Fine-grained Verifier NIZK and Its Applications

Xiangyu Liu[1,2], Shengli Liu[1,2,3(✉)], Shuai Han[1,2(✉)], and Dawu Gu[1]

[1] School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
{xiangyu_liu,slliu,dalen17,dwgu}@sjtu.edu.cn
[2] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[3] Westone Cryptologic Research Center, Beijing 100070, China

Abstract. In this paper, we propose a new type of non-interactive zero-knowledge (NIZK), called *Fine-grained Verifier NIZK (FV-NIZK)*, which provides more flexible and more fine-grained verifiability of proofs than standard NIZK that supports public verifiability and designated-verifier NIZK (DV-NIZK) that supports private verifiability. FV-NIZK has two statistically equivalent verification approaches:

- a master verification using the master secret key msk;
- a fine-grained verification using a derived secret key sk_d, which is derived from msk w.r.t. d (which may stand for user identity, email address, vector, etc.).

We require *unbounded simulation soundness (USS)* of FV-NIZK to hold, even if an adversary obtains derived secret keys sk_d with d of its choices, and define *proof pseudorandomness* which stipulates the pseudorandomness of proofs for adversaries that are not given any secret key.

We present two instantiations of FV-NIZK for linear subspace languages, based on the matrix decisional Diffie-Hellman (MDDH) assumption. One of the FV-NIZK instantiations is *pairing-free* and achieves almost tight USS and proof pseudorandomness.

We illustrate the usefulness of FV-NIZK by showing two applications and obtain the following pairing-free schemes:

- the *first* almost tightly multi-challenge CCA (mCCA)-secure inner-product functional encryption (IPFE) scheme *without pairings*;
- the *first* public-key encryption (PKE) scheme that reconciles the inherent contradictions between public verifiability and anonymity. We formalize such PKE as *Fine-grained Verifiable PKE (FV-PKE)*, which derives a special key from the decryption secret key, such that for those who obtain the derived key, they can check the validity of ciphertexts but the anonymity is lost from their views (CCA-security still holds for them), while for others who do not get the derived key, they cannot do the validity check but the anonymity holds for them.

  Our FV-PKE scheme achieves almost tight mCCA-security for adversaries who obtain the derived keys, and achieves almost tight ciphertext pseudorandomness (thus anonymity) for others who do not get any derived key.

1 Introduction

NIZK with Unbounded Simulation Soundness (USS). Over decades, non-interactive zero-knowledge (NIZK) proofs have shown great power in constructing a variety of cryptographic primitives, e.g., public-key encryption (PKE) [27, 14], digital signatures [7], etc. Towards better efficiency and shorter proofs, Jutla and Roy [23] defined a weaker notion called *quasi-adaptive* NIZK (QA-NIZK), where the common reference string (CRS) might depend on the specific language. In this paper, we will focus on quasi-adaptive NIZK and omit the term "quasi-adaptive" for simplicity.

One important security property for NIZK is *unbounded simulation soundness* (USS) [30, 25], which plays an important role in many applications of NIZK, e.g., CCA-secure PKE [21, 16], publicly verifiable CCA identity-based encryption (IBE) [22], structure preserving signatures [5, 4], etc. Loosely speaking, USS requires the computational hardness for an adversary to generate a valid proof for an instance outside the language, even if the adversary has access to an oracle that outputs simulated proofs for instances (not necessarily in the language) of its choices.

Tight Security and NIZK with Tight USS. The security of a cryptographic primitive is usually proved via a reduction, which turns an adversary $\mathcal{A}$ that breaks the security of the primitive with running time t and advantage ϵ into an algorithm $\mathcal{B}$ that solves some hard problem with running time $t' \approx t$ and advantage ϵ'. Intuitively, we would desire ϵ' to be as large as ϵ. To reflect this, we define $L := \epsilon/\epsilon'$ as the security loss factor, which is the smaller the better. We call the reduction *tight* if L is a small constant or *almost tight* if L is linear (or even better, logarithmic) in the security parameter λ. For a loose reduction, L usually depends on $\mathcal{A}$'s behaviours, e.g., the number of $\mathcal{A}$'s queries, which can be as large as 2^{50} in practical settings.

Pursuing (almost) tight security has both theoretical and practical significance. For a scheme with a loose security reduction, the deployer has to choose larger security parameters to compensate the security loss, resulting in larger elements and lower efficiency. In contrast, schemes with (almost) tight security enjoy many advantages like universal key recommendations and more flexible choices of parameters. Recently, (almost) tight security has been explored in many areas, including PKE [21, 16, 17, 20], signatures [21, 24, 8, 19], IBE [11, 9], etc.

In the scenario of NIZK, Libert et al. [25] proposed the first scheme with (almost) tight USS, and Gay et al. [16] gave a more efficient construction later. In both schemes, the size of the CRS (in terms of the number of group elements) is linear in λ. The first (almost) tightly secure NIZK with constant-size CRS was designed by Abe et al. [5]. Recently in [4], Abe et al. proposed a shorter NIZK with both constant-size CRS and proofs.

Designated-Verifier NIZK (DV-NIZK). Standard NIZK allows *public verification*, so that anyone who gets the CRS can verify the validity of proofs. Such a property is useful in certain applications, e.g., when constructing signature

schemes [7, 4], the public verifiability of signatures requires the public verifiability of NIZK proofs. However, in some other applications such as constructing CCA-secure PKE [12, 16], public verification is not necessary, and in fact, a *designated-verifier* NIZK (DV-NIZK) [16] that supports only private verification of proofs is sufficient. Roughly speaking, DV-NIZK is the same as NIZK except that, the verification algorithm additionally takes a secret key sk as input, so that only the designated verifier can check the validity of proofs. Moreover, the secret key should be kept private, since otherwise the (simulation) soundness might not hold any more.

Compared to NIZK, DV-NIZK usually has more succinct and more efficient constructions, since it is only required to support private verification. For example, the efficient hash proof systems (HPS) in [12] can be viewed as DV-NIZKs. As another example, to the best of our knowledge, all NIZK schemes with tight USS (constructed in discrete-logarithm setting) relies on bilinear pairings to support public verification [25, 16, 5, 4], while DV-NIZK with tight USS can be constructed without pairings [16].

However, both NIZK (that supports public verification) and DV-NIZK (that supports private verification) have their limitations on the flexibility of verification in certain applications. We demonstrate with two examples below.

Fine-grained Verification Setting in IPFE. Inner-product functional encryption (IPFE) [1] is a special subclass of functional encryption [28, 10] for inner-product functions. In an IPFE scheme, a ciphertext is an encryption of a vector $\mathbf{x} \in \mathbb{Z}^m$, a secret key $\widetilde{sk}_{\mathbf{y}}$ (delegated from the master secret key $\widetilde{msk}$) is related with a vector $\mathbf{y} \in \mathbb{Z}^m$, and the decryption just returns their inner product $\langle \mathbf{x}, \mathbf{y} \rangle$. The inner-product function supports a large set of computation formulas, ranging from conjunctions and disjunctions to descriptive statistics and polynomial evaluations.

There are many explorations of CPA-secure IPFE schemes over the past years, e.g., [6, 2, 31]. All ciphertexts in these constructions fall into the HPS paradigm [12] with a pattern (c, v), where c is an instance in a language specified by the public key and v masks the message m.

To lift these CPA-secure IPFE schemes to CCA-secure IPFE schemes, one may want to resort to NIZK or DV-NIZK to reject ill-formed ciphertexts (i.e., ciphertexts with c outside the language) in decryption, thus making the decryption oracle useless to the adversary. This can be done by adding a NIZK/DV-NIZK proof in the ciphertext to prove that c belongs to the language. However, here comes the dilemma when choosing a suitable NIZK argument:

- DV-NIZK does not work in this setting with the following reason. To verify the well-formedness of ciphertexts, the decryption algorithm of IPFE has to know the secret key sk of DV-NIZK to verify the DV-NIZK proofs in ciphertexts. Thus all secret keys $\widetilde{sk}_{\mathbf{y}}$ of IPFE should contain the secret key sk. However, note that an adversary in the CPA/CCA-security experiment of IPFE is free to ask $\widetilde{sk}_{\mathbf{y}}$ for vectors $\mathbf{y}$ of its choices. Consequently, the adversary only needs to ask a single $\widetilde{sk}_{\mathbf{y}}$ to know the secret key sk of DV-

NIZK, in which case the (simulation) soundness of DV-NIZK might not hold any more, and consequently, the CCA-security of IPFE might not hold.

– In contrast, NIZK with public verification is sufficient, but seems to be overqualified in this setting. In fact, it is not necessary for everyone, but only those who hold secret keys $\widetilde{sk}_{\mathbf{y}}$, to be able to check the well-formedness of ciphertexts in decryption.

In summary, DV-NIZK does not work in converting CPA-secure IPFE schemes into CCA-secure ones but it has more efficient constructions (e.g., pairing-free constructions), while NIZK is sufficient but at the price of heavy constructions (especially, the pairing operations) and it seems to be overqualified.

Actually, what we need is a NIZK with *fine-grained verifiability*, lying between public verifiability and private verifiability. More precisely, there is a master secret key msk for verification, and the ability of verification can be delegated via deriving different secret keys sk_d from msk w.r.t. different d (which stands for, e.g., user identity, email address, vector, etc.), so that one can use sk_d to do the verification of NIZK proofs (hence execute decryptions of IPFE). On the one hand, all these verification approaches, no matter using msk or using sk_d w.r.t. any d, are statistically equivalent. On the other hand, (simulation) soundness is guaranteed even if the adversary obtains several sk_d with d chosen by itself, as long as msk is not leaked to the adversary.

In this work, we will formalize such NIZK as *Fine-grained Verifier NIZK (FV-NIZK)*, and show that it is sufficient for lifting CPA-secure IPFE schemes to CCA-secure ones. FV-NIZK has pairing-free constructions, and hence solves the aforementioned dilemma.

Fine-grained Verification Setting in PKE. In traditional PKE setting, only the owner of the secret key sk can check the validity of a ciphertext (i.e., whether a ciphertext decrypts to some plaintext or the decryption fails). In some applications, it is desirable to outsource this validity check to others. For example, a manager may ask an assistant to filter out invalid ciphertexts for her/him so that the manager can decrypt only the valid ciphertexts herself/himself, but the manager does not want to reveal the secret key to the assistant. To solve such problems, the concept of *publicly verifiable* PKE (PV-PKE) [3, 21] is developed, in which anyone can check the validity of a ciphertext with only the public key of the owner.

Though public verifiability is desirable in some scenarios, it also brings the disadvantage of *losing anonymity*. Namely, anyone can identify the intended receiver of a ciphertext, by just doing a verification under someone's public key.

In order to reconcile the inherent contradictions between public verifiability and anonymity, we put forward a new primitive called *Fine-grained Verifiable PKE (FV-PKE)*, which can derive a special key (for validity check of ciphertexts) from the secret key (for decryption). Roughly speaking, with the derived key, one can check the validity of ciphertexts but cannot decrypt the ciphertexts, while without the key, the anonymity of ciphertexts holds. Let us move back to the above example. Now the manager can safely give this derived key to the

assistant to filter out invalid ciphertexts. For the assistant, the anonymity is lost but the CCA-security of the PKE still holds. For others who only obtain the public key of the manager, the anonymity of ciphertexts holds. Furthermore, we allow that different keys (for validity check) can be derived from the secret key (for decryption), to achieve fine-grained verifiability.

Now we consider how to construct FV-PKE. Let us start from any CPA-secure PKE scheme. To lift it to CCA-secure FV-PKE, one may want to resort to NIZK (as in [27, 14]) or DV-NIZK (as in [12, 16]) to reject ill-formed ciphertexts. However, neither NIZK nor DV-NIZK leads to FV-PKE:

- DV-NIZK does not support the delegation of verifiability. Thus to check the validity of ciphertexts, the derived key of PKE should contain the secret key of DV-NIZK. Then for anyone with the derived key (e.g., the assistant in the above example), the (simulation) soundness of DV-NIZK might not hold, and consequently, the CCA-security of PKE might not hold.
- NIZK allows public verification of proofs. Thus anyone (who obtains the CRS of NIZK from the public key of PKE[4]) can check the validity of ciphertexts, and consequently the anonymity of PKE is sacrificed. Even in the setting that all users of a group (e.g., a company or a college) share the same CRS, the identity of the group is still leaked.

In fact, our new *Fine-grained Verifier NIZK (FV-NIZK)* is suitable in this setting and can successfully convert a CPA-secure PKE into a CCA-secure FV-PKE. More precisely, the owner can derive an sk_d from the master secret key msk of FV-NIZK, so that sk_d can be used to do validity check of ciphertexts. Meanwhile, obtaining sk_d does not compromise the (simulation) soundness of FV-NIZK, and hence CCA-security of PKE holds, even for those who have the derived key. Furthermore, for others who do not obtain the derived key, the anonymity of PKE holds, as long as the underlying CPA-secure PKE is anonymous and FV-NIZK has pseudorandom proofs.

Our Contributions. Now we summarize our contributions in this paper. We introduce a new primitive called *Fine-grained Verifier NIZK (FV-NIZK)*, which provides more flexible and more fine-grained verifiability than standard NIZK (with public verifiability) and DV-NIZK (with private verifiability). Intuitively, FV-NIZK has two main verification approaches:

- a master verification (MVer) using the master secret key msk;
- a fine-grained verification (FVer) using a derived secret key sk_d, which is derived from msk w.r.t. $d \in \mathcal{D}$. Here d belongs to a delegation space $\mathcal{D}$, and may stand for user identity, email address, vector, etc.

We equip FV-NIZK with a set of useful security properties. The statistical *verification equivalence* property requires that the two verification approaches, no

[4] Note that the CRS of NIZK is contained in the public key of PKE, since the encryption algorithm of PKE involves NIZK proof generation which requires the CRS.

matter using msk or using sk_d w.r.t. any $d \in \mathcal{D}$, are statistically equivalent. Besides, we adapt *unbounded simulation soundness (USS)* to FV-NIZK, by additionally allowing the adversary to obtain derived secret keys sk_d with d of its choices. We also define *proof pseudorandomness* which stipulates the pseudorandomness of proofs for adversaries that are not given any secret key.

Then we propose two instantiations of FV-NIZK with almost tight USS for linear subspace languages, based on the matrix decisional Diffie-Hellman (MDDH) assumption [15] (which covers the standard DDH and k-Linear assumptions).

- Our first instantiation is inspired by the DV-NIZK scheme constructed in [16]. The resulting FV-NIZK is *pairing-free*, and achieves almost tight USS and proof pseudorandomness, with a linear loss factor $L = O(\lambda)$.
- Our second instantiation is inspired by the DV-NIZK and NIZK schemes in [4]. The resulting FV-NIZK is pairing-based, but involves *less pairing operations* than the NIZK scheme in [4]. It achieves almost tight USS with a loss factor $L = O(\log \lambda)$, logarithmic in the security parameter λ.

Finally, we illustrate the usefulness of FV-NIZK by showing two applications.

- The first application is in constructing CCA-secure IPFE. Using our FV-NIZK with almost tight USS as the core technique tool, we construct a tightly multi-challenge CCA (mCCA)-secure IPFE scheme from the almost tightly multi-challenge CPA (mCPA)-secure IPFE proposed in [31].

  By instantiating FV-NIZK, we obtain the first almost tightly mCCA-secure IPFE scheme *without pairings*, where the loss factor is $L = O(\lambda)$. We also obtain another almost tightly mCCA-secure IPFE scheme that uses less pairing operations than the only known scheme [26] (12 *vs.* $2m+16$ pairings, with m the vector dimension of IPFE), where the loss factor is $L = O(\log \lambda)$, the same as [26].
- The second application is in constructing *Fine-grained Verifiable PKE (FV-PKE)*. This is a new primitive formalized in this paper to reconcile the inherent contradictions between public verifiability and anonymity of PKE. Loosely speaking, FV-PKE derives a special key from the decryption secret key, such that for those who obtain the derived key, they can check the validity of ciphertexts but the anonymity is lost from their views (CCA-security still holds for them), while for others who do not get the derived key, they cannot do the validity check but the anonymity holds for them.

  By using our first FV-NIZK instantiation with almost tight USS and proof pseudorandomness as the core building block, we construct the first FV-PKE scheme that achieves both almost tight mCCA-security and almost tight ciphertext pseudorandomness (thus anonymity). Moreover, the FV-PKE scheme is pairing-free.

Technical Overview of Our FV-NIZK Instantiations. Below we give a high-level overview of our FV-NIZK instantiations from the MDDH assumption.

Let $\mathbb{G}$ be a cyclic group of order q with generator g. For a matrix $\mathbf{A} := (a_{ij}) \in \mathbb{Z}_q^{n_1 \times n_2}$, we define $[\mathbf{A}] := (g^{a_{ij}}) \in \mathbb{G}^{n_1 \times n_2}$ as the implicit representation of $\mathbf{A}$ in $\mathbb{G}$ [15]. Our FV-NIZK instantiations are for linear subspace language $\mathscr{L}_{[\mathbf{A}]} := \mathsf{Span}([\mathbf{A}]) := \{[\mathbf{c}] \in \mathbb{G}^{n_1} \mid \exists \mathbf{s}\ s.t.\ \mathbf{c} = \mathbf{A}\mathbf{s}\}$ and the delegation space is $\mathcal{D} := \mathbb{Z}_q^m$.

