbb{Z}_q$ elements and all the algorithms can be defined via pairing-product equations.

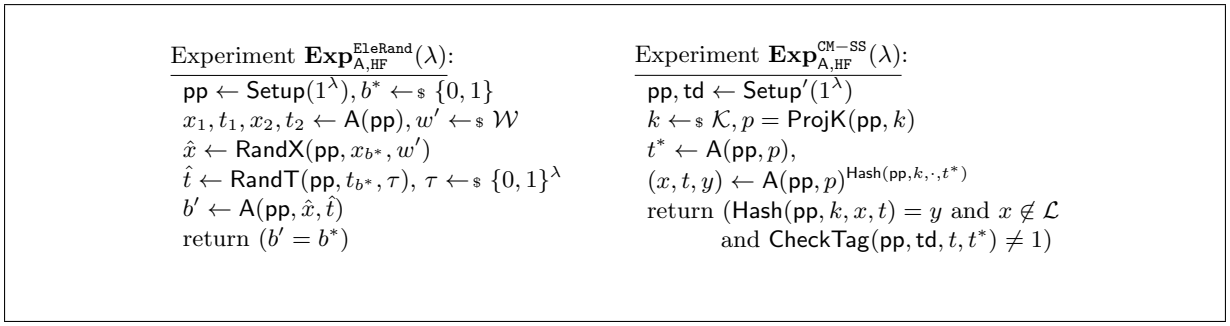

Fig. 9: The malleable SPHF experiments.

The definition of Chen *et al.* considers malleability for both the projection keys and the elements. Although for the Rand-RCCA scheme in Section 3 we do not need malleability for the projection keys, we define it for completeness in this section. Looking ahead, our construction of mSPHF indeed satisfies this property.

Definition 13 (Projection Key Malleability). *A* HF *is projection key-malleable iff for any* $\mathsf{pp} \in \mathsf{Setup}$*, any* $k, k' \in \mathcal{K}$ *let* $p = \mathsf{ProjK}(\mathsf{pp}, k)$*, any element* $x \in \mathcal{X}$*, (1)* $\mathsf{MaulK}(\mathsf{pp}, p, k') = \mathsf{ProjK}(\mathsf{pp}, k + k')$ *and (2)* $\mathsf{Hash}(\mathsf{pp}, k + k', x) = \mathsf{Hash}(\mathsf{pp}, k, x) + \mathsf{Hash}(\mathsf{pp}, k', x)$*.*

The definition of Chen *et al.* differs from ours in some aspects that we clarify. First, their definition additionally considers a security game where the adversary wins if it can distinguish between the re-randomization of two adaptively chosen projection keys. We do not need to define the same security game because, by (1), by the fact that we restrict on SPHF with an algebraic structure (namely, all the spaces are groups) and by sampling k' uniformly at random from $\mathcal{K}$ we obtain a similar (actually stronger) property.

Next, we extend the *Element Re-Randomizability* property of Chen *et al.* to the setting with tags.

Definition 14 (Element and Tag Re-randomizability). *A* HF *is element-rerandomizable if the following holds:*

- **Element Indistinguishability.** *For any PPT adversary* A*:*

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

  where the experiment is defined in Fig. 9.
- **(Perfect) Re-randomization Consistency.** *For any* $\mathsf{pp} \in \mathsf{Setup}$*, any* $k \in \mathcal{K}$*, any element* $x \in \mathcal{X}$*, any witness* w'*, let* $\hat{x} = \mathsf{RandX}(\mathsf{pp}, x, w')$ *and* $\hat{t} = \mathsf{RandT}(\mathsf{pp}, t, \tau)$ *then*

$$\mathsf{Hash}(\mathsf{pp}, k, \hat{x}, \hat{t}) = \mathsf{RandH}(\mathsf{pp}, x, t, y, w', \tau).$$

- **Membership Preservation.** *For any element* $x \in \mathcal{X}$ *and witness* w' *let* $\hat{x} = \mathsf{RandX}(\mathsf{pp}, x, w', r)$*; then we have* $\hat{c} \in \mathcal{L}$ *if and only if* $c \in \mathcal{L}$*.*
- **Tag Preservation.** *For any* $t \in \mathcal{T}$ *and randomness* τ*, we have*

$$\mathsf{CheckTag}(\mathsf{pp}, t, \mathsf{RandT}(\mathsf{pp}, t, \tau)) = 1.$$

Finally, we require the following additional properties:

Definition 15. *A* HF *is controlled-malleable simulation-sound if the following holds:*

– *for any PPT adversary* A*:*

$$\mathbf{Adv}^{\mathtt{CM-SS}}_{\mathsf{A},\mathtt{HF}}(\lambda) := \Pr\left[\mathbf{Exp}^{\mathtt{CM-SS}}_{\mathsf{A},\mathtt{HF}}(\lambda) = 1\right] \leq 1/|\mathcal{Y}|$$

where the experiment is defined in Fig. 9.
– *The public parameters as output by* Setup *and* Setup' *are identically distributed.*

Some remarks follow. First, our notion extends the syntax of Chen *et al.* by adding two extra algorithms, $\mathsf{CheckTag}$ and RandT whose semantic is related to adding support for tags. In particular, RandT allows to re-randomize a tag, while $\mathsf{CheckTag}$ allows to check that the re-randomized tag is computed, indeed, as a re-randomization of the original tag. Second, we consider an extra setup algorithm that outputs a trapdoor td. Notice that the element indistinguishability property does not hold when given the trapdoor (as $\hat{t}$ can be traced back to either t_0 or t_1). Finally, we require that the advantage above is less or equal to $1/|\mathcal{Y}|$ so that the definition is indeed a strictly stronger notion of smoothness. One could consider a more generous definition where the advantage above is negligible. However, our construction achieves this stronger notion.