Our starting point is the tag-based DV-NIZK scheme proposed by Gay et al. [16], which is pairing-free and has almost tight USS, as recalled below. The CRS is $\mathsf{crs} := ([\mathbf{k}^\top \mathbf{A}], [\mathbf{B}], \{[\widehat{\mathbf{k}}_{\ell,b}^\top \mathbf{B}]\}_{\ell,b})$, and the secret key msk for verification is $msk := (\mathbf{k}, \{\widehat{\mathbf{k}}_{\ell,b}\}_{\ell,b})$, where $\mathbf{k} \xleftarrow{\$} \mathbb{Z}_q^{n_1}$, $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{3k \times k}$ and $\widehat{\mathbf{k}}_{\ell,b} \xleftarrow{\$} \mathbb{Z}_q^{3k}$ for $1 \le \ell \le \lambda, b \in \{0,1\}$. With respect to a tag $\tau \in \{0,1\}^\lambda$, the proof of $[\mathbf{c}] = [\mathbf{A}]\mathbf{s} \in \mathscr{L}_{[\mathbf{A}]}$ is $\pi := ([\mathbf{t}], [u])$, where $[\mathbf{t}] := [\mathbf{B}]\mathbf{r}$ for $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_q^k$ and

$$[u] := [\mathbf{k}^\top \mathbf{A}]\mathbf{s} + [\widehat{\mathbf{k}}_\tau^\top \mathbf{B}]\mathbf{r}, \quad \text{with } \widehat{\mathbf{k}}_\tau := \textstyle\sum_{\ell=1}^{\lambda} \widehat{\mathbf{k}}_{\ell,\tau_\ell},$$

which can be verified via $[u] \stackrel{?}{=} \mathbf{k}^\top [\mathbf{c}] + \widehat{\mathbf{k}}_\tau^\top [\mathbf{t}]$ using msk.

How to derive keys for fine-grained verification? To support deriving keys for different delegations $\mathbf{d} \in \mathcal{D} = \mathbb{Z}_q^m$, a natural idea is to extend the master secret key in the DV-NIZK above from a set of vectors to a sets of matrices, i.e., $\mathsf{crs} := ([\mathbf{K}\mathbf{A}], [\mathbf{B}], \{[\widehat{\mathbf{K}}_{\ell,b}\mathbf{B}]\}_{\ell,b})$ and $msk := (\mathbf{K}, \{\widehat{\mathbf{K}}_{\ell,b}\}_{\ell,b})$ with $\mathbf{K} \xleftarrow{\$} \mathbb{Z}_q^{m \times n_1}$ and $\widehat{\mathbf{K}}_{\ell,b} \xleftarrow{\$} \mathbb{Z}_q^{m \times 3k}$. Accordingly, the proof is $\pi := ([\mathbf{t}], [\mathbf{u}])$ with

$$[\mathbf{u}] := [\mathbf{K}\mathbf{A}]\mathbf{s} + [\widehat{\mathbf{K}}_\tau \mathbf{B}]\mathbf{r}, \quad \text{with } \widehat{\mathbf{K}}_\tau := \textstyle\sum_{\ell=1}^{\lambda} \widehat{\mathbf{K}}_{\ell,\tau_\ell},$$

and the *master* verification checks $[\mathbf{u}] \stackrel{?}{=} \mathbf{K}[\mathbf{c}] + \widehat{\mathbf{K}}_\tau [\mathbf{t}]$ using msk. One can view it as m-parallel DV-NIZKs in [16].

Now we can derive a key $sk_{\mathbf{d}}$ w.r.t. a delegation $\mathbf{d} \in \mathcal{D} = \mathbb{Z}_q^m$ as follows

$$sk_{\mathbf{d}} := (\mathbf{d}, \mathbf{d}^\top \mathbf{K}, \{\mathbf{d}^\top \widehat{\mathbf{K}}_{\ell,b}\}_{\ell,b}),$$

and the *fine-grained* verification using $sk_{\mathbf{d}}$ checks

$$\mathbf{d}^\top [\mathbf{u}] \stackrel{?}{=} \mathbf{d}^\top \mathbf{K}[\mathbf{c}] + \mathbf{d}^\top \widehat{\mathbf{K}}_\tau [\mathbf{t}].$$

Intuitively, delegation algorithm for $\mathbf{d}$ derives a "projection" of msk on $\mathbf{d}$, so that this derived secret key can be used to check the proof on $\mathbf{d}$'s projection.

However, here come two problems. Firstly, the two verification approaches are not statistically equivalent. In fact, given only crs, an adversary $\mathcal{A}$ can easily produce a proof $\pi^* = ([\mathbf{t}^*], [\mathbf{u}^*])$ for $[\mathbf{c}]$ such that it passes the fine-grained verification w.r.t. $sk_{\mathbf{d}}$, but does not pass the master verification, i.e.,

$$\mathbf{d}^\top [\mathbf{u}^*] = \mathbf{d}^\top \mathbf{K}[\mathbf{c}] + \mathbf{d}^\top \widehat{\mathbf{K}}_\tau [\mathbf{t}^*], \text{ but } [\mathbf{u}^*] \neq \mathbf{K}[\mathbf{c}] + \widehat{\mathbf{K}}_\tau [\mathbf{t}^*].$$

This can be done as follows. $\mathcal{A}$ first generates a proof $\pi = ([\mathbf{t}], [\mathbf{u}])$ for an instance $[\mathbf{c}] \in \mathscr{L}_{[\mathbf{A}]}$ honestly using crs, and then chooses a pair of non-zero orthogonal vectors $\mathbf{d}, \mathbf{e} \in \mathbb{Z}_q^m$ s.t. $\mathbf{d}^\top \mathbf{e} = 0$, and sets $\pi^* = ([\mathbf{t}^*], [\mathbf{u}^*]) := ([\mathbf{t}], [\mathbf{u} + \mathbf{e}])$. Clearly

$[\mathbf{u}^*] - \mathbf{K}[\mathbf{c}] - \widehat{\mathbf{K}}_\tau[\mathbf{t}^*] = [\mathbf{u}^*] - [\mathbf{u}] = [\mathbf{e}] \neq [\mathbf{0}]$, but $\mathbf{d}^\top([\mathbf{u}^*] - \mathbf{K}[\mathbf{c}] - \widehat{\mathbf{K}}_\tau[\mathbf{t}^*]) = \mathbf{d}^\top[\mathbf{e}] = [0]$.

Moreover, USS cannot hold if an adversary $\mathcal{A}$ is allowed to obtain derived keys. Due to the linearity of $sk_{\mathbf{d}}$ in $\mathbf{d}$, each derived key $sk_{\mathbf{d}}$ leaks a part of information about msk. If $\mathcal{A}$ asks derived keys for m linearly independent vectors $\mathbf{d}$, then the whole msk is exposed to $\mathcal{A}$, and consequently, $\mathcal{A}$ can easily generate a valid proof for an instance $[\mathbf{c}] \notin \mathscr{L}_{[\mathbf{A}]}$ via computing $[\mathbf{u}] := \mathbf{K}[\mathbf{c}] + \widehat{\mathbf{K}}_\tau[\mathbf{t}]$.

First Idea. Introducing a Random Matrix as a Secret Permutation. In order to solve the aforementioned problems, we introduce a uniformly random matrix $\mathbf{M} \in \mathbb{Z}_q^{m\times m}$ in msk, i.e., $msk := (\mathbf{K}, \{\widehat{\mathbf{K}}_{\ell,b}\}_{\ell,b}, \mathbf{M})$ with $\mathbf{M} \xleftarrow{\$} \mathbb{Z}_q^{m\times m}$. The crs, the proof generation and the master verification approach are the same as before, while the key deriving process and fine-grained verification are changed as follows. Now the derived key $sk_{\mathbf{d}}$ w.r.t. $\mathbf{d} \in \mathbb{Z}_q^m$ is

$$sk_{\mathbf{d}} := (\mathbf{d}^\top\mathbf{M}, \mathbf{d}^\top\mathbf{M}\mathbf{K}, \{\mathbf{d}^\top\mathbf{M}\widehat{\mathbf{K}}_{\ell,b}\}_{\ell,b}),$$

and the *fine-grained* verification using $sk_{\mathbf{d}}$ checks

$$\mathbf{d}^\top\mathbf{M}[\mathbf{u}] \stackrel{?}{=} \mathbf{d}^\top\mathbf{M}\mathbf{K}[\mathbf{c}] + \mathbf{d}^\top\mathbf{M}\widehat{\mathbf{K}}_\tau[\mathbf{t}].$$

Intuitively, now the $sk_{\mathbf{d}}$ no longer projects msk on vector $\mathbf{d}$, but on a random vector $\mathbf{d}^\top\mathbf{M}$ which secretly rotates $\mathbf{d}$ by the matrix $\mathbf{M}$ in msk. As long as $\mathbf{d}^\top\mathbf{M}$ contains enough entropy from an adversary $\mathcal{A}$'s view[5], it is impossible for $\mathcal{A}$ to output a proof $\pi^* = ([\mathbf{t}^*], [\mathbf{u}^*])$ for $[\mathbf{c}]$ such that

$$\mathbf{d}^\top\mathbf{M}[\mathbf{u}^*] = \mathbf{d}^\top\mathbf{M}\mathbf{K}[\mathbf{c}] + \mathbf{d}^\top\mathbf{M}\widehat{\mathbf{K}}_\tau[\mathbf{t}^*], \text{ but } [\mathbf{u}^*] \neq \mathbf{K}[\mathbf{c}] + \widehat{\mathbf{K}}_\tau[\mathbf{t}^*],$$

except with negligible probability, since otherwise $[\mathbf{u}^*] - \mathbf{K}[\mathbf{c}] - \widehat{\mathbf{K}}_\tau[\mathbf{t}^*]$ constitutes a non-zero vector in the right kernel space of $\mathbf{d}^\top\mathbf{M}$. As a result, verification equivalence is guaranteed.

However, USS still cannot hold, since the whole msk is still exposed to $\mathcal{A}$ if $\mathcal{A}$ asks derived keys for m linearly independent vectors $\mathbf{d}$.

Second Idea. Enlarging the Random Matrix as an Entropy Filter. To rescue USS, we enlarge $\mathbf{M}$ to be a matrix in $\mathbb{Z}_q^{m\times(m+1)}$. Now even if $\mathcal{A}$ queries derived keys $sk_{\mathbf{d}}$ for m linearly independent vectors $\mathbf{d}$, the information about msk leaked to $\mathcal{A}$ is limited in

$$(\mathbf{M}, \mathbf{M}\mathbf{K}, \{\mathbf{M}\widehat{\mathbf{K}}_{\ell,b}\}_{\ell,b}),$$

and there is still entropy left. More precisely, let $\mathbf{m}^\perp \in \mathbb{Z}_q^{m+1}$ be a vector s.t. $\mathbf{M}\mathbf{m}^\perp = \mathbf{0}$, and let $(\mathbf{K}, \{\widehat{\mathbf{K}}_{\ell,b}\}_{\ell,b}) := (\mathbf{K}' + \mathbf{m}^\perp\boxed{\widetilde{\mathbf{k}}}, \{\widehat{\mathbf{K}}'_{\ell,b} + \mathbf{m}^\perp\boxed{\widetilde{\mathbf{k}}_{\ell,b}}\}_{\ell,b})$, where

[5] This entropy requirement is necessary to achieve verification equivalence, see Remark 1 in Sect. 3 for more discussions.

$\mathbf{K}' \xleftarrow{\$} \mathbb{Z}_q^{m\times n_1}$, $\widehat{\mathbf{K}}'_{\ell,b} \xleftarrow{\$} \mathbb{Z}_q^{m\times 3k}$ and $\boxed{\widetilde{\mathbf{k}} \xleftarrow{\$} \mathbb{Z}_q^{1\times n_1}, \widetilde{\mathbf{k}}_{\ell,b} \xleftarrow{\$} \mathbb{Z}_q^{1\times 3k}}$. Then the entropy of $\boxed{(\widetilde{\mathbf{k}}, \{\widetilde{\mathbf{k}}_{\ell,b}\}_{\ell,b})}$ is reserved from the derived key queries, by observing that

$$(\mathbf{M}, \mathbf{MK}, \{\mathbf{M}\widehat{\mathbf{K}}_{\ell,b}\}_{\ell,b}) = (\mathbf{M}, \mathbf{MK}', \{\mathbf{M}\widehat{\mathbf{K}}'_{\ell,b}\}_{\ell,b}).$$

Consequently, the enlarged matrix $\mathbf{M}$ also works as an entropy filter in our FV-NIZK instantiation.

Finally, by using the reserved $\boxed{(\widetilde{\mathbf{k}}, \{\widetilde{\mathbf{k}}_{\ell,b}\}_{\ell,b})}$ (which in turn corresponds to the msk of the DV-NIZK in [16]), we can prove the almost tight USS of our FV-NIZK following the proof strategy in [16].

Others. By using the MDDH assumption, we further prove the almost tight pseudorandomness of the proofs $\pi = ([\mathbf{t}], [\mathbf{u}])$ for adversaries that are not given any derived secret key. This property serves as the core technical tool to achieve anonymity in the fine-grained verifiable PKE application.

Moreover, we note that our aforementioned ideas seem to be general ideas to lift a DV-NIZK scheme with good linearity to an FV-NIZK. Following the similar ideas, we also extend the DV-NIZK scheme proposed by Abe et al. [4] to an FV-NIZK, as our second instantiation.

Roadmap. In Sect. 2 we present notations and recall the MDDH assumptions. The definition and security properties of FV-NIZK are formally described in Sect. 3. Then in Sect. 4, we propose two instantiations of FV-NIZK with almost tight USS for linear subspace languages. In Sect. 5, we illustrate two applications of FV-NIZK in IPFE and FV-PKE, respectively.

2 Preliminaries

Let $\lambda \in \mathbb{N}$ denote the security parameter and $\emptyset$ the empty set. For $\mu \in \mathbb{N}$, define $[\mu] := \{1, 2, ..., \mu\}$. For $a, b \in \mathbb{Z}$ with $a < b$, define $[a, b] := \{a, a+1, ..., b\}$. Denote by $x := y$ the operation of assigning y to x. Denote by $x \xleftarrow{\$} \mathcal{Q}$ the operation of sampling x uniformly at random from a set $\mathcal{Q}$. For a distribution $\mathcal{D}$, denote by $x \leftarrow \mathcal{D}$ the operation of sampling x according to $\mathcal{D}$. For an algorithm $\mathcal{A}$, denote by $y \leftarrow \mathcal{A}(x; r)$, or simply $y \leftarrow \mathcal{A}(x)$, the operation of running $\mathcal{A}$ with input x and randomness r and assigning the output to y. "PPT" is short for probabilistic polynomial-time. $\mathsf{poly}(\lambda)$ and $\mathsf{negl}(\lambda)$ denote polynomial and negligible functions in λ, respectively.

We use bold lower-case letters to denote vectors (e.g., $\mathbf{x}$), and bold upper-case letters to denote matrices (e.g., $\mathbf{A}$). Unless specific description, all vectors are column vectors in this paper. For matrices $\mathbf{A}$ and $\mathbf{B}$, we use $\mathbf{A} \otimes \mathbf{B}$ for their tensor (or Kronecker) product $(a_{i,j}\mathbf{B})_{i,j}$. For vectors $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^m$, let $\langle \mathbf{x}, \mathbf{y} \rangle$ denote their inner product $\mathbf{x}^\top \mathbf{y} \in \mathbb{Z}$. Let $\mathbf{I}_n$ and $\mathbf{0}_{n_1\times n_2}$ denote the identity and zero matrices respectively.

For random variables X and Y, the min-entropy of X is defined as $\mathbf{H}_\infty(X) := -\log(\max_x \Pr[X = x])$, and the average min-entropy of X conditioned on Y is defined as $\widetilde{\mathbf{H}}_\infty(X|Y) := -\log(\mathbb{E}_{y\leftarrow Y}[\max_x \Pr[X = x|Y = y]])$, following [13].

Definition 1 (Collision Resistant Hash Families). *Let $\mathcal{X}, \mathcal{Y}$ be two finite sets. A family of hash functions $\mathcal{H} = \{H : \mathcal{X} \to \mathcal{Y}\}$ is collision resistant, if for any PPT adversary $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{cr}_{\mathcal{H},\mathcal{A}}(\lambda) := \Pr[H \xleftarrow{\$} \mathcal{H}, (x, x') \leftarrow \mathcal{A}(H) : x \neq x' \wedge H(x) = H(x')] \leq \mathsf{negl}(\lambda).$$

2.1 Group Assumptions

Let $\mathcal{G} = (\mathbb{G}, g, q) \leftarrow \mathsf{GGen}$ be a group generation algorithm that inputs 1^λ and returns a cyclic group $\mathbb{G}$ of order q with generator g. For matrix $\mathbf{A} := (a_{ij})_{n_1\times n_2}$ with $a_{ij} \in \mathbb{Z}_q$, we define $[\mathbf{A}] := (g^{a_{ij}})_{n_1\times n_2}$ as the implicit representation of $\mathbf{A}$ in $\mathbb{G}$ [15]. For $\mathbf{A} \in \mathbb{Z}_q^{n_1\times n_2}$, the linear subspace spanned by $\mathbf{A}$ is $\mathsf{Span}(\mathbf{A}) := \{\mathbf{c} \mid \exists \mathbf{s}\ s.t.\ \mathbf{c} = \mathbf{A}\mathbf{s}\}$, and similarly, $\mathsf{Span}([\mathbf{A}]) := \{[\mathbf{c}] \mid \exists \mathbf{s}\ s.t.\ \mathbf{c} = \mathbf{A}\mathbf{s}\}$. Given $\mathbf{A} \in \mathbb{Z}_q^{n_1\times n_2}$, it is efficient to sample an $\mathbf{A}^\perp \in \mathbb{Z}_q^{(n_1-n_2)\times n_1}$ s.t. $\mathbf{A}^\perp\mathbf{A} = \mathbf{0}$.

Let $\ell, k \in \mathbb{N}$ and $\ell > k$. A matrix distribution $\mathcal{D}_{\ell,k}$ is a probabilistic distribution that outputs matrices in $\mathbb{Z}_q^{\ell\times k}$ of full rank k in polynomial time. Especially, if $\mathcal{D}_{\ell,k}$ is a uniform distribution, then we denote it by $\mathcal{U}_{\ell,k}$. In the case $\ell = k+1$, we simply denote it as $\mathcal{D}_k$ or $\mathcal{U}_k$.

Definition 2 ($\mathcal{D}_{\ell,k}$-MDDH Assumption). *Let $\mathcal{D}_{\ell,k}$ be a matrix distribution. The $\mathcal{D}_{\ell,k}$-Matrix Decisional Diffie-Hellman ($\mathcal{D}_{\ell,k}$-MDDH) assumption holds in $\mathbb{G}$, if for any PPT adversary $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{mddh}_{\mathcal{D}_{\ell,k},\mathbb{G},\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}], [\mathbf{A}\mathbf{s}]) = 1] - \Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}], [\mathbf{u}]) = 1]| \leq \mathsf{negl}(\lambda),$$

where $\mathcal{G} \leftarrow \mathsf{GGen}(1^\lambda)$, $\mathbf{A} \leftarrow \mathcal{D}_{\ell,k}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^k$, and $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^\ell$.

Definition 3 (n-fold $\mathcal{D}_{\ell,k}$-MDDH Assumption). *Let $n \geq 1$ and let $\mathcal{D}_{\ell,k}$ be a matrix distribution. The n-fold $\mathcal{D}_{\ell,k}$-MDDH assumption holds in $\mathbb{G}$, if for any PPT adversary $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{n\text{-}mddh}_{\mathcal{D}_{\ell,k},\mathbb{G},\mathcal{A}} := |\Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}], [\mathbf{A}\mathbf{S}]) = 1] - \Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}], [\mathbf{U}]) = 1]| \leq \mathsf{negl}(\lambda),$$

where $\mathcal{G} \leftarrow \mathsf{GGen}(1^\lambda)$, $\mathbf{A} \leftarrow \mathcal{D}_{\ell,k}$, $\mathbf{S} \xleftarrow{\$} \mathbb{Z}_q^{k\times n}$, and $\mathbf{U} \xleftarrow{\$} \mathbb{Z}_q^{\ell\times n}$.

Lemma 1 (Random Self-Reducibility [15, 16]). *Let $n \geq 1$. For any adversary $\mathcal{A}$, there exists an algorithm $\mathcal{B}$ s.t. $Time(\mathcal{B}) \approx Time(\mathcal{A}) + n \cdot \mathsf{poly}(\lambda)$, and $\mathsf{Adv}^{n\text{-}mddh}_{\mathcal{D}_{\ell,k},\mathbb{G},\mathcal{A}}(\lambda) \leq (\ell - k)\mathsf{Adv}^{mddh}_{\mathcal{D}_{\ell,k},\mathbb{G},\mathcal{B}}(\lambda) + \frac{1}{q-1}$.*

For any adversary $\mathcal{A}$, there exists an algorithm $\mathcal{B}$ s.t. $Time(\mathcal{B}) \approx Time(\mathcal{A}) + n \cdot \mathsf{poly}(\lambda)$, and $\mathsf{Adv}^{n\text{-}mddh}_{\mathcal{U}_{\ell,k},\mathbb{G},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{mddh}_{\mathcal{U}_{\ell,k},\mathbb{G},\mathcal{B}}(\lambda) + \frac{1}{q-1}$.

Lemma 2 ($\mathcal{D}_{\ell,k}$-MDDH $\Rightarrow$ $\mathcal{U}_k$-MDDH $\Leftrightarrow$ $\mathcal{U}_{\ell,k}$-MDDH [15, 16]). *Let $\ell, k \in \mathbb{N}$ and $\ell > k$. For any adversary $\mathcal{A}$, there exists an algorithm $\mathcal{B}$ s.t. $Time(\mathcal{B}) \approx Time(\mathcal{A})$, and $\mathsf{Adv}^{mddh}_{\mathcal{U}_k,\mathbb{G},\mathcal{B}}(\lambda) \leq \mathsf{Adv}^{mddh}_{\mathcal{D}_{\ell,k},\mathbb{G},\mathcal{A}}(\lambda)$.*

For any adversary $\mathcal{A}$, there exists an algorithm $\mathcal{B}$ (and vice versa) s.t. $Time(\mathcal{B}) \approx Time(\mathcal{A})$, and $\mathsf{Adv}^{mddh}_{\mathcal{U}_k,\mathbb{G},\mathcal{A}}(\lambda) = \mathsf{Adv}^{mddh}_{\mathcal{U}_{\ell,k},\mathbb{G},\mathcal{B}}(\lambda)$.

3 Fine-grained Verifier NIZK: Definition and Security

In this section, we give the formal definition of *Fine-grained Verifier NIZK (FV-NIZK)*, and propose a set of useful security properties for it.

Let $\mathscr{L} = \{\mathscr{L}_\rho\}$ be a collection of NP-languages indexed by parameter ρ. Each language $\mathscr{L}_\rho$ is determined by a binary relation R_ρ, such that an instance c belongs to $\mathscr{L}_\rho$ iff there exists a witness w s.t. $R_\rho(c, w) = 1$. We consider $\mathscr{L}_\rho$ with a trapdoor td_ρ, which can be used to decide the membership of $\mathscr{L}_\rho$ efficiently.

Definition 4 (Tag-Based FV-NIZK). *A tag-based Fine-grained Verifier quasi-adaptive Non-Interactive Zero-Knowledge (FV-NIZK) argument consists of seven PPT algorithms, namely* $\Pi = (\mathsf{Par}, \mathsf{Gen}, \mathsf{Prove}, \mathsf{MVer}, \mathsf{Sim}, \mathsf{Delegate}, \mathsf{FVer})$.

- $\mathsf{pp} \leftarrow \mathsf{Par}(1^\lambda, \mathscr{L}_\rho)$. *Initialization algorithm takes the security parameter* λ *and a language* $\mathscr{L}_\rho$ *as inputs, and outputs a public parameter* pp*, which defines the tag space* $\mathcal{T}$ *and the delegation space* $\mathcal{D}$.
- $(\mathsf{crs}, \mathsf{td}, msk) \leftarrow \mathsf{Gen}(\mathsf{pp})$. *Generation algorithm takes* pp *as input, and outputs a common reference string* crs*, a trapdoor* td*, and a master secret key* msk*. Without loss of generality, we assume* crs *contains* pp*, and it serves as an implicit input of* MVer, Sim, Delegate, and FVer.
- $\pi \leftarrow \mathsf{Prove}(\mathsf{crs}, c, w, \tau)$. *Proof algorithm takes* crs*, an instance* $c \in \mathscr{L}_\rho$ *along with a witness* w*, and a tag* $\tau \in \mathcal{T}$ *as inputs, and outputs a proof* π.
- $0/1 \leftarrow \mathsf{MVer}(msk, c, \tau, \pi)$. *Master verification algorithm takes* msk*, an instance* c*, a tag* $\tau \in \mathcal{T}$ *and a proof* π *as inputs, and outputs a decision bit.*
- $\pi \leftarrow \mathsf{Sim}(\mathsf{td}, c, \tau)$. *Simulation algorithm takes* td*, an instance* c *and a tag* $\tau \in \mathcal{T}$ *as inputs, and outputs a simulated proof* π.
- $sk_d \leftarrow \mathsf{Delegate}(msk, d)$. *Delegation algorithm takes* msk *and a delegation* $d \in \mathcal{D}$ *as inputs, and outputs a delegated secret key* sk_d.
- $0/1 \leftarrow \mathsf{FVer}(sk_d, c, \tau, \pi)$. *Fine-grained verification algorithm takes* sk_d*, an instance* c*, a tag* $\tau \in \mathcal{T}$ *and a proof* π *as inputs, and outputs a decision bit.*

If the tag space $\mathcal{T}$ *is the empty set* $\emptyset$ *or contains only one element (e.g.,* $\{0\}$*), we call* Π *an FV-NIZK argument.*

We require Π *to have completeness and (perfect) zero-knowledge.*

Completeness. *For all* $\mathsf{pp} \leftarrow \mathsf{Par}(1^\lambda, \mathscr{L}_\rho)$, $(\mathsf{crs}, \mathsf{td}, msk) \leftarrow \mathsf{Gen}(\mathsf{pp})$, (c, w) *s.t.* $R_\rho(c, w) = 1$, $\tau \in \mathcal{T}$ *and* $\pi \leftarrow \mathsf{Prove}(\mathsf{crs}, c, w, \tau)$*, it holds that*

(1) $\mathsf{MVer}(msk, c, \tau, \pi) = 1$*, and*

(2) $\mathsf{FVer}(sk_d, c, \tau, \pi) = 1$ *for all* $sk_d \leftarrow \mathsf{Delegate}(msk, d)$ *of all* $d \in \mathcal{D}$.

Perfect Zero-Knowledge. *For all* $\mathsf{pp} \leftarrow \mathsf{Par}(1^\lambda, \mathscr{L}_\rho)$, $(\mathsf{crs}, \mathsf{td}, msk) \leftarrow \mathsf{Gen}(\mathsf{pp})$, (c, w) *s.t.* $R_\rho(c, w) = 1$ *and* $\tau \in \mathcal{T}$*, the following two distributions are identical:*

$$\mathsf{Prove}(\mathsf{crs}, c, w, \tau) \quad \equiv \quad \mathsf{Sim}(\mathsf{td}, c, \tau).$$

Note that the first five algorithms ($\mathsf{Par}, \mathsf{Gen}, \mathsf{Prove}, \mathsf{MVer}, \mathsf{Sim}$) of FV-NIZK basically constitute a DV-NIZK scheme as defined in [16]. Moreover, the two additional algorithms ($\mathsf{Delegate}, \mathsf{FVer}$) provide the fine-grained verification ability, by allowing different users owning different secret keys sk_d ($d \in \mathcal{D}$) to verify proofs in different ways by invoking $\mathsf{FVer}(sk_d, \cdot, \cdot, \cdot)$.

Now, we define a statistical property called *verification equivalence* for FV-NIZK. Intuitively, it requires that all proofs passing the master verification algorithm MVer using msk also pass the fine-grained verification algorithm FVer using any secret key sk_d of any d, and (with high probability) vice versa.

Definition 5 (Verification Equivalence). *Let $\delta, \epsilon > 0$. A tag-based FV-NIZK Π has (δ, ϵ)-verification equivalence, if the following two properties hold.*

1. $\underline{\mathsf{MVer} \Longrightarrow \mathsf{FVer}}$*:* *For all* $\mathsf{pp} \leftarrow \mathsf{Par}(1^\lambda, \mathscr{L}_\rho)$, $(\mathsf{crs}, \mathsf{td}, msk) \leftarrow \mathsf{Gen}(\mathsf{pp})$, *instances c, proofs π and tags $\tau \in \mathcal{T}$, if* $\mathsf{MVer}(msk, c, \tau, \pi) = 1$ *holds, then* $\mathsf{FVer}(sk_d, c, \tau, \pi) = 1$ *holds for all* $sk_d \leftarrow \mathsf{Delegate}(msk, d)$ *of all* $d \in \mathcal{D}$.
2. $\underline{\mathsf{MVer} \overset{w.h.p.}{\Longleftarrow} \mathsf{FVer}}$*:* *For any (even unbounded) adversary $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{ver\text{-}equ}_{\Pi,\mathcal{A},\delta}(\lambda) := \Pr[\mathsf{Exp}^{ver\text{-}equ}_{\Pi,\mathcal{A},\delta}(\lambda) \Rightarrow 1] \leq \epsilon,$$

where the experiment $\mathsf{Exp}^{ver\text{-}equ}_{\Pi,\mathcal{A},\delta}(\lambda)$ *is defined in Fig. 1.*

$\underline{\mathsf{Exp}^{ver\text{-}equ}_{\Pi,\mathcal{A},\delta}(\lambda)}$:	$\underline{\textsc{Sim}(c, \tau)}$:
$\mathsf{pp} \leftarrow \mathsf{Par}(1^\lambda, \mathscr{L}_\rho)$, $(\mathsf{crs}, \mathsf{td}, msk) \leftarrow \mathsf{Gen}(\mathsf{pp})$	$\pi \leftarrow \mathsf{Sim}(\mathsf{td}, c, \tau)$
$\mathcal{Q}_{sim} := \emptyset$, $\mathcal{Q}_{sk} := \emptyset$	$\mathcal{Q}_{sim} := \mathcal{Q}_{sim} \cup \{(c, \tau, \pi)\}$
$(c^*, \tau^*, \pi^*, d^*) \leftarrow \mathcal{A}^{\textsc{Sim}(\cdot,\cdot), \textsc{Delegate}(\cdot)}(\mathsf{pp}, \mathsf{crs})$	Return π
$sk_{d^*} \leftarrow \mathsf{Delegate}(msk, d^*)$	
If $\widetilde{\mathbf{H}}_\infty(sk_{d^*} \vert \mathsf{crs}, \mathcal{Q}_{sim}, \mathcal{Q}_{sk}, d^*) > \delta$	$\underline{\textsc{Delegate}(d)}$:
$\wedge\ \mathsf{FVer}(sk_{d^*}, c^*, \tau^*, \pi^*) = 1$	$sk_d \leftarrow \mathsf{Delegate}(msk, d)$
$\wedge\ \mathsf{MVer}(msk, c^*, \tau^*, \pi^*) = 0$: output 1	$\mathcal{Q}_{sk} := \mathcal{Q}_{sk} \cup \{(d, sk_d)\}$
Otherwise: output 0	Return sk_d

Fig. 1. The verification equivalence experiment $\mathsf{Exp}^{ver\text{-}equ}_{\Pi,\mathcal{A},\delta}(\lambda)$ for tag-based FV-NIZK. In the condition "$\widetilde{\mathbf{H}}_\infty(sk_{d^*} \vert \mathsf{crs}, \mathcal{Q}_{sim}, \mathcal{Q}_{sk}, d^*)$", sk_{d^*} means the distribution $\mathsf{Delegate}(msk, d^*; r)$ with uniformly chosen randomness r, rather than a fixed value.

Remark 1 (On the formalization of "$\mathsf{MVer} \overset{w.h.p.}{\Longleftarrow} \mathsf{FVer}$"). We stress that we do not require MVer and FVer perform identically on all inputs. In other words, there might exist (c, τ, π) such that $\mathsf{FVer}(sk_d, c, \tau, \pi) = 1$ for some sk_d but $\mathsf{MVer}(msk, c, \tau, \pi) = 0$. Similarly, for different d_1, d_2, FVer using sk_{d_1} and FVer using sk_{d_2} might perform differently on some inputs, i.e., there might exist (c, τ, π) such that $\mathsf{FVer}(sk_{d_1}, c, \tau, \pi) = 1$ but $\mathsf{FVer}(sk_{d_2}, c, \tau, \pi) = 0$.

In fact, what our "$\mathsf{MVer} \overset{w.h.p.}{\Longleftarrow} \mathsf{FVer}$" property tries to characterize is that for any (unbounded) adversary $\mathcal{A}$ who does not get enough information about sk_{d^*} (and thus msk), it is hard to find a (c^*, τ^*, π^*) that makes MVer and FVer perform differently. This also explains the condition "$\widetilde{\mathbf{H}}_\infty(sk_{d^*}|\mathsf{crs}, \mathcal{Q}_{sim}, \mathcal{Q}_{sk}, d^*) > \delta$" in Fig. 1 for $\mathcal{A}$ to win. Otherwise, if the min-entropy of sk_{d^*} is lower than some threshold (say δ), $\mathcal{A}$ can guess sk_{d^*} correctly with a noticeable probability. Meanwhile, it can obtain sk_d for some $d \neq d^*$ by querying $\textsc{Delegate}(d)$. With the knowledge of sk_{d^*} and sk_d, it is feasible for $\mathcal{A}$ to find (c^*, τ^*, π^*) such that $\mathsf{FVer}(sk_{d^*}, c^*, \tau^*, \pi^*) = 1$ but $\mathsf{FVer}(sk_d, c^*, \tau^*, \pi^*) = 0$ (e.g., via brute-force search). According to the first property "$\mathsf{MVer} \Longrightarrow \mathsf{FVer}$", $\mathsf{FVer}(sk_d, c^*, \tau^*, \pi^*) = 0$ implies $\mathsf{MVer}(msk, c^*, \tau^*, \pi^*) = 0$, and consequently $\mathcal{A}$ wins in $\mathsf{Exp}^{ver\text{-}equ}_{\Pi,\mathcal{A},\delta}(\lambda)$. To prevent such trivial attacks, we require $\widetilde{\mathbf{H}}_\infty(sk_{d^*}|\mathsf{crs}, \mathcal{Q}_{sim}, \mathcal{Q}_{sk}, d^*) > \delta$.

Remark 2 (On the parameter δ). Jumping ahead, both our FV-NIZK constructions in Sect. 4 has (δ, ϵ)-verification equivalence with $\delta = 0$. It seems that the only way to achieve verification equivalence is if the parameter δ is either exactly 0 (as in our case) or large, but nothing in between.

Next, we adapt the *unbounded simulation soundness (USS)* of NIZK to our FV-NIZK. Recall that USS for NIZK and DV-NIZK ensures that a PPT adversary cannot generate a valid proof for a fresh and false statement $c \notin \mathscr{L}_\rho$, even if it can obtain multiple simulated proofs for instances not necessarily in $\mathscr{L}_\rho$ [30, 16]. For FV-NIZK, we also allow the adversary to obtain many secret keys sk_d with d of its choices. Moreover, we consider a *strong* USS by giving the adversary multiple chances to win, following [16].