D.1 Our Construction

Consider the following mSPHF $\mathtt{HF}$ with parameters $\ell, k \in \mathbb{N}$:

– $\mathsf{Setup}'(1^\lambda)$ runs $\mathtt{prm}_G \leftarrow \mathsf{GGen}(1^\lambda)$ to generate group parameters and samples $\mathbf{D}, \mathbf{E} \leftarrow_\$ \mathcal{D}_k$, finds $\mathbf{z} \neq 0$ such that $\mathbf{z}^\top \mathbf{E} = 0$, set $\mathcal{K} = \mathbb{Z}_q^{k+1\times \ell+1}, \mathcal{P} = \mathbb{G}_1^{k+2}, \mathcal{X} = \mathbb{G}_1^{k+1}, \mathcal{Y} = \mathbb{G}_T, \mathcal{W} = \mathbb{Z}_q^k$ and $\mathcal{T} = \mathbb{G}_2^\ell$ outputs $\mathsf{pp} = \mathtt{prm}_G, [\mathbf{D}]_1, [\mathbf{E}]_2$ and the trapdoor $\mathsf{td} = \mathbf{z}$.
– $\mathsf{ProjK}(\mathsf{pp}, k)$ where $k = (\mathbf{f}, \mathbf{F})$ outputs $p = ([\mathbf{f}^\top \mathbf{D}, \mathbf{F}^\top \mathbf{D}]_1, [\mathbf{F}\mathbf{E}]_2)$.
– $\mathsf{Hash}(\mathsf{pp}, k, x, t)$ where $x = [\mathbf{u}]_1$ and $t = [\mathbf{v}]_2$ outputs:

$$y = e(\mathbf{f}^\top \cdot [\mathbf{u}]_1, [1]_2) + e(\mathbf{F}^\top \cdot [\mathbf{u}]_1, [\mathbf{v}]_2).$$

– $\mathsf{PHash}(\mathsf{pp}, p, x, w, t)$ where $w = \mathbf{r}$ and $[\mathbf{x}] = [\mathbf{D}]\mathbf{r}$ outputs:

$$y = e([\mathbf{f}^\top \mathbf{D}]_1 \cdot \mathbf{r}, [1]_2) + e([\mathbf{F}^\top \mathbf{D}]_1 \cdot \mathbf{r}, [\mathbf{v}]_2).$$

– $\mathsf{CheckTag}(\mathsf{pp}, \mathsf{td}, t, t')$ where $\mathsf{td} = \mathbf{z}$, $t = [\mathbf{v}]_2$, and $t' = [\mathbf{v}']_2$ return 1 if and only if $\mathbf{z}^\top \cdot [\mathbf{v} - \mathbf{v}']_2 = 0$.
– $\mathsf{MaulK}(\mathsf{pp}, p, k')$ where $k' = \mathbf{f}', \mathbf{F}'$ outputs:

$$[(\mathbf{f} + \mathbf{f}')^\top \mathbf{D}, (\mathbf{F} + \mathbf{F}')^\top \mathbf{D}]_1, [(\mathbf{F} + \mathbf{F}')\mathbf{E}]_2.$$

– $\mathsf{RandX}(\mathsf{pp}, x, w')$ where $w' = \hat{\mathbf{r}}$ outputs $[\mathbf{u}]_1 + [\mathbf{D}]_1 \cdot \hat{\mathbf{r}}$.
– $\mathsf{RandT}(\mathsf{pp}, t, \tau)$ where $\tau = \hat{\mathbf{s}}$ outputs $[\mathbf{v}]_2 + [\mathbf{E}]_2 \cdot \hat{\mathbf{s}}$.
– $\mathsf{RandH}(\mathsf{pp}, x, t, y, w', \tau)$ outputs:

$$y + e([\mathbf{f}^\top \mathbf{D}]_1 \hat{\mathbf{r}}, [1]_2) + e([\mathbf{F}^\top \mathbf{D}]_1 \cdot \hat{\mathbf{r}}, [\mathbf{v} + \mathbf{E}\hat{\mathbf{s}}])_2 + e([\mathbf{u}]_1, [\mathbf{F}\mathbf{E}]_2 \cdot \hat{\mathbf{s}})$$

Theorem 6. *The SPHF* $\mathtt{HF}$ *described above is structure preserving, projection key malleable, element-tag rerandomizable and controlled-malleable simulation sound.*

Proof (Sketch.). The proof of projection key-malleability is straightforward. It is easy to prove also that $\mathtt{HF}$ is element-tag re-randomizable, under the $\mathcal{D}_k$ assumption for $[\mathbf{D}]_1$ and $[\mathbf{E}]_2$. In fact, in an hybrid step we can compute both $\hat{x} = [\mathbf{u}]_1 + [\mathbf{u}^*]_1$ and $\hat{t} = [\mathbf{v}]_2 + [\mathbf{v}^*]_2$ where $[\mathbf{u}^*] \leftarrow_\$ \mathbb{G}_1^{k+1}$ and $[\mathbf{v}^*]_2 \leftarrow_\$ \mathbb{G}_2^\ell$. Once in this new hybrid experiment, the distributions of $\hat{x}$ and $\hat{t}$ are independent of the challenge bit.

Finally, the controlled-malleable simulation soundness follows from Lemma 1. In a little bit more of details, the game of the Lemma gives the adversary access to $(\mathbf{f} + \mathbf{F})\mathbf{v}^*$ for a $\mathbf{v}^*$ of its choice. Given such information, the reduction can easily simulate the oracle $\mathsf{Hash}(\mathsf{pp}, (\mathbf{f}, \mathbf{F}), \cdot, [\mathbf{v}^*])$. Moreover, the check done by $\mathsf{CheckTag}$ holds if and only if $\mathbf{v} - \mathbf{v}^* \notin span(\mathbf{E})$, in fact, if $\mathbf{v} - \mathbf{v}^* \in span(\mathbf{E})$ then there exists $\mathbf{s}'$ such that $\mathbf{z}^\top(\mathbf{v} - \mathbf{v}^*) = \mathbf{z}^\top\mathbf{E}\mathbf{s}' = 0$. On the other hand, $\mathbf{E}$ has full rank and thus $\mathbf{z}|\mathbf{E}$ is a basis for $\mathbb{Z}_q^{k+1}$.

E Auditable Protocols with Bulletin Board

UC Security Model. We use the Universal Composability model of Canetti [4], our notation for multi-party protocols comes from the book of Cramer, Damgård, and Nielsen [10]. In this formulation of the UC model, the basic computational components are called *interactive agents*. Briefly, an interactive agent $\mathcal{A}$ is a computational device that receives and sends messages on named ports, and holds an internal state. More in particular, an interactive agent $\mathcal{A}$ has a collection of named *inports* where it receives messages and a collection of named *outports* where it sends messages. Given two agents $\mathcal{A}$ and $\mathcal{B}$ the union of them, denoted as $\mathcal{A} \circ \mathcal{B}$ is the *interactive system* where every outport of $\mathcal{A}$ (resp. $\mathcal{B}$) with name `N` is connected to an inport of $\mathcal{B}$ (resp. $\mathcal{A}$) with the same name `N` (if it exists). An interactive system where all the inports and outports are connected is said *closed*. The collection of all the ports that are not connected is called the *open ports* of the interactive system.

All the agents in the framework have one or more inports with name ending in `infl` or $\texttt{infl}_i$. We call these special ports the *influence ports* of the agent. If an agent has a inport with named `N.infl` then it must have a matching port with name `N.lk`. We call these special ports the *leakage ports* of the agent. Leakage and influence ports are very important: agents are activated by a special message on the influence port and they return their activation on the leakage port; parties of a protocol are corrupted by a special message on the influence port and from that point on they return their full state on their leakage port and proxy all the messages from the influence port (resp. to the leakage port) to (resp. from) their other ports (in this paper we consider only *active* corruption of the parties); the leakage and influence port of an ideal functionality model which information is allowed to leak and how an adversary can influence the functionality[11] (for more details see chapter 4 of [10]).