Definition 6 (Strong USS). *A tag-based FV-NIZK Π has strong USS, if for any PPT adversary $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{uss}_{\Pi,\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{uss}_{\Pi,\mathcal{A}}(\lambda) \Rightarrow 1] \leq \mathsf{negl}(\lambda),$$

where the experiment $\mathsf{Exp}^{uss}_{\Pi,\mathcal{A}}(\lambda)$ is defined in Fig. 2.

Remark 3 (On the formalization of strong USS). Note that in the strong USS experiment in Fig. 2, $\textsc{Sim}(c, \tau)$ returns $\perp$ directly if τ was queried to $\textsc{Sim}(\cdot, \cdot)$ before, following the definition of strong USS for DV-NIZK in [16]. Similar to [16], such a requirement is not an obstacle in many applications. For example, as we will see, in all our applications in Sect. 5, τ is a hash of some random values. Thus τ is different with overwhelming probability each time $\textsc{Sim}(\cdot, \cdot)$ is invoked when the security of applications is reduced to the strong USS.

Moreover, we note that in the strong USS defined in [16], $\textsc{Ver}(\cdot, \tau, \cdot)$ also returns $\perp$ if τ was queried to $\textsc{Sim}(\cdot, \tau)$ before, while ours does not have such a requirement. This relaxation seems reasonable when considering the security of NIZK, and it helps us to construct other cryptographic algorithms in a more straightforward way (e.g., constructing CCA-secure PKE without resorting to one-time signatures or authenticated encryption, as shown in Subsect. 5.2).

$\underline{\mathsf{Exp}^{uss}_{\Pi,\mathcal{A}}(\lambda)}$:	$\underline{\textsc{Sim}(c,\tau)}$:
$\mathsf{pp} \leftarrow \mathsf{Par}(1^\lambda, \mathscr{L}_\rho)$, $(\mathsf{crs}, \mathsf{td}, msk) \leftarrow \mathsf{Gen}(\mathsf{pp})$ $\mathcal{Q}_{sim} := \emptyset$, $\mathcal{Q}_{sk} := \emptyset$ $\mathsf{win} := 0$ // A flag indicating whether $\mathcal{A}$ wins $\perp \leftarrow \mathcal{A}^{\textsc{Sim}(\cdot,\cdot),\textsc{Delegate}(\cdot),\textsc{Ver}(\cdot,\cdot,\cdot)}(\mathsf{pp}, \mathsf{crs})$ Output win	If $(\cdot,\tau,\cdot) \in \mathcal{Q}_{sim}$: return $\perp$ $\pi \leftarrow \mathsf{Sim}(\mathsf{td}, c, \tau)$ $\mathcal{Q}_{sim} := \mathcal{Q}_{sim} \cup \{(c,\tau,\pi)\}$ Return π
$\underline{\textsc{Delegate}(d)}$: $sk_d \leftarrow \mathsf{Delegate}(msk, d)$ $\mathcal{Q}_{sk} := \mathcal{Q}_{sk} \cup \{(d, sk_d)\}$ Return sk_d	$\underline{\textsc{Ver}(c,\tau,\pi)}$: If $(c,\tau,\pi) \in \mathcal{Q}_{sim}$: return $\perp$ If $\mathsf{MVer}(msk, c, \tau, \pi) = 1 \ \wedge \ c \notin \mathscr{L}_\rho$: $\mathsf{win} := 1$ Return $\mathsf{MVer}(msk, c, \tau, \pi)$

Fig. 2. The strong USS experiment $\mathsf{Exp}^{uss}_{\Pi,\mathcal{A}}(\lambda)$ for tag-based FV-NIZK.

Finally, we define *proof pseudorandomness* for FV-NIZK, which stipulates the pseudorandomness of proofs for PPT adversaries that are not given any secret key but allowed to access the verification oracle. Jumping ahead, this property serves as the core technical tool for the ciphertext pseudorandomness (thus anonymity) of our fine-grained verifiable PKE in Subsect. 5.2.

Definition 7 (Proof Pseudorandomness). *A tag-based FV-NIZK Π has proof pseudorandomness, if for any PPT adversary $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{pp}_{\Pi,\mathcal{A}}(\lambda) := |\Pr[\mathsf{Exp}^{pp}_{\Pi,\mathcal{A},0}(\lambda) \Rightarrow 1] - \Pr[\mathsf{Exp}^{pp}_{\Pi,\mathcal{A},1}(\lambda) \Rightarrow 1]| \leq \mathsf{negl}(\lambda),$$

where the experiments $\mathsf{Exp}^{pp}_{\Pi,\mathcal{A},\beta}(\lambda)$ ($\beta \in \{0,1\}$) are defined in Fig. 3.

$\underline{\mathsf{Exp}^{pp}_{\Pi,\mathcal{A},\beta}(\lambda)}$: // $\beta \in \{0,1\}$	$\underline{\textsc{Sam}(\cdot)}$:	$\underline{\textsc{Sim}(c,\tau)}$:
$\mathsf{pp} \leftarrow \mathsf{Par}(1^\lambda, \mathscr{L}_\rho)$, $(\mathsf{crs}, \mathsf{td}, msk) \leftarrow \mathsf{Gen}(\mathsf{pp})$ $\mathcal{Q}_c := \emptyset$, $\mathcal{Q}_{sim} := \emptyset$ $\beta' \leftarrow \mathcal{A}^{\textsc{Sam}(\cdot),\textsc{Sim}(\cdot,\cdot),\textsc{Ver}(\cdot,\cdot,\cdot)}(\mathsf{pp}, \mathsf{crs})$ Output β'	If $\beta = 0$: $c \xleftarrow{\$} \mathscr{L}_\rho$ If $\beta = 1$: $c \xleftarrow{\$} \mathcal{X}$ $\mathcal{Q}_c := \mathcal{Q}_c \cup \{c\}$ Return c	If $c \notin \mathcal{Q}_c$: return $\perp$ If $(\cdot,\tau,\cdot) \in \mathcal{Q}_{sim}$: return $\perp$ If $\beta = 0$: $\pi \leftarrow \mathsf{Sim}(\mathsf{td}, c, \tau)$ If $\beta = 1$: $\pi \xleftarrow{\$} \mathcal{P}$ $\mathcal{Q}_c := \mathcal{Q}_c \backslash \{c\}$ $\mathcal{Q}_{sim} := \mathcal{Q}_{sim} \cup \{(c,\tau,\pi)\}$ Return π
$\underline{\textsc{Ver}(c,\tau,\pi)}$: If $(c,\tau,\pi) \in \mathcal{Q}_{sim}$: return $\perp$ Return $\mathsf{MVer}(msk, c, \tau, \pi)$		

Fig. 3. The proof pseudorandomness experiments $\mathsf{Exp}^{pp}_{\Pi,\mathcal{A},\beta}(\lambda)$ for tag-based FV-NIZK, where $\mathcal{X}$ denotes the instance space, and $\mathcal{P}$ denotes the proof space of Π.

Remark 4 (On the formalization of proof pseudorandomness). In fact, the proof pseudorandomness asks the pseudorandomness of proofs for instances *uniformly sampled* from the language $\mathscr{L}_\rho$. Moreover, the adversary $\mathcal{A}$ in Fig. 3 has access to two oracles, $\textsc{Sam}(\cdot)$ and $\textsc{Sim}(\cdot,\cdot)$, to obtain instances and simulated proofs,

respectively. In particular, the oracle $\textsc{Sim}(c, \tau)$ returns proofs only for instances c output by $\textsc{Sam}(\cdot)$, but τ can be determined by $\mathcal{A}$. Indeed, in certain applications of tag-based NIZK, the tag τ may depend on the instance c. For example, in our application in PKE (cf. Subsect. 5.2), τ is a hash of c. Our formalization captures such dependency between c and τ.

Remark 5 (Extension to the multi-user setting). We can naturally extend the definitions of strong USS and proof pseudorandomness (i.e., Def. 6 and Def. 7) to the multi-user setting, and define strong μ-USS and μ-proof pseudorandomness in the setting of $\mu \in \mathbb{N}$ users. The formal definitions can be found in the full version. More precisely, all μ users share the same pp and each user $i \in [\mu]$ invokes $\mathsf{Gen}(\mathsf{pp})$ independently to get its own $(\mathsf{crs}^{(i)}, \mathsf{td}^{(i)}, msk^{(i)})$. Accordingly, the adversary $\mathcal{A}$ has access to $\textsc{Sim}(i, \cdot, \cdot), \textsc{Delegate}(i, \cdot), \textsc{Ver}(i, \cdot, \cdot, \cdot)$ which additionally take a user index $i \in [\mu]$ as input and prepare the responses using $(\mathsf{crs}^{(i)}, \mathsf{td}^{(i)}, msk^{(i)})$.

Jumping ahead, both the two schemes in Sect. 4 have almost tight strong USS (and the first one also have almost tight proof pseudorandomness) in the multi-user setting.

4 FV-NIZK for Linear Subspace Languages

In this section, we propose two tightly secure FV-NIZK schemes for linear subspace languages, based on the MDDH assumption. The first scheme is pairing-free and the second one relies on pairings.

Let $\mathcal{G} = (\mathbb{G}, g, q)$ be a cyclic group $\mathbb{G}$ of order q with generator g. Let $\mathbf{A} \in \mathbb{Z}_q^{n_1 \times n_2}$ with $n_1 > n_2$. The linear subspace language is $\mathscr{L}_{[\mathbf{A}]} := \mathsf{Span}([\mathbf{A}]) := \{[\mathbf{c}] \mid \exists \mathbf{s} \in \mathbb{Z}_q^{n_2} \ s.t. \ \mathbf{c} = \mathbf{A}\mathbf{s}\}$ with $\mathbf{A}$ the trapdoor of $\mathscr{L}_{[\mathbf{A}]}$.

4.1 The First Construction without Pairings

Let $m, k, n_1, n_2 \in \mathbb{N}$ and $\mathcal{D}_{3k,k}$ be a matrix distribution. Let $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_q$ be a family of collision resistant hash functions. Our first construction of tag-based FV-NIZK Π is shown in Fig. 4, where the tag space is $\mathcal{T} = \{0,1\}^\lambda$ and the delegation space is $\mathcal{D} = \mathbb{Z}_q^m$. Note that this construction is pairing-free.

Completeness and perfect zero-knowledge follow directly from the fact that

$$\begin{aligned}\mathbf{u} &= (\mathbf{K}_0 + \theta\mathbf{K}_1)\mathbf{A}\mathbf{s} + \widehat{\mathbf{K}}_\tau \mathbf{B}\mathbf{r} \ = (\mathbf{K}_0 + \theta\mathbf{K}_1)\mathbf{c} + \widehat{\mathbf{K}}_\tau \mathbf{t} && \text{// completeness (1)} \\ &= (\mathbf{K}_0 + \theta\mathbf{K}_1)\mathbf{c} + \widehat{\mathbf{K}}_\tau \mathbf{B}\mathbf{r}, && \text{// perfect zero-knowledge}\end{aligned}$$

which implies $\mathbf{d}^\top \mathbf{M}\mathbf{u} = \mathbf{d}^\top \mathbf{M}(\mathbf{K}_0 + \theta\mathbf{K}_1)\mathbf{c} + \mathbf{d}^\top \mathbf{M}\widehat{\mathbf{K}}_\tau \mathbf{t}$. // completeness (2)

Next, we show the verification equivalence of Π.

Theorem 1 (Verification Equivalence). *The tag-based FV-NIZK scheme Π in Fig. 4 has $(0, 1/q)$-verification equivalence.*

$\underline{\mathsf{Par}(1^\lambda, [\mathbf{A}] \in \mathbb{G}^{n_1 \times n_2})}$:
$\mathbf{B} \leftarrow \mathcal{D}_{3k,k}$; $H \xleftarrow{\$} \mathcal{H}$
Return $\mathsf{pp} := ([\mathbf{A}], [\mathbf{B}], H)$

$\underline{\mathsf{Gen}(\mathsf{pp})}$:
$\mathbf{K}_0, \mathbf{K}_1 \xleftarrow{\$} \mathbb{Z}_q^{(m+1)\times n_1}$; $\mathbf{M} \xleftarrow{\$} \mathbb{Z}_q^{m\times(m+1)}$
For $\ell \in [\lambda], b \in \{0,1\}$: $\widehat{\mathbf{K}}_{\ell,b} \xleftarrow{\$} \mathbb{Z}_q^{(m+1)\times 3k}$
$\mathsf{crs} := ([\mathbf{K}_0\mathbf{A}], [\mathbf{K}_1\mathbf{A}], \{[\widehat{\mathbf{K}}_{\ell,b}\mathbf{B}]\}_{\ell,b})$
$\mathsf{td} := (\mathbf{K}_0, \mathbf{K}_1)$
$msk := (\mathbf{K}_0, \mathbf{K}_1, \{\widehat{\mathbf{K}}_{\ell,b}\}_{\ell,b}, \mathbf{M})$
Return $(\mathsf{crs}, \mathsf{td}, msk)$

$\underline{\mathsf{Prove}(\mathsf{crs}, [\mathbf{c}], \mathbf{s}, \tau)}$: // $\mathbf{c} = \mathbf{A}\mathbf{s}$
$\mathbf{r} \xleftarrow{\$} \mathbb{Z}_q^k$; $[\mathbf{t}] := [\mathbf{B}]\mathbf{r}$
$\theta := H([\mathbf{c}], \tau, [\mathbf{t}])$; $\widehat{\mathbf{K}}_\tau := \sum_{\ell=1}^{\lambda} \widehat{\mathbf{K}}_{\ell,\tau_\ell}$
$[\mathbf{u}] := [(\mathbf{K}_0 + \theta\mathbf{K}_1)\mathbf{A}]\mathbf{s} + [\widehat{\mathbf{K}}_\tau\mathbf{B}]\mathbf{r} \in \mathbb{G}^{m+1}$
Return $\pi := ([\mathbf{t}], [\mathbf{u}])$

$\underline{\mathsf{MVer}(msk, [\mathbf{c}], \tau, \pi = ([\mathbf{t}], [\mathbf{u}]))}$:
$\theta := H([\mathbf{c}], \tau, [\mathbf{t}])$; $\widehat{\mathbf{K}}_\tau := \sum_{\ell=1}^{\lambda} \widehat{\mathbf{K}}_{\ell,\tau_\ell}$
If $[\mathbf{u}] = (\mathbf{K}_0 + \theta\mathbf{K}_1)[\mathbf{c}] + \widehat{\mathbf{K}}_\tau[\mathbf{t}]$: return 1
Otherwise: return 0

$\underline{\mathsf{Sim}(\mathsf{td}, [\mathbf{c}], \tau)}$:
$\mathbf{r} \xleftarrow{\$} \mathbb{Z}_q^k$; $[\mathbf{t}] := [\mathbf{B}]\mathbf{r}$
$\theta := H([\mathbf{c}], \tau, [\mathbf{t}])$; $\widehat{\mathbf{K}}_\tau := \sum_{\ell=1}^{\lambda} \widehat{\mathbf{K}}_{\ell,\tau_\ell}$
$[\mathbf{u}] := (\mathbf{K}_0 + \theta\mathbf{K}_1)[\mathbf{c}] + [\widehat{\mathbf{K}}_\tau\mathbf{B}]\mathbf{r} \in \mathbb{G}^{m+1}$
Return $\pi := ([\mathbf{t}], [\mathbf{u}])$

$\underline{\mathsf{Delegate}(msk, \mathbf{d} \in \mathbb{Z}_q^m)}$:
Return $sk_{\mathbf{d}} := (\mathbf{d}^\top\mathbf{M}, \mathbf{d}^\top\mathbf{M}\mathbf{K}_0, \mathbf{d}^\top\mathbf{M}\mathbf{K}_1, \{\mathbf{d}^\top\mathbf{M}\widehat{\mathbf{K}}_{\ell,b}\}_{\ell,b})$

$\underline{\mathsf{FVer}(sk_{\mathbf{d}}, [\mathbf{c}], \tau, \pi = ([\mathbf{t}], [\mathbf{u}]))}$:
$\theta := H([\mathbf{c}], \tau, [\mathbf{t}])$; $\widehat{\mathbf{K}}_\tau := \sum_{\ell=1}^{\lambda} \widehat{\mathbf{K}}_{\ell,\tau_\ell}$
If $\mathbf{d}^\top\mathbf{M}[\mathbf{u}] = \mathbf{d}^\top\mathbf{M}(\mathbf{K}_0 + \theta\mathbf{K}_1)[\mathbf{c}] + \mathbf{d}^\top\mathbf{M}\widehat{\mathbf{K}}_\tau[\mathbf{t}]$: return 1
Otherwise: return 0

Fig. 4. The pairing-free construction of tag-based FV-NIZK Π.

Proof. The first property ($\mathsf{MVer} \Longrightarrow \mathsf{FVer}$) is straightforward, since $[\mathbf{u}] = (\mathbf{K}_0 + \theta\mathbf{K}_1)[\mathbf{c}] + \widehat{\mathbf{K}}_\tau[\mathbf{t}]$ directly implies $\mathbf{d}^\top\mathbf{M}[\mathbf{u}] = \mathbf{d}^\top\mathbf{M}(\mathbf{K}_0 + \theta\mathbf{K}_1)[\mathbf{c}] + \mathbf{d}^\top\mathbf{M}\widehat{\mathbf{K}}_\tau[\mathbf{t}]$.

To show the second property ($\mathsf{MVer} \overset{w.h.p.}{\Longleftarrow} \mathsf{FVer}$), we consider an (unbounded) adversary $\mathcal{A}$ that finally outputs $([\mathbf{c}^*], \tau^*, \pi^* = ([\mathbf{t}^*], [\mathbf{u}^*]), \mathbf{d}^*)$ in the experiment $\mathsf{Exp}^{ver\text{-}equ}_{\Pi,\mathcal{A},0}(\lambda)$ (cf. Fig. 1). Let $\mathbf{D}$ denote the matrix consisting of all vectors $\mathbf{d}$ that $\mathcal{A}$ queried $\textsc{Delegate}(\cdot)$. We analyze $\mathcal{A}$'s advantage as follows.

Note that the algorithm $\mathsf{Delegate}$ is deterministic and linear in $\mathbf{d}$. That is, if $\mathbf{d}^* \in \mathsf{Span}(\mathbf{D})$, then $sk_{\mathbf{d}^*}$ is totally determined by $\mathcal{Q}_{sk} = \{(\mathbf{d}, sk_{\mathbf{d}})\}$ and $\mathbf{d}^*$, and hence has no entropy left at all. Therefore, for $\mathcal{A}$ to win, $\widetilde{\mathbf{H}}_\infty(sk_{\mathbf{d}^*}|\mathsf{crs}, \mathcal{Q}_{sim}, \mathcal{Q}_{sk}, \mathbf{d}^*) > 0$ holds, and we must have $\mathbf{d}^* \notin \mathsf{Span}(\mathbf{D})$. Moreover, since the algorithm Sim does not involve $\mathbf{M}$ at all, $\mathcal{A}$ obtains nothing about $\mathbf{M}$ from $\textsc{Sim}(\cdot,\cdot)$. Thus, $\mathbf{d}^* \notin \mathsf{Span}(\mathbf{D})$ implies that ${\mathbf{d}^*}^\top\mathbf{M}$ is uniformly random over $\mathbb{Z}_q^{1\times(m+1)}$ from $\mathcal{A}$'s view. And consequently, the event $\mathsf{FVer}(sk_{\mathbf{d}^*}, [\mathbf{c}^*], \tau^*, \pi^*) = 1 \wedge \mathsf{MVer}(msk, [\mathbf{c}^*], \tau^*, \pi^*) = 0$, i.e.,

$$ {\mathbf{d}^*}^\top\mathbf{M} \underbrace{\left(\mathbf{u}^* - (\mathbf{K}_0 + \theta^*\mathbf{K}_1)\mathbf{c}^* - \widehat{\mathbf{K}}_{\tau^*}\mathbf{t}^*\right)}_{\neq \mathbf{0}} = 0, $$

occurs with probability at most $1/q$. This shows $\mathsf{Adv}^{ver\text{-}equ}_{\Pi,\mathcal{A},0}(\lambda) \leq 1/q$. □

Now we show that Π has almost tight strong USS and almost tight proof pseudorandomness via the following two theorems.

Theorem 2 (Almost Tight Strong USS). *If the $\mathcal{D}_{3k,k}$-MDDH assumption holds in $\mathbb{G}$ and $\mathcal{H}$ is a family of collision resistant hash functions, then the tag-*

based FV-NIZK scheme Π in Fig. 4 has strong USS. More precisely, for any adversary $\mathcal{A}$ against the strong USS security of Π, there exist algorithms $\mathcal{B}_1, \mathcal{B}_2$ s.t. $\max(Time(\mathcal{B}_1), Time(\mathcal{B}_2)) \approx Time(\mathcal{A}) + (Q_{sim} + Q_{ver} + Q_{del}) \cdot \mathsf{poly}(\lambda)$*, and*

$$\mathsf{Adv}^{uss}_{\Pi,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{cr}_{\mathcal{H},\mathcal{B}_1}(\lambda) + (8\lambda k + 2k) \cdot \mathsf{Adv}^{mddh}_{\mathcal{D}_{3k,k},\mathbb{G},\mathcal{B}_2}(\lambda) + \tfrac{(2\lambda+2)Q_{ver}+4\lambda+1}{q-1},$$

where $Q_{sim}, Q_{ver}, Q_{del}$ denote the numbers of queries to Sim, Ver, Delegate*, respectively.*

Theorem 3 (Almost Tight Proof Pseudorandomness). *Let $n_1 \geq 2n_2$. If the $\mathcal{D}_{n_1,n_2}$-MDDH assumption and the $\mathcal{D}_{3k,k}$-MDDH assumption hold in $\mathbb{G}$, and $\mathcal{H}$ is a family of collision resistant hash functions, then the tag-based FV-NIZK scheme Π in Fig. 4 has proof pseudorandomness. More precisely, for any adversary $\mathcal{A}$ against the proof pseudorandomness of Π, there exist algorithms $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ s.t.* $\max(Time(\mathcal{B}_1), Time(\mathcal{B}_2)\ Time(\mathcal{B}_3)) \approx Time(\mathcal{A}) + (Q_{sim} + Q_{ver}) \cdot \mathsf{poly}(\lambda)$*, and*

$$\begin{aligned}\mathsf{Adv}^{pp}_{\Pi,\mathcal{A}}(\lambda) \leq & (n_1 - n_2 + 2)\mathsf{Adv}^{mddh}_{\mathcal{D}_{n_1,n_2},\mathbb{G},\mathcal{B}_1}(\lambda) + (16\lambda k + 6k)\mathsf{Adv}^{mddh}_{\mathcal{D}_{3k,k},\mathbb{G},\mathcal{B}_2}(\lambda) \\ & + 2\mathsf{Adv}^{cr}_{\mathcal{H},\mathcal{B}_3}(\lambda) + \tfrac{(4\lambda+4)Q_{ver}+8\lambda+6}{q-1},\end{aligned}$$

where Q_{sim} and Q_{ver} denote the numbers of queries to Sim *and* Ver*, respectively.*

We prove Theorem 2 and Theorem 3 in our full version due to space limitations. See Sect. 1 for a high-level proof sketch.

Remark 6 (On the almost tightness of strong USS and proof pseudorandomness). The terms $\frac{(2\lambda+2)Q_{ver}+4\lambda+1}{q-1}$ and $\frac{(4\lambda+4)Q_{ver}+8\lambda+6}{q-1}$ in Theorem 2 and Theorem 3 do not affect the tightness of the reductions since they are statistically small. Moreover, n_1, n_2, k are parameters of the MDDH assumptions and are constants (e.g., $n_1 = 2, n_2 = 1, k = 1$). Consequently, the strong USS and proof pseudorandomness have security loss factors $O(\lambda)$, and thus are almost tight.

4.2 The Second Construction with Pairings

Let $m, k, n_1, n_2 \in \mathbb{N}$ and $\mathcal{D}_{2k,k}$ be a matrix distribution. Let $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_q$ be a family of collision resistant hash functions. Similar to [4], we use a NIZK proof $\Pi_{or} = (\Pi_{or}.\mathsf{Gen}, \Pi_{or}.\mathsf{TGen}, \Pi_{or}.\mathsf{Prove}, \Pi_{or}.\mathsf{Sim}, \Pi_{or}.\mathsf{Ver})$ for OR-language $\mathscr{L}^{\vee}_{[\mathbf{B}_0],[\mathbf{B}_1]} := \mathsf{Span}([\mathbf{B}_0]) \cup \mathsf{Span}([\mathbf{B}_1]) := \{[\mathbf{t}] \mid \exists \mathbf{r} \in \mathbb{Z}_q^k \text{ s.t. } \mathbf{t} = \mathbf{B}_0\mathbf{r} \ \vee \mathbf{t} = \mathbf{B}_1\mathbf{r}\}$ as a building block, where $\mathbf{B}_0, \mathbf{B}_1 \in \mathbb{Z}_q^{2k\times k}$. We refer our full version for the syntax of NIZK proofs and a concrete MDDH-based scheme of Π_{or} proposed in [18, 29]. Our second construction of tag-based FV-NIZK Π is shown in Fig. 5, where the tag space is $\mathcal{T} = \{0,1\}^*$ and the delegation space is $\mathcal{D} = \mathbb{Z}_q^m$. Note that compared to the QA-NIZK scheme proposed in [4], our FV-NIZK scheme uses less pairing operations, since only $\Pi_{or}.\mathsf{Ver}$ involves pairings.

$\underline{\mathsf{Par}(1^\lambda, [\mathbf{A}] \in \mathbb{G}^{n_1 \times n_2})}$:
$\mathbf{B}_0, \mathbf{B}_1 \leftarrow \mathcal{D}_{2k,k}$; $H \xleftarrow{\$} \mathcal{H}$
$\mathsf{crs}_{or} \leftarrow \Pi_{or}.\mathsf{Gen}(1^\lambda, [\mathbf{B}_0], [\mathbf{B}_1])$
Return $\mathsf{pp} := ([\mathbf{A}], [\mathbf{B}_0], \mathsf{crs}_{or}, H)$

$\underline{\mathsf{Gen}(\mathsf{pp})}$:
$\mathbf{K}_0, \mathbf{K}_1 \xleftarrow{\$} \mathbb{Z}_q^{(m+1)\times n_1}$
$\widehat{\mathbf{K}} \xleftarrow{\$} \mathbb{Z}_q^{(m+1)\times 2k}$; $\mathbf{M} \xleftarrow{\$} \mathbb{Z}_q^{m\times(m+1)}$
$\mathsf{crs} := ([\mathbf{K}_0\mathbf{A}], [\mathbf{K}_1\mathbf{A}], [\widehat{\mathbf{K}}\mathbf{B}_0])$
$\mathsf{td} := (\mathbf{K}_0, \mathbf{K}_1)$
$msk := (\mathbf{K}_0, \mathbf{K}_1, \widehat{\mathbf{K}}, \mathbf{M})$
Return $(\mathsf{crs}, \mathsf{td}, msk)$

$\underline{\mathsf{Prove}(\mathsf{crs}, [\mathbf{c}], \mathbf{s}, \tau)}$: // $\mathbf{c} = \mathbf{A}\mathbf{s}$
$\mathbf{r} \xleftarrow{\$} \mathbb{Z}_q^k$; $[\mathbf{t}] := [\mathbf{B}_0]\mathbf{r}$
$\pi_{or} \leftarrow \Pi_{or}.\mathsf{Prove}(\mathsf{crs}_{or}, [\mathbf{t}], \mathbf{r})$
$\theta := H([\mathbf{c}], \tau, [\mathbf{t}], \pi_{or})$
$[\mathbf{u}] := [(\mathbf{K}_0 + \theta\mathbf{K}_1)\mathbf{A}]\mathbf{s} + [\widehat{\mathbf{K}}\mathbf{B}_0]\mathbf{r} \in \mathbb{G}^{m+1}$
Return $\pi := ([\mathbf{t}], [\mathbf{u}], \pi_{or})$

$\underline{\mathsf{MVer}(msk, [\mathbf{c}], \tau, \pi = ([\mathbf{t}], [\mathbf{u}], \pi_{or}))}$:
If $\Pi_{or}.\mathsf{Ver}(\mathsf{crs}_{or}, [\mathbf{t}], \pi_{or}) = 0$: return 0
$\theta := H([\mathbf{c}], \tau, [\mathbf{t}], \pi_{or})$
If $[\mathbf{u}] = (\mathbf{K}_0 + \theta\mathbf{K}_1)[\mathbf{c}] + \widehat{\mathbf{K}}[\mathbf{t}]$: return 1
Otherwise: return 0

$\underline{\mathsf{Sim}(\mathsf{td}, [\mathbf{c}], \tau)}$:
$\mathbf{r} \xleftarrow{\$} \mathbb{Z}_q^k$; $[\mathbf{t}] := [\mathbf{B}_0]\mathbf{r}$
$\pi_{or} \leftarrow \Pi_{or}.\mathsf{Prove}(\mathsf{crs}_{or}, [\mathbf{t}], \mathbf{r})$
$\theta := H([\mathbf{c}], \tau, [\mathbf{t}], \pi_{or})$
$[\mathbf{u}] := (\mathbf{K}_0 + \theta\mathbf{K}_1)[\mathbf{c}] + [\widehat{\mathbf{K}}\mathbf{B}_0]\mathbf{r} \in \mathbb{G}^{m+1}$
Return $\pi := ([\mathbf{t}], [\mathbf{u}], \pi_{or})$

$\underline{\mathsf{Delegate}(msk, \mathbf{d} \in \mathbb{Z}_q^m)}$:
Return $sk_{\mathbf{d}} := (\mathbf{d}^\top\mathbf{M}, \mathbf{d}^\top\mathbf{M}\mathbf{K}_0, \mathbf{d}^\top\mathbf{M}\mathbf{K}_1, \mathbf{d}^\top\mathbf{M}\widehat{\mathbf{K}})$

$\underline{\mathsf{FVer}(sk_{\mathbf{d}}, [\mathbf{c}], \tau, \pi = ([\mathbf{t}], [\mathbf{u}], \pi_{or}))}$:
If $\Pi_{or}.\mathsf{Ver}(\mathsf{crs}_{or}, [\mathbf{t}], \pi_{or}) = 0$: return 0
$\theta := H([\mathbf{c}], \tau, [\mathbf{t}], \pi_{or})$
If $\mathbf{d}^\top\mathbf{M}[\mathbf{u}] = \mathbf{d}^\top\mathbf{M}(\mathbf{K}_0 + \theta\mathbf{K}_1)[\mathbf{c}] + \mathbf{d}^\top\mathbf{M}\widehat{\mathbf{K}}[\mathbf{t}]$: return 1
Otherwise: return 0

Fig. 5. The pairing-based construction of tag-based FV-NIZK Π, where $\Pi_{or} = (\Pi_{or}.\mathsf{Gen}, \Pi_{or}.\mathsf{TGen}, \Pi_{or}.\mathsf{Prove}, \Pi_{or}.\mathsf{Sim}, \Pi_{or}.\mathsf{Ver})$ is a NIZK proof for OR-language $\mathscr{L}^\vee_{[\mathbf{B}_0],[\mathbf{B}_1]}$.

Completeness and perfect zero-knowledge follow directly from the fact that

$$\begin{aligned}\mathbf{u} &= (\mathbf{K}_0 + \theta\mathbf{K}_1)\mathbf{A}\mathbf{s} + \widehat{\mathbf{K}}\mathbf{B}_0\mathbf{r} = (\mathbf{K}_0 + \theta\mathbf{K}_1)\mathbf{c} + \widehat{\mathbf{K}}\mathbf{t} && \text{// completeness (1)}\\ &= (\mathbf{K}_0 + \theta\mathbf{K}_1)\mathbf{c} + \widehat{\mathbf{K}}\mathbf{B}_0\mathbf{r}, && \text{// perfect zero-knowledge}\end{aligned}$$

which implies $\mathbf{d}^\top\mathbf{M}\mathbf{u} = \mathbf{d}^\top\mathbf{M}(\mathbf{K}_0 + \theta\mathbf{K}_1)\mathbf{c} + \mathbf{d}^\top\mathbf{M}\widehat{\mathbf{K}}\mathbf{t}$. // completeness (2)

Next, we show the verification equivalence and almost tight strong USS of Π.

Theorem 4 (Verification Equivalence). *The tag-based FV-NIZK scheme Π in Fig. 5 has $(0, 1/q)$-verification equivalence.*

The proof is very similar to that of Theorem 1 and we show it in the full version.

Theorem 5 (Almost Tight Strong USS). *If the $\mathcal{D}_{2k,k}$-MDDH assumption holds in $\mathbb{G}$, $\mathcal{H}$ is a family of collision resistant hash functions, and Π_{or} is a NIZK proof for $\mathscr{L}^\vee_{[\mathbf{B}_0],[\mathbf{B}_1]}$ with completeness, perfect soundness and zero-knowledge, then the tag-based FV-NIZK scheme Π in Fig. 5 has strong USS. More precisely, for any adversary $\mathcal{A}$ against the strong USS security of Π,*

there exist algorithms $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ *s.t.* $\max(Time(\mathcal{B}_1), Time(\mathcal{B}_2), Time(\mathcal{B}_3)) \approx Time(\mathcal{A}) + (Q_{sim} + Q_{ver} + Q_{del}) \cdot \mathsf{poly}(\lambda)$*, and*

$$\begin{aligned}\mathsf{Adv}^{uss}_{\Pi,\mathcal{A}}(\lambda) \leq & \mathsf{Adv}^{cr}_{\mathcal{H},\mathcal{B}_1}(\lambda) + (2\mathfrak{n}+2) \cdot \mathsf{Adv}^{zk}_{\Pi_{or},\mathcal{B}_2}(\lambda) \\ & + (4k\mathfrak{n}+2k) \cdot \mathsf{Adv}^{mddh}_{\mathcal{D}_{2k,k},\mathbb{G},\mathcal{B}_3}(\lambda) + \tfrac{(\mathfrak{n}+1)(Q_{sim}Q_{ver}+4)}{q-1}.\end{aligned}$$

where $Q_{sim}, Q_{ver}, Q_{del}$ *denote the numbers of queries to* Sim, Ver, Delegate*, respectively, and* $\mathfrak{n} := \lceil \log Q_{sim} \rceil$.

The proof is provided in the full version due to space limitations.

Remark 7 (On the almost tightness of strong USS). Similar to Remark 6, the term $\frac{(\mathfrak{n}+1)(Q_{sim}Q_{ver}+4)}{q-1}$ in Theorem 5 does not affect the tightness of the reduction since it is statistically small. Moreover, k is the parameter of the MDDH assumption (e.g., $k = 1$ corresponds to the standard DDH assumption). Consequently, the strong USS has security loss factor $O(\mathfrak{n}) = O(\lceil \log Q_{sim} \rceil)$, which is $O(\log \lambda)$ for PPT adversaries due to $Q_{sim} = \mathsf{poly}(\lambda)$, and thus is almost tight.