The agent of an ideal functionality with name `F` additionally has n inports named $\texttt{F.in}_1, \ldots,$ $\texttt{F.in}_n$, and n outports named $\texttt{F.out}_1, \ldots, \texttt{F.out}_n$. These $2n$ ports are called the *protocol ports*. A protocol Π consists of n parties $\mathcal{P}_1, \ldots, \mathcal{P}_n$ where each party is an interactive agent. A protocol Π has a *protocol name* `F`. We let a protocol Π and the ideal functionality $\mathcal{F}$ that Π is supposed to implement have the same name. We name the port `N` of the protocol Π as `F.N`. A protocol also has ports connecting it to the resource $\mathcal{R}$ with *resource name* `R`. The *resource* $\mathcal{R}$ takes care of moving messages between parties of the protocol. In addition to moving messages, it also models the leakage of the communication network and the possible influence that an attacker might have on the network.

A simulator S for a protocol Π with name `F` is defined as a polytime interactive system with an open inport named `F.infl` and an open outport named `F.lk`; these two ports allow S to connect to an ideal functionality $\mathcal{F}$ with name `F`. In addition to these two ports, a simulator has a collection of ports corresponding to all the open ports of the protocol Π with resource $\mathcal{R}$. As a consequence, the interactive systems $(\mathcal{F} \circ \mathsf{S})$ and $(\Pi \circ \mathcal{R})$ have the same open ports.

[11] For example, the ideal functionality of a private channel might send message on the leakage port of the form "P_1 sent a message to P_2 of lenght n" and might receive message on influence port of the form "deliver the message to P_2".

An environment $\mathcal{Z}$ for a protocol Π with resource $\mathcal{R}$ is an interactive system that has the dual open ports of $\Pi \circ \mathcal{R}$ (namely, $\mathcal{Z} \circ \Pi \circ \mathcal{R}$ forms a closed interactive system). In this paper we consider protocols that are secure against environments which perform static corruption of the parties. This means that corruption takes place before the protocol starts its execution and, as already mentioned above, the adversary can deviate from the protocol specification in any arbitrary way. Let Env^{static} be the set of all polytime interactive agents $\mathcal{Z}$ of this flavor. Specifically, $\mathcal{Z} \in Env^{\mathsf{static}}$ if, at the first activation, and only in this activation, it sends a list of messages `corrupt` to the influence ports of the protocol. Two systems are indistinguishable to an environment $\mathcal{Z}$ if it cannot tell them apart, except with negligible advantage, by sending and receiving messages on the open ports of the systems. More formally, the environment plays with a system and then, at the end of the execution outputs a bit b. Two systems $\mathcal{A}$ and $\mathcal{B}$ are indistinguishable to a class of environments Env if for any $\mathcal{Z} \in Env$ the bit output by $\mathcal{Z}$ interacting with $\mathcal{A}$ and the bit output by $\mathcal{Z}$ interacting with $\mathcal{B}$ are indistinguishable.

Definition 16. *Let* $\mathtt{F}$ *and* $\mathtt{R}$ *denote arbitrary protocol names. Let* $\mathcal{F}$ *be any ideal functionality with name* $\mathtt{F}$*,* Π *be any protocol with protocol name* $\mathtt{F}$ *and resource name* $\mathtt{R}$*, and let* $\mathcal{R}$ *be an ideal functionality with name* $\mathtt{R}$*. We say that* $\Pi \circ \mathcal{R}$ $\mathsf{UC\text{-}realizes}$ $\mathcal{F}$ *in the environments from Env if there exist a simulator* S *such that* $\Pi \circ \mathcal{R}$ *and* $\mathsf{S} \circ \mathcal{F}$ *are indistinguishable to the class Env.*

We state the UC Theorem (see Thm 4.20 of [10]).

Theorem 7 (The UC Theorem). *Let Env be a class of environments. Let* $\Pi_{\mathtt{F}}$ *be a protocol with protocol name* $\mathtt{F}$ *and resource name* $\mathtt{G}$*. Let* $\Pi_{\mathtt{G}}$ *be a protocol with protocol name* $\mathtt{G}$ *and resource name* $\mathtt{H}$*. Let* $\mathcal{F}, \mathcal{G}, \mathcal{H}$ *be ideal functionalities with name* $\mathtt{F}, \mathtt{G}, \mathtt{H}$*.*

If $\Pi_{\mathtt{F}} \circ \mathcal{G}$ $\mathsf{UC\text{-}realizes}$ $\mathcal{F}$ *and* $\Pi_{\mathtt{G}} \circ \mathcal{H}$ $\mathsf{UC\text{-}realizes}$ $\mathcal{G}$*, then* $(\Pi_{\mathtt{F}} \circ \Pi_{\mathtt{G}}) \circ \mathcal{H}$ $\mathsf{UC\text{-}realizes}$ $\mathcal{F}$*.*

Auditable Protocols. In our model all the protocols have, as resource, a bulletin board BB with name `BB`. The functionality bulletin board is the same as defined in [48], and can be realized using an authenticated broadcast channel (c.f. [24,38]). This resource is shared by all the protocols. To formalize this, we slightly tweak the notion of composition of interactive agents. Specifically, given two protocols Π_1 and Π_2 both with resource BB, we denote the composition of $(\Pi_1 \circ \mathrm{BB})$ and $(\Pi_2 \circ \mathrm{BB})$ as $(\Pi_1 \circ \Pi_2 \circ \mathrm{BB})$, where the protocol ports of the resource BB in the composed system are the union of the protocol ports of BB in $\Pi_1 \circ \mathrm{BB}$ and $\Pi_2 \circ \mathrm{BB}$.

An auditable protocol is a tuple (Π, Audit), where Audit is a non-interactive PPT algorithm and Π is a multi-party protocol. To define auditability we consider an auditor party $\mathcal{P}_A$ defined as an agent with name `A` and parametrized by an algorithm Audit(see Fig. 10). More in detail, the party $\mathcal{P}_A$ cannot be corrupted, (i.e., it would reject all corruption messages sent to its influence port), and whenever activated it reads the full content of the BB, it selects a relevant portion τ (e.g., the messages of the specific protocol execution to audit), runs the audit algorithm $b \leftarrow \mathsf{Audit}(\tau)$, and returns b on its leakage port.