Remark 8. We note that our tag-based FV-NIZK scheme Π in Fig. 5 does not achieve proof pseudorandomness, since its proof π contains a proof π_{or} of the underlying NIZK scheme Π_{or} which supports public verification, so that anyone who obtains crs_{or} from pp can check the validity of π_{or}.

5 Applications of FV-NIZK

In this section, we illustrate the usefulness of tag-based FV-NIZK by showing two applications, including CCA-secure IPFE in Subsect. 5.1 and CCA-secure fine-grained verifiable PKE (FV-PKE) in Subsect. 5.2.

By instantiating with the almost tightly secure FV-NIZK schemes constructed in Sect. 4, we immediately obtain IPFE and FV-PKE schemes that achieve almost tight mCCA (multi-challenge CCA) security. Moreover, the resulting schemes are either pairing-free (when using the FV-NIZK scheme in Subsect. 4.1), or use less pairing operations than existing works (when using the FV-NIZK scheme in Subsect. 4.2).

5.1 Almost Tightly mCCA-Secure IPFE Schemes

In [26], Liu et al. proposed the first almost tightly mCCA secure IPFE scheme, based on a tightly mCPA secure scheme [31] and an almost tightly secure QA-NIZK argument for linear subspace languages [4]. However, the QA-NIZK argument in [4] involves pairings, so does Liu et al.'s IPFE.

To reduce the number of pairing operations or even get rid of pairings, we replace the QA-NIZK with our tag-based FV-NIZK for linear subspace languages in the IPFE construction. When the tag-based FV-NIZK is instantiated with the construction in Subsect. 4.1, we obtain the first pairing-free IPFE scheme with almost tight mCCA security. When it is instantiated with the construction

in Subsect. 4.2, we obtain a pairing-based IPFE scheme that uses less pairing operations than [26].

Formally, we present the syntax of IPFE and its mCCA security in the full version and describe our IPFE construction as follows. Let $m, k, X, Y \in \mathbb{N}$, and let $\mathcal{D}_k$ be a matrix distribution. Let $\Pi = (\Pi.\mathsf{Par}, \Pi.\mathsf{Gen}, \Pi.\mathsf{Prove}, \Pi.\mathsf{MVer}, \Pi.\mathsf{Sim}, \Pi.\mathsf{Delegate}, \Pi.\mathsf{FVer})$ be a tag-based FV-NIZK for linear subspace language $\mathscr{L}_{[\mathbf{A}]}$ with tag space $\mathcal{T}$ and delegation space $\mathcal{D} = \mathbb{Z}_q^m$. Let $\mathcal{H} : \{0,1\}^* \to \mathcal{T}$ be a family of collision resistant hash functions. Our IPFE construction $\mathsf{IPFE}_{\mathsf{mcca}} = (\mathsf{Par}, \mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ is described in Fig. 6, where the message space is $[-X, X]^m \subseteq \mathbb{Z}_q^m$ and the inner product function is defined by $\mathbf{y} \in [-Y, Y]^m \subseteq \mathbb{Z}_q^m$. Similar to [31, 26], we require mXY to be a polynomial in λ.

The correctness of $\mathsf{IPFE}_{\mathsf{mcca}}$ follows from the completeness of Π and the fact that for $\mathbf{x} \in [-X, X]^m$ and $\mathbf{y} \in [-Y, Y]^m$, it holds

$$d = \mathbf{y}^\top(\mathbf{WAs} + \mathbf{x}) - \mathbf{y}^\top\mathbf{W}(\mathbf{As}) = \mathbf{y}^\top\mathbf{x} \in [-mXY, mXY].$$

$\underline{\mathsf{Par}(1^\lambda)}$:
$\widetilde{\mathbf{A}} \leftarrow \mathcal{D}_k$; $\mathbf{A} := \mathbf{I}_{km} \otimes \widetilde{\mathbf{A}}$
$\widehat{\mathsf{pp}} \leftarrow \Pi.\mathsf{Par}(1^\lambda, [\mathbf{A}])$, $H \xleftarrow{\$} \mathcal{H}$
Return $\mathsf{pp} := ([\widetilde{\mathbf{A}}], \widehat{\mathsf{pp}}, H)$

$\underline{\mathsf{Setup}(1^m, \mathsf{pp})}$:
$\mathbf{W} \xleftarrow{\$} \mathbb{Z}_q^{m \times k(k+1)m}$
$(\mathsf{crs}, \mathsf{td}, \widehat{msk}) \leftarrow \Pi.\mathsf{Gen}(\Pi.\mathsf{pp})$
Return $mpk := ([\mathbf{WA}], \mathsf{crs})$, $msk := (\mathbf{W}, \widehat{msk})$

$\underline{\mathsf{KeyGen}(msk, \mathbf{y} \in [-Y, Y]^m)}$:
$\widehat{sk}_{\mathbf{y}} \leftarrow \Pi.\mathsf{Delegate}(\widehat{msk}, \mathbf{y})$
Return $sk_{\mathbf{y}} := (\mathbf{y}, \mathbf{y}^\top\mathbf{W}, \widehat{sk}_{\mathbf{y}})$

$\underline{\mathsf{Enc}(mpk, \mathbf{x} \in [-X, X]^m)}$:
$\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^{k^2 m}$; $[\mathbf{c}] := [\mathbf{A}]\mathbf{s} \in \mathbb{G}^{k(k+1)m}$
$[\mathbf{v}] := [\mathbf{WA}]\mathbf{s} + [\mathbf{x}] \in \mathbb{Z}_q^m$
$\tau := H(mpk, [\mathbf{c}], [\mathbf{v}])$
$\pi \leftarrow \Pi.\mathsf{Prove}(\mathsf{crs}, [\mathbf{c}], \mathbf{s}, \tau)$
Return $ct := ([\mathbf{c}], [\mathbf{v}], \pi)$

$\underline{\mathsf{Dec}(sk_{\mathbf{y}}, ct)}$:
Parse $ct = ([\mathbf{c}], [\mathbf{v}], \pi)$
$\tau := H(mpk, [\mathbf{c}], [\mathbf{v}])$
If $\Pi.\mathsf{FVer}(\widehat{sk}_{\mathbf{y}}, [\mathbf{c}], \tau, \pi) = 1$:
  $[d] := \mathbf{y}^\top[\mathbf{v}] - \mathbf{y}^\top\mathbf{W}[\mathbf{c}]$
  Return $d \in [-mXY, mXY]$
Otherwise: return $\perp$

Fig. 6. Construction of $\mathsf{IPFE}_{\mathsf{mcca}}$ from tag-based FV-NIZK Π. For the ease of reading, we emphasize different parts with [26] in gray boxes.

Theorem 6 (Almost Tight mCCA Security of $\mathsf{IPFE}_{\mathsf{mcca}}$). *If the $\mathcal{D}_k$-MDDH assumption holds in $\mathbb{G}$, $\mathcal{H}$ is a family of collision resistant hash functions, and Π is a tag-based FV-NIZK with $(0, \epsilon)$-verification equivalence and strong USS, then $\mathsf{IPFE}_{\mathsf{mcca}}$ shown in Fig. 6 is mCCA-secure. Concretely, for any PPT adversary $\mathcal{A}$, there exist PPT algorithms $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ s.t. $\max(Time(\mathcal{B}_1), Time(\mathcal{B}_2), Time(\mathcal{B}_3)) \approx Time(\mathcal{A}) + (Q_{enc} + Q_{sk} + Q_{dec}) \cdot \mathsf{poly}(\lambda, m)$ with $\mathsf{poly}(\lambda, m)$ independent of $\mathcal{A}$, and*

$$\mathsf{Adv}^{mcca}_{\mathsf{IPFE}_{\mathsf{mcca}}, \mathcal{A}}(\lambda) \leq 2\mathsf{Adv}^{cr}_{\mathcal{H}, \mathcal{B}_1}(\lambda) + 4\mathsf{Adv}^{mddh}_{\mathcal{D}_k, \mathbb{G}, \mathcal{B}_2}(\lambda) + 2\mathsf{Adv}^{uss}_{\Pi, \mathcal{B}_3}(\lambda) + 2Q_{dec} \cdot \epsilon + \tfrac{2}{q-1},$$

where Q_{enc}, Q_{sk} and Q_{dec} denote the total numbers of encryption, key generation and decryption queries, respectively.

The proof is shown in the full version due to space limitations.

5.2 Almost Tightly mCCA-Secure FV-PKE Schemes

In this subsection, we formalize the new primitive called *Fine-grained Verifiable PKE (FV-PKE)*, and define verification soundness, mCCA security, and ciphertext pseudorandomness for it. Then we show how to construct FV-PKE based on our tag-based FV-NIZK. By instantiating with the almost tightly secure FV-NIZK scheme proposed in Subsect. 4.1, we obtain the first FV-PKE scheme with almost tight mCCA security and ciphertext pseudorandomness.

We first present the syntax of FV-PKE.

Definition 8 (FV-PKE). *A Fine-grained Verifiable Public-Key Encryption (FV-PKE) scheme consists of six PPT algorithms, namely* $\mathsf{FPKE} = (\mathsf{Par}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Delegate}, \mathsf{Ver})$.

- $\mathsf{pp} \leftarrow \mathsf{Par}(1^\lambda)$*: Initialization algorithm takes the security parameter λ as input and outputs a public parameter* pp*, which defines the message space $\mathcal{M}$ and the delegation space $\mathcal{D}$.*
- $(pk, sk) \leftarrow \mathsf{Gen}(\mathsf{pp})$*: Generation algorithm takes* pp *as inputs, and outputs a public key pk and a secret key sk. We assume pk contains* pp*, and it serves as an implicit input of* $\mathsf{Enc}, \mathsf{Dec}, \mathsf{Delegate}$*, and* Ver*.*
- $ct \leftarrow \mathsf{Enc}(pk, M)$*: Encryption algorithm takes pk and a message $M \in \mathcal{M}$ as inputs, and outputs a ciphertext ct.*
- $M'/\perp \leftarrow \mathsf{Dec}(sk, ct)$*: Decryption algorithm takes sk and a ciphertext ct as inputs, and outputs a message $M' \in \mathcal{M}$ or a special failure symbol $\perp$.*
- $sk_d \leftarrow \mathsf{Delegate}(sk, d)$*: Delegation algorithm takes sk and a delegation $d \in \mathcal{D}$ as inputs, and outputs a delegated secret key sk_d.*
- $0/1 \leftarrow \mathsf{Ver}(sk_d, ct)$*: Verification algorithm takes sk_d and ct as inputs, and outputs a bit indicating whether ct is a valid ciphertext or not.*

We require FPKE *to have decryption correctness and verification correctness.*

Decryption Correctness. *For all* pp*,* $(pk, sk) \leftarrow \mathsf{Gen}(\mathsf{pp})$*,* $M \in \mathcal{M}$ *and* $ct \leftarrow \mathsf{Enc}(pk, M)$*, it holds that* $\mathsf{Dec}(sk, ct) = M$*.*

Verification Correctness. *For all* pp*,* $(pk, sk) \leftarrow \mathsf{Gen}(\mathsf{pp})$*,* $M \in \mathcal{M}$ *and* $ct \leftarrow \mathsf{Enc}(pk, M)$*, it holds* $\mathsf{Ver}(sk_d, ct) = 1$ *for all* $sk_d \leftarrow \mathsf{Delegate}(sk, d)$ *of all* $d \in \mathcal{D}$*.*

Note that the first four algorithms $(\mathsf{Par}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ of FV-PKE basically constitute a standard PKE scheme. Moreover, the two additional algorithms $(\mathsf{Delegate}, \mathsf{Ver})$ provide the fine-grained ability for verifying ciphertext validity.

Next, we define a statistical property called *verification soundness* for FV-PKE. Loosely speaking, it essentially requires that for any ciphertext ct and any sk_d, $\mathsf{Ver}(sk_d, ct)$ outputs 1 if and only if ct is a valid ciphertext, i.e., $\mathsf{Dec}(sk, ct)$ succeeds, except for a negligible probability.

Definition 9 (Verification Soundness of FV-PKE). *Let $\delta, \epsilon > 0$. An FV-PKE scheme* FPKE *has (δ, ϵ)-verification soundness, if for any (even unbounded) adversary $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{ver\text{-}snd}_{\mathsf{FPKE},\mathcal{A},\delta}(\lambda) := \Pr[\mathsf{Exp}^{ver\text{-}snd}_{\mathsf{FPKE},\mathcal{A},\delta}(\lambda) \Rightarrow 1] \leq \epsilon,$$

where the experiment $\mathsf{Exp}^{ver\text{-}snd}_{\mathsf{FPKE},\mathcal{A},\delta}(\lambda)$ is defined in Fig. 7.

$\underline{\mathsf{Exp}^{ver\text{-}snd}_{\Pi,A,\delta}(\lambda)}$:
$\mathsf{pp} \leftarrow \mathsf{Par}(1^\lambda)$, $(pk, sk) \leftarrow \mathsf{Gen}(\mathsf{pp})$, $\mathcal{Q}_{sk} := \emptyset$
$(ct^*, d^*) \leftarrow \mathcal{A}^{\textsc{Delegate}(\cdot)}(\mathsf{pp}, pk)$
$sk_{d^*} \leftarrow \mathsf{Delegate}(sk, d^*)$

If $\widetilde{\mathbf{H}}_\infty(sk_{d^*}|pk, \mathcal{Q}_{sk}, d^*) > \delta$
$\wedge \begin{pmatrix} (\mathsf{Ver}(sk_{d^*}, ct^*) = 1 \ \wedge \ \mathsf{Dec}(sk, ct^*) = \perp) \\ \vee \ (\mathsf{Ver}(sk_{d^*}, ct^*) = 0 \ \wedge \ \mathsf{Dec}(sk, ct^*) \neq \perp) \end{pmatrix}$: output 1
Otherwise: output 0

$\underline{\textsc{Delegate}(d)}$:
$sk_d \leftarrow \mathsf{Delegate}(sk, d)$
$\mathcal{Q}_{sk} := \mathcal{Q}_{sk} \cup \{(d, sk_d)\}$
Return sk_d

Fig. 7. The verification soundness experiment $\mathsf{Exp}^{ver\text{-}snd}_{\mathsf{FPKE},\mathcal{A},\delta}(\lambda)$ for FV-PKE.

Remark 9 (On the formalization of verification soundness). We stress that we do not require Ver can always correctly decide whether a ciphertext is valid or not. That is, there might exist a ciphertext ct and a pair (d, sk_d) s.t., $\mathsf{Dec}(sk, ct) = \perp$ but $\mathsf{Ver}(sk_d, ct) = 1$, or $\mathsf{Dec}(sk, ct) \neq \perp$ but $\mathsf{Ver}(sk_d, ct) = 0$. Nevertheless, verification soundness of FV-PKE ensures that even for an (unbounded) adversary $\mathcal{A}$, if it does not get enough information about sk_{d^*} (and thus sk), it is hard for $\mathcal{A}$ to find a ct^* that makes $\mathsf{Dec}(sk, \cdot)$ and $\mathsf{Ver}(sk_{d^*}, \cdot)$ perform inconsistently. Similar to Remark 1, we require "$\widetilde{\mathbf{H}}_\infty(sk_{d^*}|pk, \mathcal{Q}_{sk}, d^*) > \delta$" in Fig. 7 to prevent trivial attacks, since for those who get sk_{d^*}, it might be easy for them to produce such a ct^*.

Remark 10 (On the motivation for defining FV-PKE with the delegation space $\mathcal{D}$). The main motivation for defining FV-PKE with the delegation d is to provide the flexibility of verification, which can be used to make the verification result closer to the validity of ciphertexts, as explained below. Let us go back to the motivating example described in the introduction, where a manager asks an assistant to filter out invalid ciphertexts. By using FV-PKE, the manager can give a delegated key sk_d to the assistant, and the property of verification soundness guarantees that verification using sk_d can correctly decide the validity for ciphertexts generated by the outsider (i.e., anyone other than the manager and the assistant). However, since the assistant has sk_d, it does not exclude the possibility that the assistant itself produces ill-formed ciphertexts which are invalid but pass the verification, or are valid but do not pass the verification. We refer to this as an "insider" attack.

Thanks to the fact that FV-PKE supports delegation d, such "insider" attacks can be easily prevented: the manager can ask several assistants, give them different delegated keys $(sk_{d_1}, sk_{d_2}, ...)$, and regard a ciphertext valid only if it passes all the verifications. As long as not all the assistants collude, it is hard for them to produce ill-formed ciphertexts which are invalid but pass all the verifications, or are valid but do not pass all the verifications. Of course, the manager can also set a threshold, and regard a ciphertext valid if the number of verifications that it passes is above the threshold, in order to tolerate inadvertent errors. This reflects the flexibility of verification. Stepping back, even if an "insider" attack occurs, the manager can identify which assistant produced the ill-formed ciphertexts, by tracing the delegation d from sk_d.

Then we formalize the mCCA security for FV-PKE. Compared to the CCA security for standard PKE, we also allow the adversary to obtain delegated keys sk_d with d of its choices.

Definition 10 (mCCA Security of FV-PKE). *An FV-PKE scheme* FPKE *is indistinguishable under chosen ciphertext attacks in the multi-challenge setting (mCCA), if for any PPT adversary* $\mathcal{A}$*, it holds that*

$$\mathsf{Adv}^{mcca}_{\mathsf{FPKE},\mathcal{A}}(\lambda) := \left|\Pr[\mathsf{Exp}^{mcca}_{\mathsf{FPKE},\mathcal{A},0}(\lambda) \Rightarrow 1] - \Pr[\mathsf{Exp}^{mcca}_{\mathsf{FPKE},\mathcal{A},1}(\lambda) \Rightarrow 1]\right| \leq \mathsf{negl}(\lambda),$$

where the experiments $\mathsf{Exp}^{mcca}_{\mathsf{FPKE},\mathcal{A},\beta}(\lambda)$ *($\beta \in \{0,1\}$) are defined in Fig. 8.*

$\underline{\mathsf{Exp}^{mcca}_{\mathsf{FPKE},\mathcal{A},\beta}(\lambda)}$: // $\beta \in \{0,1\}$	
$\mathsf{pp} \leftarrow \mathsf{Par}(1^\lambda)$, $(pk, sk) \leftarrow \mathsf{Gen}(\mathsf{pp})$ $\mathcal{Q}_{enc} := \emptyset$; $\mathcal{Q}_{sk} := \emptyset$ $\beta' \leftarrow \mathcal{A}^{\textsc{Enc}(\cdot,\cdot),\textsc{Dec}(\cdot),\textsc{Delegate}(\cdot)}(\mathsf{pp}, pk)$ Output β'	$\underline{\textsc{Enc}(M^0, M^1)}$: $ct \leftarrow \mathsf{Enc}(pk, M^\beta)$ $\mathcal{Q}_{enc} := \mathcal{Q}_{enc} \cup \{ct\}$ Return ct
$\underline{\textsc{Delegate}(d)}$: $sk_d \leftarrow \mathsf{Delegate}(sk, d)$ $\mathcal{Q}_{sk} := \mathcal{Q}_{sk} \cup \{(d, sk_d)\}$ Return sk_d	$\underline{\textsc{Dec}(ct)}$: If $ct \in \mathcal{Q}_{enc}$: return $\perp$ Return $\mathsf{Dec}(sk, ct)$

Fig. 8. The IND-mCCA security experiments $\mathsf{Exp}^{mcca}_{\mathsf{FPKE},\mathcal{A},\beta}(\lambda)$ for FV-PKE.

Finally, we define *ciphertext pseudorandomness* for FV-PKE, which requires the pseudorandomness of ciphertexts for PPT adversaries that are not given any secret key but allowed to access the decryption oracle. This clearly implies anonymity.

Definition 11 (Ciphertext Pseudorandomness of FV-PKE). *An FV-PKE scheme* FPKE *has ciphertext pseudorandomness in the multi-challenge setting, if for any PPT adversary* $\mathcal{A}$*, it holds that*

$$\mathsf{Adv}^{cp}_{\mathsf{FPKE},\mathcal{A}}(\lambda) := |\Pr[\mathsf{Exp}^{cp}_{\mathsf{FPKE},\mathcal{A},0}(\lambda) \Rightarrow 1] - \Pr[\mathsf{Exp}^{cp}_{\mathsf{FPKE},\mathcal{A},1}(\lambda) \Rightarrow 1]| \leq \mathsf{negl}(\lambda),$$

$\underline{\mathsf{Exp}^{cp}_{\mathsf{FPKE},\mathcal{A},\beta}(\lambda)}$: // $\beta \in \{0,1\}$	
$\mathsf{pp} \leftarrow \mathsf{Par}(1^\lambda)$, $(pk, sk) \leftarrow \mathsf{Gen}(\mathsf{pp})$, $\mathcal{Q}_{enc} := \emptyset$ $\beta' \leftarrow \mathcal{A}^{\textsc{Enc}(\cdot),\textsc{Dec}(\cdot)}(\mathsf{pp}, pk)$ Output β'	$\underline{\textsc{Enc}(M)}$: If $\beta = 0$: $ct \leftarrow \mathsf{Enc}(pk, M)$ If $\beta = 1$: $ct \xleftarrow{\$} \mathcal{CT}$ $\mathcal{Q}_{enc} := \mathcal{Q}_{enc} \cup \{ct\}$ Return ct
$\underline{\textsc{Dec}(ct)}$: If $ct \in \mathcal{Q}_{enc}$: return $\perp$ Return $\mathsf{Dec}(sk, ct)$	

Fig. 9. The ciphertext pseudorandomness experiments $\mathsf{Exp}^{cp}_{\mathsf{FPKE},\mathcal{A},\beta}(\lambda)$ for FV-PKE, where $\mathcal{CT}$ denotes the ciphertext space.

where the experiments $\mathsf{Exp}^{cp}_{\mathsf{FPKE},\mathcal{A},\beta}(\lambda)$ *(*$\beta \in \{0,1\}$*) are defined in Fig. 9.*

Construction of FV-PKE. Now we describe our FV-PKE construction as follows. Let $\Pi = (\Pi.\mathsf{Par}, \Pi.\mathsf{Gen}, \Pi.\mathsf{Prove}, \Pi.\mathsf{MVer}, \Pi.\mathsf{Sim}, \Pi.\mathsf{Delegate}, \Pi.\mathsf{FVer})$ be a tag-based FV-NIZK for linear subspace language $\mathscr{L}_{[\mathbf{A}]}$ with tag space $\mathcal{T}$ and delegation space $\mathcal{D}$. Let $\mathcal{H} : \{0,1\}^* \to \mathcal{T}$ be a family of collision resistant hash functions. Our FV-PKE construction $\mathsf{FPKE}_{\mathsf{mcca}} = (\mathsf{Par}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Delegate}, \mathsf{Ver})$ is described in Fig. 10, where the message space is $\mathbb{G}$ and the delegation space is $\mathcal{D}$.

The decryption correctness follows from the completeness (1) of Π and the fact that

$$[v] - \mathbf{w}^\top[\mathbf{c}] = ([\mathbf{w}^\top\mathbf{A}]\mathbf{s} + M) - \mathbf{w}^\top[\mathbf{As}] = M,$$

and the verification correctness follows from the completeness (2) of Π.

Theorem 7 (Verification Soundness of $\mathsf{FPKE}_{\mathsf{mcca}}$). *If Π is a tag-based FV-NIZK with (δ, ϵ)-verification equivalence, then* FPKE *shown in Fig. 10 has (δ, ϵ)-verification soundness.*

Proof. The proof is straightforward. Since Π has (δ, ϵ)-verification equivalence, the algorithms $\Pi.\mathsf{MVer}$ and $\Pi.\mathsf{FVer}$ perform identically, except with probability at most ϵ. Consequently, it is hard for an (even unbounded) adversary to find (ct^*, d^*) that passes the verification algorithm Ver of FPKE (i.e., passing $\Pi.\mathsf{FVer}$) but fails the decryption of ct^* (i.e., not passing $\Pi.\mathsf{MVer}$), or fails to pass Ver (i.e., not passing $\Pi.\mathsf{FVer}$) but decrypts successfully (i.e., passing $\Pi.\mathsf{MVer}$). □

Now we show that $\mathsf{FPKE}_{\mathsf{mcca}}$ has almost tight mCCA security and almost tight ciphertext pseudorandomness via the following two theorems.

Theorem 8 (Almost Tight mCCA Security of $\mathsf{FPKE}_{\mathsf{mcca}}$). *If the $\mathcal{D}_{2k,k}$-MDDH assumption holds in $\mathbb{G}$, $\mathcal{H}$ is a family of collision resistant hash functions, and Π is a tag-based FV-NIZK with strong USS, then* $\mathsf{FPKE}_{\mathsf{mcca}}$ *shown in Fig. 10 is mCCA-secure. Concretely, for any PPT adversary $\mathcal{A}$, there exist PPT algorithms $\mathcal{B}_1, \mathcal{B}_3, \mathcal{B}_3$ s.t.* $\max(Time(\mathcal{B}_1), Time(\mathcal{B}_2), Time(\mathcal{B}_3)) \approx Time(\mathcal{A}) + (Q_{enc} + Q_{sk} + Q_{dec}) \cdot \mathsf{poly}(\lambda)$ *with* $\mathsf{poly}(\lambda)$ *independent of $\mathcal{A}$, and*

$$\mathsf{Adv}^{mcca}_{\mathsf{FPKE}_{\mathsf{mcca}},\mathcal{A}}(\lambda) \leq 2\mathsf{Adv}^{cr}_{\mathcal{H},\mathcal{B}_1}(\lambda) + (2k+4)\mathsf{Adv}^{mddh}_{\mathcal{D}_{2k,k},\mathbb{G},\mathcal{B}_2}(\lambda) + 2\mathsf{Adv}^{uss}_{\Pi,\mathcal{B}_3}(\lambda) + \tfrac{6}{q-1},$$

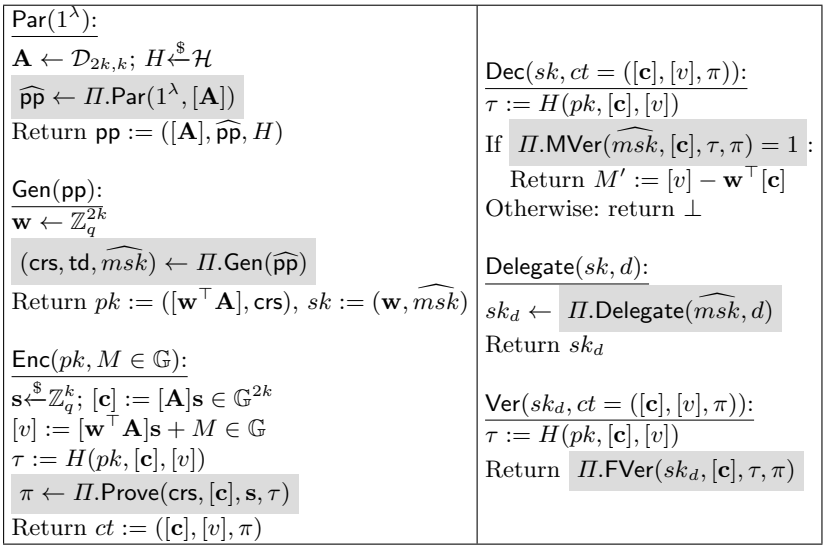

Fig. 10. Construction of $\mathsf{FPKE}_{\mathsf{mcca}}$ from tag-based FV-NIZK Π. For the ease of reading, we emphasize the parts related to Π in gray boxes.

where Q_{enc}, Q_{sk} and Q_{dec} denote the total numbers of encryption, delegation and decryption queries, respectively.

Proof. We prove the theorem via a series of games $\mathsf{G}_0^\beta, ..., \mathsf{G}_5^\beta$ ($\beta \in \{0,1\}$), where the first two games G_0^β are the mCCA experiments $\mathsf{Exp}^{mcca}_{\mathsf{FPKE},\mathcal{A},\beta}(\lambda)$ (cf. Fig. 8), and G_5^0, G_5^1 are identical.

Game G_0^β. They are just the original experiments $\mathsf{Exp}^{mcca}_{\mathsf{FPKE},\mathcal{A},\beta}(\lambda)$, except that we use secret key $\mathbf{w}$ to do the encryption. Due to the equation $[\mathbf{w}^\top \mathbf{A}]\mathbf{s} = \mathbf{w}^\top [\mathbf{A}\mathbf{s}] = \mathbf{w}^\top [\mathbf{c}]$, we have that

$$\Pr[\mathsf{Exp}^{mcca}_{\mathsf{FPKE},\mathcal{A},\beta}(\lambda) \Rightarrow 1] = \Pr[\mathsf{G}_0^\beta \Rightarrow 1], \text{ for } \beta \in \{0,1\}.$$

Game G_1^β. In this two games, whenever there is an encryption or decryption query with tag τ' that collides with some τ used in encryption before, the experiment returns $\perp$ and aborts. By the collision resistance of $\mathcal{H}$, we have

$$|\Pr[\mathsf{G}_0^\beta \Rightarrow 1] - \Pr[\mathsf{G}_1^\beta \Rightarrow 1]| \leq \mathsf{Adv}^{cr}_{\mathcal{H},\mathcal{B}_1}(\lambda), \text{ for } \beta \in \{0,1\}.$$

Game G_2^β. In this two games, $\textsc{Enc}(M^0, M^1)$ generates proofs π via $\Pi.\mathsf{Sim}(\mathsf{td}, \cdot, \cdot)$. G_1^β and G_2^β are the same due to the perfect zero-knowledge of Π, and we have

$$\Pr[\mathsf{G}_1^\beta \Rightarrow 1] = \Pr[\mathsf{G}_2^\beta \Rightarrow 1], \text{ for } \beta \in \{0,1\}.$$

Game G_3^β. In this two games, we sample $\mathbf{A}_0 \xleftarrow{\$} \mathbb{Z}_q^{2k \times k}$ in the beginning of the experiment. Meanwhile, $\textsc{Enc}(M^0, M^1)$ computes $[\mathbf{c}] := [\mathbf{A}_0]\mathbf{s}$, instead of $[\mathbf{c}] :=$

$[\mathbf{A}]\mathbf{s}$ for $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^k$. By the $\mathcal{D}_{2k,k}$-MDDH assumption and Lemma 1, we have

$$|\Pr[\mathsf{G}_2^\beta \Rightarrow 1] - \Pr[\mathsf{G}_3^\beta \Rightarrow 1]| \leq (k+1)\mathsf{Adv}^{mddh}_{\mathcal{D}_{2k,k},\mathbb{G},\mathcal{B}_3} + \tfrac{2}{q-1}, \text{ for } \beta \in \{0,1\}.$$

Game G_4^β. In this two games, the decryption oracle $\textsc{Dec}([\mathbf{c}^*],[v^*],\pi^*)$ returns $\perp$ directly if $([\mathbf{c}^*],[v^*],\pi^*) \notin \mathcal{Q}_{enc}$ and $[\mathbf{c}^*] \notin \mathscr{L}_{[\mathbf{A}]}$.

Define by bad the event that there exists a query $\textsc{Dec}([\mathbf{c}^*],[v^*],\pi^*)$, such that $([\mathbf{c}^*],[v^*],\pi^*) \notin \mathcal{Q}_{enc}$, $[\mathbf{c}^*] \notin \mathscr{L}_{[\mathbf{A}]}$, and there is no hash collision, but $\mathit{\Pi}.\mathsf{MVer}(\widehat{msk},[\mathbf{c}^*],\tau^*,\pi^*) = 1$, where $\tau^* := H(pk,[\mathbf{c}^*],[v^*])$. Obviously, G_3^β and G_4^β are identical unless bad happens. Thanks to the strong USS of $\mathit{\Pi}$, we have the following lemma.

Lemma 3. *For* $\beta \in \{0,1\}$, $|\Pr[\mathsf{G}_3^\beta \Rightarrow 1] - \Pr[\mathsf{G}_4^\beta \Rightarrow 1]| \leq \Pr[\mathsf{bad}] \leq \mathsf{Adv}^{uss}_{\mathit{\Pi},\mathcal{B}_4}(\lambda)$.

Game G_5^β. In this two games, $\textsc{Enc}(M^0,M^1)$ uniformly samples $[\mathbf{c}] \xleftarrow{\$} \mathbb{G}^{2k}$ and $[v] \xleftarrow{\$} \mathbb{G}$, instead of computing $[\mathbf{c}] := [\mathbf{A}_0]\mathbf{s}$ for $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^k$ and $[v] := \mathbf{w}^\top[\mathbf{c}] + M^\beta$.

Lemma 4. *For* $\beta \in \{0,1\}$, $|\Pr[\mathsf{G}_4^\beta \Rightarrow 1] - \Pr[\mathsf{G}_5^\beta \Rightarrow 1]| \leq \mathsf{Adv}^{mddh}_{\mathcal{U}_k,\mathbb{G},\mathcal{B}_5} + \frac{1}{q-1}$.

Proof. First we argue that in G_4^β, $\mathbf{w}$ still contains some entropy which is not leaked via pk and $\textsc{Dec}(\cdot,\cdot,\cdot)$. Then we show that the left entropy helps us change $[\mathbf{c}]$ from $[\mathbf{c}] := [\mathbf{A}_0]\mathbf{s}$ to $[\mathbf{c}] \xleftarrow{\$} \mathbb{G}^{2k}$, and change $[v]$ from $[v] := \mathbf{w}^\top[\mathbf{c}] + M^\beta$ to $[v] \xleftarrow{\$} \mathbb{G}$, based on the Q_{sim}-fold $\mathcal{U}_{2k+1,k}$-MDDH assumption.

To see this, we redefine $\mathbf{w}^\top$ as $\mathbf{w}^\top := \mathbf{w'}^\top + \mathbf{z}^\top\mathbf{A}^\perp$, where $\mathbf{w'} \xleftarrow{\$} \mathbb{Z}_q^{2k}$, $\mathbf{z} \xleftarrow{\$} \mathbb{Z}_q^k$, and $\mathbf{A}^\perp \xleftarrow{\$} \mathbb{Z}_q^{k\times 2k}$ s.t. $\mathbf{A}^\perp\mathbf{A} = \mathbf{0}$. We argue that the information of $\mathbf{z}$ is totally hidden to $\mathcal{A}$.