To make auditable protocols composable, and in particular to enable the composability of Audit algorithms, we define an ideal audit procedure for the output of an ideal functionality. Notice that the output of an ideal functionality is, by definition, correct as it is produced as the correct computation on the private inputs. Therefore, given an ideal functionality/resource $\mathcal{F}$ with name $\mathtt{F}$ we define the ideal audit $\mathsf{Audit}^*_{\mathtt{F}}$ that trivially accepts if it finds in the BB a specific output authenticated by $\mathcal{F}$.

However, we need to instruct the ideal functionality to write its output messages on the bulletin board. To get this, given an ideal functionality $\mathcal{F}$, we consider a wrapper $\mathcal{W}[\mathcal{F}]$ that proxies all the messages of $\mathcal{F}$ and also forwards all the messages in the protocol's output ports to the bulletin board using a special port name. This way, the bulletin board can syntactically

distinguish what event to register in the "ideal board" $\mathcal{D}^I$ and what to register in the "real board" $\mathcal{D}^R$. The auditor party $\mathcal{P}_A$ is the only agent that can get access to the ideal board $\mathcal{D}^I$.

Formally, the auditor receives the message X from the environment, reads the bulletin boards $\mathcal{D}^R, \mathcal{D}^I$ and, when running the ideal audit procedure $\mathsf{Audit}^*_{\mathtt{F}}$ of the ideal functionality $\mathcal{F}$ (with name $\mathtt{F}$), returns true iff $\exists c : (c, \mathtt{F}, X) \in \mathcal{D}^I$ (here, c is just the index in the table $\mathcal{D}^I$ of the tuple $(\mathtt{F}, X)$).

Definition 17. *Given an ideal functionality $\mathcal{F}$ with name $\mathtt{F}$, a resource $\mathcal{R}$ with name $\mathtt{R}$, and a class of environments Env, we say that an auditable protocol (Π, Audit), where Audit is a non-interactive PPT algorithm, and Π is a protocol with protocol name $\mathtt{F}$ and resource name $\mathtt{R}$,* UC-auditable-realizes *$\mathcal{F}$ with resource $\mathcal{R}$ for environments in Env if:*

1. *The protocol Π with resources $\mathcal{W}[\mathcal{R}]$ and* BB UC-realizes *$\mathcal{F} \circ$ BB for all environments $\mathcal{Z} \in Env$.*
2. *Let S be the simulator of Π, the agent systems $(\mathcal{W}[\mathsf{S}] \circ \mathcal{W}[\mathcal{F}] \circ \mathrm{BB} \circ \mathcal{P}_A[\mathsf{Audit}])$ and $(\mathcal{W}[\mathsf{S}] \circ \mathcal{W}[\mathcal{F}] \circ \mathrm{BB} \circ \mathcal{P}_A[\mathsf{Audit}^*_{\mathtt{F}}])$ are indistinguishable for all environments $\mathcal{Z} \in Env$.*

We show an equivalent of the UC master theorem for the auditable protocols. First before we need to define what the composition of auditing algorithms means.

Definition 18 (Composition of PPT algorithms). *Let $(\Pi_{\mathtt{F}}, \mathsf{Audit}_{\mathtt{F}})$* UC-auditable-realizes *$\mathcal{F}$ with resource $\mathcal{G}$ and let $(\Pi_{\mathtt{G}}, \mathsf{Audit}_{\mathtt{G}})$* UC-auditable-realizes *$\mathcal{G}$ with resource $\mathcal{H}$.*

Let $\mathsf{Audit}_{\mathtt{F}}/\mathsf{Audit}_{\mathtt{G}}$ be the algorithm that is syntactically equivalent to $\mathsf{Audit}_{\mathtt{F}}$ but that, whenever $\mathsf{Audit}_{\mathtt{F}}$ makes a call to (the ideal auditing algorithm) $\mathsf{Audit}^*_{\mathtt{F}}$ the algorithm $\mathsf{Audit}_{\mathtt{F}}/\mathsf{Audit}_{\mathtt{G}}$ makes a call to (the real auditing algorithm) $\mathsf{Audit}_{\mathtt{F}}$.

For simplicity, below we state the UC theorem for the class of enviroments Env^{static}:

Theorem 8. *Let $(\Pi_{\mathtt{F}}, \mathsf{Audit}_{\mathtt{F}})$* UC-auditable-realizes *$\mathcal{F}$ with resource $\mathcal{G}$ and class of enviroments Env^{static} and let $(\Pi_{\mathtt{G}}, \mathsf{Audit}_{\mathtt{G}})$* UC-auditable-realizes *$\mathcal{G}$ with resource $\mathcal{H}$ and class of enviroments Env^{static}. Let $\Pi' := \Pi_{\mathtt{F}} \circ \Pi_{\mathtt{G}}$ and $\mathsf{Audit}' := \mathsf{Audit}_{\mathtt{F}}/\mathsf{Audit}_{\mathtt{G}}$, the protocol (Π', Audit')* UC-auditable-realizes *$\mathcal{F}$ with resource $\mathcal{H}$ and class of enviroments Env^{static}.*

Proof. The point 1 of the definition 17 readily holds by the UC theorem (Thm.7). So we need to prove the point 2 of the definition.

First we notice that the simulator of Π' is equal to $\mathsf{Sim}_{\mathtt{F}} \circ \mathsf{Sim}_{\mathtt{G}}$ where $\mathsf{Sim}_{\mathtt{F}}$ (resp. $\mathsf{Sim}_{\mathtt{G}}$) is the simulator provided by the security of $\Pi_{\mathtt{F}}$ (resp. $\Pi_{\mathtt{G}}$). Also we notice that:

$$\mathcal{W}[\mathsf{Sim}_{\mathtt{F}} \circ \mathsf{Sim}_{\mathtt{G}}] \equiv \mathcal{W}[\mathsf{Sim}_{\mathtt{F}}] \circ \mathcal{W}[\mathsf{Sim}_{\mathtt{G}}].$$

This property can be verified by inspection on the wrapper $\mathcal{W}$.

Finally we define a party $\mathcal{P}_{\tilde{A}}$. Such party has the same name of $\mathcal{P}_A[\mathsf{Audit}']$ and the same set of ports connected to the environment of $\mathcal{P}_A[\mathsf{Audit}']$, and it interacts with the influence and leakage ports of $\mathcal{P}_A[\mathsf{Audit}_{\mathtt{G}}]$. Moreover, it is connected to the BB. Also $\mathcal{P}_{\tilde{A}}$ is not corruptible and runs the code of $\mathcal{P}_A[\mathsf{Audit}_{\mathtt{F}}]$ but whenever the algorithm makes a call to $\mathsf{Audit}^*_{\mathtt{G}}$, which would look at $\mathcal{D}^I$, the party will, instead, send a message to the influence port of $\mathcal{P}_A[\mathsf{Audit}_{\mathtt{G}}]$. It is easy to see that:

$$\mathcal{P}_{\tilde{A}} \circ \mathcal{P}_A[\mathsf{Audit}_{\mathtt{G}}] \equiv \mathcal{P}_A[\mathsf{Audit}'], \qquad \mathcal{P}_{\tilde{A}} \circ \mathcal{P}_A[\mathsf{Audit}^*_{\mathtt{G}}] \equiv \mathcal{P}_A[\mathsf{Audit}_{\mathtt{F}}] \tag{6}$$

By the facts stated above we can write the following derivation of equivalences:

$$\begin{aligned} \mathcal{W}[\mathsf{Sim}_{\mathtt{F}} \circ \mathsf{Sim}_{\mathtt{G}}] \circ \mathcal{W}[\mathcal{G}] \circ \mathrm{BB} \circ \mathcal{P}_A[\mathsf{Audit}'] \equiv \\ \mathcal{W}[\mathsf{Sim}_{\mathtt{F}}] \circ \mathcal{W}[\mathsf{Sim}_{\mathtt{G}}] \circ \mathcal{W}[\mathcal{G}] \circ \mathrm{BB} \circ \mathcal{P}_A[\mathsf{Audit}'] \equiv \\ \mathcal{W}[\mathsf{Sim}_{\mathtt{F}}] \circ \mathcal{W}[\mathsf{Sim}_{\mathtt{G}}] \circ \mathcal{W}[\mathcal{G}] \circ \mathrm{BB} \circ \mathcal{P}_{\tilde{A}} \circ \mathcal{P}_A[\mathsf{Audit}_{\mathtt{G}}]. \end{aligned}$$

Wrapper $\mathcal{W}[\mathcal{F}]$ (resp. $\mathcal{W}[\mathsf{S}]$):

The wrapper has the same open ports of the functionality $\mathcal{F}$ with name $\mathtt{F}$ (resp. the simulator S with name $\mathtt{S}$). Moreover the wrapper has an outport with name $\mathtt{in}_{\mathtt{F}}$ that connects it to the bulletin board. (Resp. the wrapper for any ideal functionality simulated by S with name $\mathtt{F}$ as an outport with name $\mathtt{in}_{\mathtt{F}}$ that connects it to the bulletin board.)

1. On activation proxy all the messages on its inports to the inports of $\mathcal{F}$ and activate $\mathcal{F}$ (resp. S and activate S);
2. Once $\mathcal{F}$ (resp. S) returns its activation proxy all the messages on the outports of $\mathcal{F}$ (resp. S) to its outports, moreover if $\mathcal{F}$ has sent a message X on an protocol port $\mathtt{out}_i$ then send the message $(\mathtt{write}, \mathtt{F}, (i, X))$ to the port $\mathtt{in}_{\mathtt{F}}$ (resp. if S has sent a message X on a protocol port $\mathtt{F}, \mathtt{out}_i$ then send the message $(\mathtt{write}, \mathtt{F}, (i, X))$ to the port $\mathtt{in}_{\mathtt{F}}$).

Ideal functionality BB:

The ideal functionality has enough protocol ports to connect with all the protocols' agents of the agent system, for each ideal functionality in the agent system, it has special protocol ports named with $\mathtt{in}_{\mathtt{F}}$ where $\mathtt{F}$ is the name of the functionality, it has a special protocol inport $\mathtt{in}_{\mathtt{A}}$ and special protocol outport named $\mathtt{out}_{\mathtt{A}}$ (connected to the auditor party).

1. The functionality holds three databases $\mathcal{D}, \mathcal{D}^I$ and $\mathcal{D}^T$. We call $\mathcal{D}^I$ the database of the ideal functionalies' messages and $\mathcal{D}^T$ the database of the messages *on transit*. At first activation initialize the databases as empty, and the counters $c \leftarrow 1, c^I \leftarrow 1$.
2. Upon message $(\mathtt{write}, \mathsf{pid}, x)$ on the inport $\mathtt{BB.in}_i$, with $x \in \{0,1\}^*$, write (pid, i, x) on the database $\mathcal{D}^T$. Send the message $(\mathtt{write}, \mathsf{pid}, i, x)$ to the outport $\mathtt{BB.lk}$.
3. Upon message $(\mathtt{write}, X)$ on the inport $\mathtt{BB.infl}$, if X appears in $\mathcal{D}^T$ then write (c, X) on the database $\mathcal{D}$, delete the entry from $\mathcal{D}^T$, and increase the counter c.
4. Upon message $(\mathtt{read}, c)$ on the inport $\mathtt{BB.in}_i$ send the message $(\mathtt{read}, i, c)$ to the outport $\mathtt{BB.lk}$.
5. Upon message $(\mathtt{read}, i, c)$ on the inport $\mathtt{BB.infl}$, read from the database $\mathcal{D}$ the tuple (p, i, x) and if it exists write $(\mathtt{read}, (p, i, x))$ to the outport $\mathtt{BB.out}_i$ and to $\mathtt{BB.lk}$.

Special commands:

6. Upon message $(\mathtt{write}, \mathtt{F}, X)$ on the inport $\mathtt{BB.in}_{\mathtt{F}}$ write (c, X) on the database $\mathcal{D}^I$ and increase the counter c^I.
7. Upon message $\mathtt{read}$ on the inport $\mathtt{BB.in}_{\mathtt{A}}$, send the message $(\mathtt{read}, \mathcal{D}, \mathcal{D}^I)$ to the outport $\mathtt{BB.out}_{\mathtt{A}}$.

Agent $\mathcal{P}_A[\mathsf{Audit}]$:

The agent has two port $\mathtt{BB.in}_{\mathtt{A}}$ and $\mathtt{BB.out}_{\mathtt{A}}$ connected to the BB. Moreover, a protocol identifier pid is assigned to the agent.

1. Upon activation, if the message $\mathtt{corrupt}$ appears in the influence port then ignore it;
2. Read the message $(\mathtt{input}, X)$ from the influence port, send the message $\mathtt{read}$ to the outport $\mathtt{BB.in}_{\mathtt{A}}$ and at next activation read the message $(\mathtt{read}, \mathcal{D}, \mathcal{D}^I)$ from $\mathtt{BB.out}_{\mathtt{A}}$. Let $\mathcal{D}^R$ be the set $\{(c, i, x) : p = \mathsf{pid}, (c, p, i, x) \in \mathcal{D}\}$ and let $\tau \leftarrow (\mathcal{D}^R, \mathcal{D}^I)$.
3. If for all $i \in [n]$ there exists $(*, i, \mathtt{endProtocol})$ then compute $b \leftarrow \mathsf{Audit}(\tau, X)$ else return the message $\mathtt{error}$ to the port port $\mathtt{A.lk}$ and return;
4. Return the message $(\mathtt{audit}, b)$ to the port $\mathtt{A.lk}$.