- pk hides the information of $\mathbf{z}$, due to
$$\mathbf{w}^\top\mathbf{A} = (\mathbf{w'}^\top + \mathbf{z}^\top\mathbf{A}^\perp)\mathbf{A} = \mathbf{w'}^\top\mathbf{A}.$$
- $\textsc{Delegate}(\cdot)$ hides the information of $\mathbf{z}$, since it does not involve $\mathbf{w}$ at all.
- $\textsc{Dec}([\mathbf{c}^*],[v^*],\pi^*)$ hides the information of $\mathbf{z}$. Thanks to the new rejection rule added in G_4, we have $[\mathbf{c}^*] \in \mathscr{L}_{[\mathbf{A}]}$ as otherwise $\textsc{Dec}([\mathbf{c}^*],[v^*],\pi^*)$ returns $\perp$ immediately. Therefore, $\mathbf{A}^\perp[\mathbf{c}^*] = [\mathbf{0}]$, and
$$\mathbf{w}^\top[\mathbf{c}^*] = (\mathbf{w'}^\top + \mathbf{z}^\top\mathbf{A}^\perp)[\mathbf{c}^*] = \mathbf{w'}^\top[\mathbf{c}^*].$$

With overwhelming probability we have $\mathbf{A}^\perp\mathbf{A}_0 \neq \mathbf{0}$. That is, $\mathbf{z}^\top\mathbf{A}^\perp\mathbf{A}_0$ is a random value over $\mathbb{Z}_q^{1\times k}$ from $\mathcal{A}$'s view. According to the Q_{sim}-fold $\mathcal{U}_{2k+1,k}$-MDDH assumption (equivalently the $\mathcal{U}_k$-MDDH assumption due to Lemma 1 and Lemma 2), we know the following two distributions are computationally indistinguishable:

$$\{[\mathbf{A}_0\mathbf{s}_j],[\mathbf{z}^\top\mathbf{A}^\perp\mathbf{A}_0\mathbf{s}_j]\}_{j\in[Q_{sim}]} \overset{c}{\approx} \{[\mathbf{c}'_j],[v'_j]\}_{j\in[Q_{sim}]},$$

where $\mathbf{s}_j \xleftarrow{\$} \mathbb{Z}_q^k, \mathbf{c}'_j \xleftarrow{\$} \mathbb{Z}_q^{2k}, v'_j \xleftarrow{\$} \mathbb{Z}_q$ for $1 \le j \le Q_{sim}$.

Recall that in G_4^β, $\textsc{Enc}(M_0, M_1)$ computes $[\mathbf{c}], [v]$ as $[\mathbf{c}] := [\mathbf{A}_0]\mathbf{s}$ and $[v] := \mathbf{w}^\top[\mathbf{c}] + M^\beta = \mathbf{w}'^\top[\mathbf{c}] + M^\beta + \mathbf{z}^\top\mathbf{A}^\perp[\mathbf{A}_0\mathbf{s}]$, which are indistinguishable from $[\mathbf{c}] \xleftarrow{\$} \mathbb{G}^{2k}$ and $[v] \xleftarrow{\$} \mathbb{G}$ according to the formula above. Then by Lemma 1, Lemma 4 holds as a result. ∎

Obviously G_5^0 and G_5^1 are identical. At last, thanks to Lemma 2, Theorem 8 follows by taking all things together. □

Theorem 9 (Almost Tight Ciphertext Pseudorandomness of $\mathsf{FPKE}_{\mathsf{mcca}}$). *If the $\mathcal{D}_{2k,k}$-MDDH assumption holds in $\mathbb{G}$, $\mathcal{H}$ is a family of collision resistant hash functions, and Π is a tag-based FV-NIZK with strong USS and proof pseudorandomness, then $\mathsf{FPKE}_{\mathsf{mcca}}$ shown in Fig. 10 has ciphertext pseudorandomness. Concretely, for any PPT adversary $\mathcal{A}$, there exist PPT algorithms $\mathcal{B}_1, ..., \mathcal{B}_4$ s.t. $\max(Time(\mathcal{B}_1), ..., Time(\mathcal{B}_4)) \approx Time(\mathcal{A}) + (Q_{enc} + Q_{dec}) \cdot \mathsf{poly}(\lambda)$ with $\mathsf{poly}(\lambda)$ independent of $\mathcal{A}$, and*

$$\begin{aligned}\mathsf{Adv}^{cp}_{\mathsf{FPKE}_{\mathsf{mcca}},\mathcal{A}}(\lambda) \le & 2\mathsf{Adv}^{cr}_{\mathcal{H},\mathcal{B}_1}(\lambda) + (2k+2)\mathsf{Adv}^{mddh}_{\mathcal{D}_{2k,k},\mathbb{G},\mathcal{B}_2}(\lambda) + 2\mathsf{Adv}^{uss}_{\Pi,\mathcal{B}_3}(\lambda) \\ & + \mathsf{Adv}^{pp}_{\Pi,\mathcal{B}_4}(\lambda) + \tfrac{4}{q-1},\end{aligned}$$

where Q_{enc} and Q_{dec} denote the total numbers of encryption and decryption queries, respectively.

Proof. Theorem 9 is proved via a series of games $\mathsf{G}_0, ..., \mathsf{G}_8$, where G_0 is the ciphertext pseudorandomness experiment $\mathsf{Exp}^{cp}_{\mathsf{FPKE},\mathcal{A},0}(\lambda)$ (cf. Fig. 9), and G_8 is indistinguishable with $\mathsf{Exp}^{cp}_{\mathsf{FPKE},\mathcal{A},1}(\lambda)$.

Due to the page limitation, we safely omit the descriptions of games $\mathsf{G}_0, ..., \mathsf{G}_5$, since they are similar with those in the proof of Theorem 8.

Game G_6. In this game, we eliminate the additional check $[\mathbf{c}^*] \in \mathsf{Span}([\mathbf{A}])$. Similar to the change from G_3 to G_4, due to the strong USS of Π, we have that

$$|\Pr[\mathsf{G}_5 \Rightarrow 1] - \Pr[\mathsf{G}_6 \Rightarrow 1]| \le \mathsf{Adv}^{uss}_{\Pi,\mathcal{B}_6}(\lambda).$$

Game G_7. In this game, $\textsc{Enc}(M)$ computes $[\mathbf{c}] := [\mathbf{A}]\mathbf{s}$ for $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^k$, instead of $[\mathbf{c}] \xleftarrow{\$} \mathbb{G}^{2k}$. By the $\mathcal{D}_{2k,k}$-MDDH assumption and Lemma 1, we have

$$|\Pr[\mathsf{G}_6 \Rightarrow 1] - \Pr[\mathsf{G}_7 \Rightarrow 1]| \le k\mathsf{Adv}^{mddh}_{\mathcal{D}_{2k,k},\mathbb{G},\mathcal{B}_7} + \tfrac{1}{q-1}.$$

Game G_8. In this game, $\textsc{Enc}(M)$ uniformly samples $[\mathbf{c}] \xleftarrow{\$} \mathbb{G}^{2k}$ and $\pi \xleftarrow{\$} \mathcal{P}$ instead of $[\mathbf{c}] := [\mathbf{A}]\mathbf{s}$ for $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^k$ and $\pi \leftarrow \Pi.\mathsf{Sim}(\mathsf{td}, [\mathbf{c}], \tau)$, where $\mathcal{P}$ denotes the proof space of Π.

Lemma 5. $|\Pr[\mathsf{G}_7 \Rightarrow 1] - \Pr[\mathsf{G}_8 \Rightarrow 1]| \le \mathsf{Adv}^{pp}_{\Pi,\mathcal{B}_8}(\lambda)$.

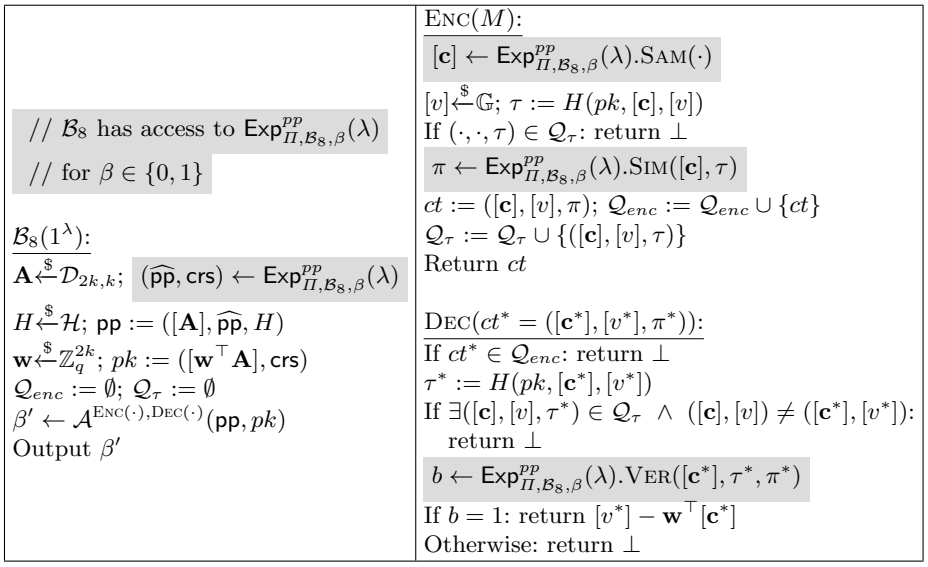

Fig. 11. $\mathcal{B}_8$'s reduction for the proof of Lemma 5.

Proof. We construct a reduction algorithm $\mathcal{B}_8$ to distinguish $\mathsf{Exp}^{pp}_{\Pi,\mathcal{B}_8,0}(\lambda)$ from $\mathsf{Exp}^{pp}_{\Pi,\mathcal{B}_8,1}(\lambda)$ for the proof pseudorandomness security of Π (cf. Fig. 3), as shown in Fig. 11. Recall that $\mathcal{B}_8$ has access to three oracles $\textsc{Sam}$, $\textsc{Sim}$, and $\textsc{Ver}$ in $\mathsf{Exp}^{pp}_{\Pi,\mathcal{B}_8,\beta}(\lambda)$.

Obviously, if $\mathcal{B}_8$ has access to $\mathsf{Exp}^{pp}_{\Pi,\mathcal{B}_8,0}(\lambda)$, then it simulates G_7 for $\mathcal{A}$; and if $\mathcal{B}_8$ has access to $\mathsf{Exp}^{pp}_{\Pi,\mathcal{B}_8,1}(\lambda)$, then it simulates G_8 for $\mathcal{A}$. Lemma 5 holds as a result. ∎

From G_8 to $\mathsf{Exp}^{pp}_{\mathsf{FPKE},\mathcal{A},1}(\lambda)$, we eliminate the additional check of hash collisions in $\textsc{Enc}(M)$ and $\textsc{Dec}(ct^*)$. With the same analysis we have

$$|\Pr[\mathsf{G}_8 \Rightarrow 1] - \Pr[\mathsf{Exp}^{cp}_{\mathsf{FPKE},\mathcal{A},1}(\lambda) \Rightarrow 1]| \leq \mathsf{Adv}^{cr}_{\mathcal{H},\mathcal{B}'_8}(\lambda).$$

Finally, taking Lemma 2 and all things together, Theorem 9 follows. □

Remark 11 (Extension to the multi-user setting). For better readability, we prove the almost tight mCCA security and ciphertext pseudorandomness of $\mathsf{FPKE}_{\mathsf{mcca}}$ in the single-user setting in Theorem 8 and Theorem 9. Now we show how to extend the proof techniques to the multi-user setting. More precisely, the public parameter $\mathsf{pp} = ([\mathbf{A}], \widehat{\mathsf{pp}}, H)$ is shared among all users, and each user $i \in [\mu]$ samples its own master secret key $(\mathbf{w}^{(i)}, \widehat{msk}^{(i)})$. In all computational steps in the proof, we modify all samples of $[\mathbf{c}]$ simultaneously, based on the random self-reducibility of the MDDH assumption. Moreover, the underlying FV-NIZK scheme Π is required to have almost tight strong USS and

proof pseudorandomness in the multi-user setting, which is satisfied by the first construction in Subsect. 4.1.

Acknowledgments. We would like to thank the anonymous reviewers for their valuable comments and suggestions. Shengli Liu and Xiangyu Liu were partially supported by National Natural Science Foundation of China (NSFC No. 61925207), Guangdong Major Project of Basic and Applied Basic Research (2019B030302008), and the National Key R&D Program of China under Grant 2022YFB2701500. Shuai Han was partially supported by National Natural Science Foundation of China (Grant No. 62002223), Shanghai Sailing Program (20YF1421100), Young Elite Scientists Sponsorship Program by China Association for Science and Technology (YESS20200185), and Ant Group through CCF-Ant Research Fund (CCF-AFSG RF20220224). Dawu Gu is partially supported by the National Key Research and Development Project (Grant No. 2020YFA0712302).

References

[1] Abdalla, M., Bourse, F., Caro, A.D., Pointcheval, D.: Simple functional encryption schemes for inner products. In: PKC 2015. vol. 9020, pp. 733–751 (2015)
[2] Abdalla, M., Bourse, F., Caro, A.D., Pointcheval, D.: Better security for functional encryption for inner product evaluations. IACR Cryptol. ePrint Arch. 2016, 11 (2016)
[3] Abe, M., David, B., Kohlweiss, M., Nishimaki, R., Ohkubo, M.: Tagged one-time signatures: Tight security and optimal tag size. In: PKC 2013. vol. 7778, pp. 312–331 (2013)
[4] Abe, M., Jutla, C.S., Ohkubo, M., Pan, J., Roy, A., Wang, Y.: Shorter QA-NIZK and SPS with tighter security. In: ASIACRYPT 2019. vol. 11923, pp. 669–699 (2019)
[5] Abe, M., Jutla, C.S., Ohkubo, M., Roy, A.: Improved (almost) tightly-secure simulation-sound QA-NIZK with applications. In: ASIACRYPT 2018. pp. 627–656 (2018)
[6] Agrawal, S., Libert, B., Stehlé, D.: Fully secure functional encryption for inner products, from standard assumptions. In: CRYPTO 2016. pp. 333–362 (2016)
[7] Bellare, M., Goldwasser, S.: New paradigms for digital signatures and message authentication based on non-interative zero knowledge proofs. In: CRYPTO 1989. vol. 435, pp. 194–211 (1989)
[8] Blazy, O., Kakvi, S.A., Kiltz, E., Pan, J.: Tightly-secure signatures from chameleon hash functions. In: PKC 2015. pp. 256–279 (2015)
[9] Blazy, O., Kiltz, E., Pan, J.: (Hierarchical) identity-based encryption from affine message authentication. In: CRYPTO 2014. pp. 408–425 (2014)
[10] Boneh, D., Sahai, A., Waters, B.: Functional encryption: Definitions and challenges. In: TCC 2011. vol. 6597, pp. 253–273 (2011)
[11] Chen, J., Wee, H.: Fully, (almost) tightly secure IBE and dual system groups. In: CRYPTO 2013. vol. 8043, pp. 435–460 (2013)
[12] Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: EUROCRYPT 2002. vol. 2332, pp. 45–64 (2002)

[13] Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.D.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM J. Comput. 38(1), 97–139 (2008)
[14] Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography (extended abstract). In: STOC 1991. pp. 542–552 (1991)
[15] Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.L.: An algebraic framework for diffie-hellman assumptions. In: CRYPTO 2013. vol. 8043, pp. 129–147 (2013)
[16] Gay, R., Hofheinz, D., Kiltz, E., Wee, H.: Tightly cca-secure encryption without pairings. In: EUROCRYPT 2016. vol. 9665, pp. 1–27 (2016)
[17] Gay, R., Hofheinz, D., Kohl, L.: Kurosawa-desmedt meets tight security. In: CRYPTO 2017. vol. 10403, pp. 133–160 (2017)
[18] Groth, J., Ostrovsky, R., Sahai, A.: New techniques for noninteractive zero-knowledge. J. ACM 59(3), 11:1–11:35 (2012)
[19] Han, S., Jager, T., Kiltz, E., Liu, S., Pan, J., Riepel, D., Schäge, S.: Authenticated key exchange and signatures with tight security in the standard model. In: CRYPTO 2021. vol. 12828, pp. 670–700 (2021)
[20] Han, S., Liu, S., Lyu, L., Gu, D.: Tight leakage-resilient cca-security from quasi-adaptive hash proof system. In: CRYPTO 2019. vol. 11693, pp. 417–447 (2019)
[21] Hofheinz, D., Jager, T.: Tightly secure signatures and public-key encryption. Des. Codes Cryptogr. 80(1), 29–61 (2016)
[22] Hofheinz, D., Jia, D., Pan, J.: Identity-based encryption tightly secure under chosen-ciphertext attacks. In: ASIACRYPT 2018. vol. 11273, pp. 190–220 (2018)
[23] Jutla, C.S., Roy, A.: Shorter quasi-adaptive NIZK proofs for linear subspaces. In: ASIACRYPT 2013. vol. 8269, pp. 1–20 (2013)
[24] Libert, B., Joye, M., Yung, M., Peters, T.: Concise multi-challenge CCA-secure encryption and signatures with almost tight security. In: ASIACRYPT 2014. pp. 1–21 (2014)
[25] Libert, B., Peters, T., Joye, M., Yung, M.: Compactly hiding linear spans - tightly secure constant-size simulation-sound QA-NIZK proofs and applications. In: ASIACRYPT 2015. vol. 9452, pp. 681–707 (2015)
[26] Liu, X., Liu, S., Han, S., Gu, D.: Tightly CCA-secure inner product functional encryption scheme. Theor. Comput. Sci. 898, 1–19 (2022)
[27] Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: STOC 1990. pp. 427–437 (1990)
[28] O'Neill, A.: Definitional issues in functional encryption. IACR Cryptol. ePrint Arch. 2010, 556 (2010)
[29] Ràfols, C.: Stretching Groth-Sahai: NIZK proofs of partial satisfiability. In: TCC 2015. vol. 9015, pp. 247–276 (2015)
[30] Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: FOCS 1999. pp. 543–553 (1999)
[31] Tomida, J.: Tightly secure inner product functional encryption: Multi-input and function-hiding constructions. In: ASIACRYPT 2019. pp. 459–488 (2019)