Fig. 10: The wrapper for auditable ideal functionalities, the ideal functionality for Bulletin Board, and the auditor party.

Now for any $\mathcal{Z}$ we can also consider a new environment $\mathcal{Z}' := \mathcal{Z} \circ \mathcal{W}[\mathsf{Sim}_{\mathtt{F}}] \circ \mathcal{P}_{\tilde{A}}$. We can see that $\mathcal{Z}'$ is in the class of environment for the protocol for the functionality $\mathcal{G}$. Rewriting the last interactive system in the equations above we can apply the fact that $(\Pi_{\mathtt{G}}, \mathsf{Audit}_{\mathtt{G}})$ is a secure auditable protocol and therefore:

$$\mathcal{W}[\mathsf{Sim}_{\mathtt{F}}] \circ \mathcal{W}[\mathsf{Sim}_{\mathtt{G}}] \circ \mathcal{W}[\mathcal{G}] \circ \mathrm{BB} \circ \mathcal{P}_{\tilde{A}} \circ \mathcal{P}_A[\mathsf{Audit}_{\mathtt{G}}] \approx_c \mathcal{W}[\mathsf{Sim}_{\mathtt{F}}] \circ \mathcal{W}[\mathsf{Sim}_{\mathtt{G}}] \circ \mathcal{W}[\mathcal{G}] \circ \mathrm{BB} \circ \mathcal{P}_{\tilde{A}} \circ \mathcal{P}_A[\mathsf{Audit}^*_{\mathtt{G}}].$$

Using again Eq. (6) we can rewrite the above equation, then we can use the fact that $(\Pi_{\mathtt{F}}, \mathsf{Audit}_{\mathtt{F}})$ is secure auditable:

$$\begin{aligned}\mathcal{W}[\mathsf{Sim}_{\mathtt{F}} \circ \mathsf{Sim}_{\mathtt{G}}] \circ \mathcal{W}[\mathcal{G}] \circ \mathrm{BB} \circ \mathcal{P}_{\tilde{A}} \circ \mathcal{P}_A[\mathsf{Audit}^*_{\mathtt{G}}] \equiv \\ \mathcal{W}[\mathsf{Sim}_{\mathtt{F}} \circ \mathsf{Sim}_{\mathtt{G}}] \circ \mathcal{W}[\mathcal{G}] \circ \mathrm{BB} \circ \mathcal{P}_A[\mathsf{Audit}_{\mathtt{F}}] \approx_c \\ \mathcal{W}[\mathsf{Sim}_{\mathtt{F}} \circ \mathsf{Sim}_{\mathtt{G}}] \circ \mathcal{W}[\mathcal{G}] \circ \mathrm{BB} \circ \mathcal{P}_A[\mathsf{Audit}^*_{\mathtt{F}}].\end{aligned}$$

This concludes the proof.

F Verifiable Threshold Decryption in the Random String model

We show how we can UC-auditable realize the ideal functionalities for CRS generation and threshold decryption that are needed by our Mix-Net protocol. In particular, we give a construction that works for the case when the PKE is our pv-RAND-RCCA PKE scheme. We begin by giving the definitions of the ideal functionalities.

Functionality $\mathcal{F}_{\mathsf{TDec}}[\mathcal{PKE}, \mathtt{prm}]$:

let $\mathcal{PKE} = (\mathsf{Init}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$.

Initialization Phase: at the first activation sample $\mathsf{pk}, \mathsf{sk} \leftarrow\!\!\$\ \mathsf{KGen}(\mathtt{prm})$ and store the tuple $(\mathsf{pk}, \mathsf{sk})$;
Public Value: on message $\mathtt{pk}$ from a party $\mathcal{P}_{m_i}$, $i \in [m]$, (resp. the adversary A) send $\mathtt{pk}$ to $\mathcal{P}_{M_i}$ (resp. the adversary A).
Decryption Value: on message $(\mathtt{dec}, \mathtt{C})$ from party $\mathcal{P}_{m_i}$, $i \in [m]$, check that the tuple $(\mathtt{C}, \mathtt{M}, \mathcal{I})$ exists in the database, if so update $\mathcal{I}$ including the index i else create the new entry $(\mathtt{C}, \mathsf{Dec}(\mathsf{sk}, \mathtt{C}), \{i\})$ in the database. if $|\mathcal{I}|$ equals m then send a public delayed output $(\mathtt{dec}, \mathtt{C}, \mathtt{M})$ to the parties $\mathcal{P}_{M_i}$ for $i \in [m]$;

Functionality $\mathcal{F}_{\mathsf{CRS}}[\mathsf{Init}, \mathtt{prm}]$:

Initialization Phase: at the first activation sample $\mathsf{crs} \leftarrow\!\!\$\ \mathsf{Init}(\mathtt{prm})$ and store it;
Public Value: on message $\mathtt{crs}$ from a party $\mathcal{P}_{M_i}$, $i \in [m]$, (resp. the adversary A) send crs to $\mathcal{P}_{M_i}$ (resp. the adversary A).

Fig. 11: Ideal Functionalities for Threshold Decryption and Common Reference String parametrized by group parameters $\mathtt{prm}$ and by a PKE scheme $\mathcal{PKE}$ and a NIZK setup Init respectively.

Building Blocks. Let $\mathcal{PKE}' = (\mathsf{KGen}', \mathsf{Enc}', \mathsf{Dec}')$ be the inner Rand-CPA PKE scheme derived from our pv-Rand-RCCA scheme $\mathcal{PKE}_2$. In particular the algorithm $\mathsf{KGen}'(\mathtt{prm})$ produces $[\mathbf{D}^*]_1, \mathbf{a}$ such that $(\mathbf{D}^*)^\top = \mathbf{D}^\top \| \mathbf{D}\mathbf{a}$, the encryption algorithm Enc' on input $[\mathbf{D}^*]_1$ and $[\mathtt{M}]_1$ output $[\mathbf{x}]_1$, and the decryption algorithm Dec' on input $\mathbf{a}$ and $[\mathbf{x}]_1$ outputs $(-\mathbf{a}^\top, 1)[\mathbf{x}]_1$. The building blocks are:

1. An ideal functionality for threshold decryption of $\mathcal{PKE}'$. Such functionality could be UC-realized easily by an auditable protocol in the random string model with static corruption. Notice that this is a generalization of a standard ideal functionality for threshold decryption for the ElGamal scheme. Since this is straightforward we only sketch the idea. The mixers at initialization phase compute an additive secret sharing $[\mathbf{a}_i]_i$ of the secret key $\mathbf{a}$; next, to decrypt a ciphertext $[\mathbf{x}]_1$, the mixers compute a decryption share $(-\mathbf{a}_i^\top, 1)[\mathbf{x}]_1$ together with a NIZK proof of its consistency.
2. An equivocable commitment $\mathcal{COMM} = (\mathsf{Init}, \mathsf{Com}, \mathsf{ECom}, \mathsf{EOpen}, \mathsf{EInit})$ for matrices in $\mathbb{G}_1$ and in $\mathbb{G}_2$ in the random-string model. The scheme is perfectly binding when the reference string is generated with Init and equivocable when the common referen string in generated with EInit. Moreover, the CRS generated in binding mode is computationally indistinguishable from an uniformly random string.
3. An adaptive sound Groth-Sahai (GS) NIZK $\mathcal{NIZK}$ for the knowledge of an opening for the commitment $\mathcal{COMM}$. More in details the relation proved is $\mathcal{R}_{\mathcal{COMM}} = \{(\mathsf{crs}^C, \mathbf{c}), ([\mathtt{M}]_i, \mathtt{r}) : = \mathsf{Com}(\mathsf{crs}^C, [\mathtt{M}]_i, \mathtt{r})\}$, the proof system needs only to prove knowledge of $[\mathtt{M}]_i$, where $i \in \{1, 2\}$.
4. An ideal functionality for the common-reference string, which generates m different common reference strings $\mathsf{crs}_1, \ldots, \mathsf{crs}_m$ for the GS proof system, m different common reference strings $\mathsf{crs}_2^C, \ldots, \mathsf{crs}_m^C$ for the commitment scheme, and another crs crs for the malleable proof system described in Sec. 4 for the pv-Rand PKE scheme. We stress that in the real protocol the CRSs are sampled as uniformly random strings.

We define the auditable protocol Π_{TDec} that realizes $\mathcal{F}_{\mathsf{TDec}}[\mathcal{PKE}_2, \mathtt{prm}]$ in the $(\mathcal{F}_{\mathsf{TDec}}[\mathcal{PKE}', \mathtt{prm}], \mathcal{F}_{\mathsf{CRS}})$-hybrid model. The idea of the protocol is rather simple and relies on the observation that the public-secret key pair of our scheme can be constructed with an additive secret sharing using its linear properties.

The mixer $\mathcal{P}_{M_j}$ does:

– **Initialization Phase:** upon first activation, it receives $[\mathbf{D}^*]$ from $\mathcal{F}_{\mathsf{TDec}}[\mathcal{PKE}']$ and $\{\mathsf{crs}_i, \mathsf{crs}_i^C\}_{j=1,\ldots,m}$ from $\mathcal{F}_{\mathsf{CRS}}$, and proceeds as described below:

1. Sample $\mathbf{f}_i, \mathbf{g}_i \leftarrow_\$ \mathbb{Z}_q^k$, and $\mathbf{F}_i \leftarrow_\$ \mathbb{Z}_q^{k+1\times k+1}, \mathbf{G}_i \leftarrow_\$ \mathbb{Z}_q^{k+1\times k+2}$;
2. Compute $[\mathbf{f}_i^\top \mathbf{D}]_1, [\mathbf{F}_i^\top \mathbf{D}]_1, [\mathbf{g}_i^\top \mathbf{E}]_2, [\mathbf{G}_i^\top \mathbf{E}]_2$ and the values $[\mathbf{G}\mathbf{D}^*]_1, [\mathbf{F}\mathbf{E}]_2$; We call pk_i be the concatenation of all these values; We notice that the values $\mathbf{f}_i, \mathbf{F}_i, \mathbf{g}_i, \mathbf{G}_i$ at this point can be deleted, as they are not necessary for decryption.
3. Commit the value pk_i, let $\mathbf{c}_i$ be such commitment and $\mathtt{r}_i$ the opening information,
4. Let $\pi_i \leftarrow \mathsf{P}(\mathsf{crs}_i, (\mathsf{crs}_i^C, \mathbf{c}_i), (\mathsf{pk}_i, \mathtt{r}_i))$ be a NIZK proofs of the knowledge of the opening of the commitment $\mathbf{c}_i$;
5. Post in the bulletin board the message $(i, \mathbf{c}_i, \pi_i)$.

At next activation, if all the mixers have sent their value and all the NIZK proofs verify then post in the bulletin board the opening $(i, \mathtt{r}_i)$ and wait that all the mixers do the same, the do as described below:

6. For any $j \in [m], j \neq i$, parse $\mathsf{pk}_j = ([\tilde{\mathbf{f}}_j]_1, [\tilde{\mathbf{F}}_j]_1, [\tilde{\mathbf{g}}_j]_2, [\tilde{\mathbf{G}}_j]_2, [\tilde{\mathbf{H}}_j]_1, [\tilde{\mathbf{I}}_j]_2)$, and check that:

$$\begin{aligned} e([\tilde{\mathbf{H}}_j]_1, [\mathbf{E}]_2) &= e([\mathbf{D}]_1, [\tilde{\mathbf{G}}_j]_2) \\ e([\mathbf{D}]_1, [\tilde{\mathbf{I}}_j]_2) &= e([\tilde{\mathbf{F}}_j]_1, [\mathbf{E}]_2) \end{aligned} \tag{7}$$

Notice that if such conditions hold then the matrices have the right form.

7. Compute the public key pk as:

$$\begin{aligned}&\mathsf{crs}, [\mathbf{D}], [\mathbf{E}], [\mathbf{D}^*],\\ &\textstyle\sum_j [\mathbf{f}_j^\top \mathbf{D}]_1, \sum_j [\mathbf{F}_j^\top \mathbf{D}]_1, \sum_j [\mathbf{g}_j^\top \mathbf{E}]_2, \sum_j [\mathbf{G}_j^\top \mathbf{E}]_2,\\ &\textstyle\sum_j [\mathbf{G}_j \mathbf{D}^*]_1, \sum_j [\mathbf{F}_j \mathbf{E}]_2\end{aligned}$$

– **Decrypt Value** On message $(\mathtt{dec}, \mathtt{C})$ the party $\mathcal{P}_{M_i}$, check if $\mathsf{V}(\mathsf{pk}, \mathtt{C}) = 1$ and if so parses $\mathtt{C}$ as $([\mathbf{x}]_1, [\mathbf{v}]_2, \Pi)$ and forwards $(\mathtt{dec}, \mathtt{C})$ to $\mathcal{F}_{\mathsf{TDec}}[\mathcal{PKE}', \mathtt{prm}]$, when the functionality answers, forward the answer as its own output.

Audit algorithm: Audit verifies the NIZK proofs $\pi_1, \ldots, \pi_m$ and checks the equations (7).

Theorem 9. *The auditable protocol* $(\Pi_{TDec}, \mathsf{Audit})$ $\mathsf{UC\text{-}auditable}$ $\mathsf{realizes}$ *the functionality* $\mathcal{F}_{\mathsf{TDec}}[\mathcal{PKE}]$.

Proof (Sketch). In this sketch we simply define the simulator S. The indistinguishability of the ideal and real world are straightforward.

Ideal adversary S:

Initialization received the set of corrupted party $\mathcal{C}$, let h^* the index of an honest party. Sample all the common reference string in perfectly binding mode with exception of $\mathsf{crs}^C_{h^*}, \mathsf{crs}_{h^*}$ that are sampled in trapdoor mode. Receive from the ideal functionality $\mathcal{F}_{\mathsf{TDec}}[\mathcal{PKE}]$ the public key pk.

Extraction extract from the NIZK proofs of the corrupted parties obtaining the values:

$$\mathsf{pk}_j = [\tilde{\mathbf{f}}_j]_1, [\tilde{\mathbf{F}}_j]_1, [\tilde{\mathbf{g}}_j]_2, [\tilde{\mathbf{G}}_j]_2, [\tilde{\mathbf{H}}_j]_1, [\tilde{\mathbf{I}}_j]_2$$

Equivocation the party $\mathcal{P}_{P_{h^*}}$ sends a fake commitment $\tilde{\mathbf{c}}_{h^*}$ together with a simulated proof $\tilde{\pi}_{h^*}$; then if the checks in the equation (7) hold for the pk_j where $j \in \mathcal{C}$ then it equivocates the commitment $\tilde{\mathbf{c}}_{h^*}$ to:

$$\mathsf{pk} - \sum_{i \neq h^*} \mathsf{pk}_j$$

else it opens the commitment to a validly generated share pk_{h^*}.

Decrypt Value the simulation of this part is trivial, the simulator follows exactly the protocol simulating the ideal functionality $\mathcal{F}_{\mathsf{TDec}}[\mathcal{PKE}']$ thanks to the outputs of $\mathcal{F}_{\mathsf{TDec}}[\mathcal{PKE}]$.

G Definitions

G.1 All-but-One label-based NIZK systems

Let $\mathcal{NIZK} = (\mathsf{Init}, \mathsf{P}, \mathsf{V})$ be a NIZK proof system with label space $\mathcal{L}$, and let $\mathsf{ABOInit}(\mathtt{prm}, \tau)$ an algorithm that upon input $\mathtt{prm}$ and a label $\tau \in \mathcal{L}$ outputs a common reference string and trapdoor information tp_e, tp_s.

We require the following property from $\mathsf{ABOInit}$:

All-but-One Composable Zero-Knowledge. For any $\tau \in \mathcal{L}$ the common reference string generated by $\mathsf{Init}(\mathtt{prm})$ and by $\mathsf{ABOInit}(\mathtt{prm}, \tau)$ are computationally indistinguishable. Moreover, for any τ', x, w where $\tau \neq \tau'$ the distributions $\mathsf{P}(\mathsf{crs}, \tau', x, w)$ and $\mathsf{Sim}(tp_s, \tau', x, w)$ are equivalently distributed.

All-but-One Adaptive Perfect Soundness. There exists an extractor Ext such that any $\mathsf{crs}, tp_s, tp_e \leftarrow \mathsf{ABOInit}(\mathtt{prm}, \tau)$ for any (possibly unbounded) adversary $(\tau, x, \pi) \leftarrow \mathsf{A}(\mathsf{crs})$ such that $\mathsf{V}(\tau, x, \pi) = 1$ then $\mathsf{Ext}(tp_e, \tau, x, \pi)$ outputs w such that $(x, w) \in \mathcal{R}$.

Construction. Consider the instatiantion of GS Proof system of [16] based on $\mathcal{D}_k$-MDDH. The common reference string is of the following two forms:

$$\begin{aligned} [\mathbf{A} \| \mathbf{A}\mathbf{w}] &\qquad \text{Perfect Sound Mode} \\ [\mathbf{A} \| \mathbf{A}\mathbf{w} - \mathbf{z}] &\qquad \text{Perfect Hiding Mode} \end{aligned}$$

where $\mathbf{A} \leftarrow_\$ \mathcal{D}_k$, $\mathbf{w} \leftarrow_\$ \mathbb{Z}_q^k$ and $\mathbf{z} \notin span(\mathbf{A})$ is a fixed and public vector. We can consider a NIZK with label where the common reference string is made by two independent CRSs $\mathsf{crs}_1, \mathsf{crs}_2$, both the verifier and the prover on input a label $\tau \in \mathbb{Z}_q$ derive a CRS $\mathsf{crs}_\tau = \mathsf{crs}_1 + \mathsf{crs}_1 \cdot \tau$. We are ready to define the $\mathsf{ABOInit}$.

$\underline{\mathsf{ABOInit}(\mathtt{prm}, \tau^*)}$:

1. Sample $\mathbf{A}_1, \mathbf{A}_2$ and $\mathbf{w}_1, \mathbf{w}_2$ and set $\mathsf{crs}'_1 = (\mathbf{A} \| \mathbf{A}\mathbf{w}_1)$ and $\mathsf{crs}'_2 = (\mathbf{A}_2 \| \mathbf{w}_2 - \mathbf{z})$;
2. Set $\mathsf{crs}_1 = \mathsf{crs}'_1 - \mathsf{crs}'_2 \cdot \tau^*$ and $\mathsf{crs}_2 = \mathsf{crs}'_2$;
3. Output $\mathsf{crs}_1, \mathsf{crs}_2$.

The all-but-one composable zero-knowledge comes readily from the $\mathcal{D}_k$-MDDH assumption and the composable zero-knowledge of GS proofs. The all-but-one adaptive perfect soundness comes readily from the adaptive perfect soundness of GS proofs, in fact we notice that $\mathsf{crs}_{\tau^*} = \mathsf{crs}'_1 - \tau^* \mathsf{crs}'_2 + \tau^* \mathsf{crs}'_2 = \mathsf{crs}'_1$ which allows for perfectly sound proofs.

G.2 Additional Definitions for Malleable NIZK

Definition 19 (Tightness for NIZK). *We say that a $\mathcal{NIZK}$ has* tight proofs *if for any (possibly unbounded) adversary* A *the following holds:*

$$\Pr\left[\pi \notin \mathsf{P}(\mathsf{crs}, x, w) \wedge \mathsf{V}(\mathsf{crs}, x, \pi) = 1 : \begin{array}{c} \mathtt{prm} \leftarrow \mathsf{Setup}(1^\lambda) \\ (\mathsf{crs}, tp_e, tp_s) \leftarrow_\$ \mathsf{Init}(\mathtt{prm}) \\ (x, w, \pi) \leftarrow \mathsf{A}(\mathsf{crs}) \end{array}\right] \in \mathtt{negl}(\lambda).$$