A Shuffle Argument Secure in the Generic Model

Prastudy Fauzi, Helger Lipmaa, and Michał Zając

University of Tartu, Estonia

Abstract. Implementation details of the Asiacrypt 2016 paper

1 Preliminaries

Let S_n be the symmetric group on n elements. For a (Laurent) polynomial or a rational function f and its monomial μ, denote by $\mathrm{coeff}_\mu(f)$ the coefficient of μ in f. We write $f(\kappa) \approx_\kappa g(\kappa)$, if $f(\kappa) - g(\kappa)$ is negligible as a function of κ.

Bilinear Maps. Let κ be the security parameter. Let q be a prime of length $O(\kappa)$ bits. Assume we use a secure bilinear group generator $\mathsf{genbp}(1^\kappa)$ that returns $\mathsf{gk} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e})$, where $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are three multiplicative groups of order q, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Within this paper, we denote the elements of $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ as in $\mathfrak{g}_1$ (i.e., by using the Fraktur typeface). It is required that $\hat{e}$ is bilinear (i.e., $\hat{e}(\mathfrak{g}_1^a, \mathfrak{g}_2^b) = \hat{e}(\mathfrak{g}_1, \mathfrak{g}_2)^{ab}$), efficiently computable, and non-degenerate. We define $\hat{e}((\mathfrak{A}_1, \mathfrak{A}_2, \mathfrak{A}_3), \mathfrak{B}) = (\hat{e}(\mathfrak{A}_1, \mathfrak{B}), \hat{e}(\mathfrak{A}_2, \mathfrak{B}), \hat{e}(\mathfrak{A}_3, \mathfrak{B}))$ and $\hat{e}(\mathfrak{B}, (\mathfrak{A}_1, \mathfrak{A}_2, \mathfrak{A}_3)) = (\hat{e}(\mathfrak{B}, \mathfrak{A}_1), \hat{e}(\mathfrak{B}, \mathfrak{A}_2), \hat{e}(\mathfrak{B}, \mathfrak{A}_3))$. Assume that $\mathfrak{g}_i$ is a generator of $\mathbb{G}_i$ for $i \in \{1, 2\}$, and set $\mathfrak{g}_T \leftarrow \hat{e}(\mathfrak{g}_1, \mathfrak{g}_2)$.

For $\kappa = 128$, the current recommendation is to use an optimal (asymmetric) Ate pairing over a subclass of Barreto-Naehrig curves. In that case, at security level of $\kappa = 128$, an element of $\mathbb{G}_1/\mathbb{G}_2/\mathbb{G}_T$ can be represented in respectively 256/512/3072 bits.

Cryptosystems. A public-key cryptosystem Π is a triple $(\mathsf{genpkc}, \mathsf{enc}, \mathsf{dec})$ of efficient algorithms. The key generation algorithm $\mathsf{genpkc}(1^\kappa)$ returns a fresh public and secret key pair $(\mathsf{pk}, \mathsf{sk})$. The encryption algorithm $\mathsf{enc}_{\mathsf{pk}}(m; r)$, given a public key pk, a message m, and a randomizer r (from some randomizer space $\mathcal{R}$), returns a ciphertext. The decryption algorithm $\mathsf{dec}_{\mathsf{sk}}(c)$, given a secret key sk and a ciphertext c, returns a plaintext m. It is required that for each $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{genpkc}(1^\kappa)$ and each m, r, it holds that $\mathsf{dec}_{\mathsf{sk}}(\mathsf{enc}_{\mathsf{pk}}(m; r)) = m$. Informally, Π is *IND-CPA secure*, if the distributions of ciphertexts corresponding to any two plaintexts are computationally indistinguishable.

We will use the ILin cryptosystem from [18]; it is distinguished from other well-known cryptosystems like the BBS cryptosystem [9] by having shorter secret and public keys. Consider group $\mathbb{G}_k$, $k \in \{1, 2\}$. In this cryptosytem, where the secret key is $\mathsf{sk} = \gamma \leftarrow_r \mathbb{Z}_q \setminus \{0, -1\}$, the public key is $\mathsf{pk}_k \leftarrow (\mathfrak{g}_k, \mathfrak{h}_k) = (\mathfrak{g}_k, \mathfrak{g}_k^\gamma)$, and the encryption of a small $m \in \mathbb{Z}_q$ is

$$\mathsf{enc}_{\mathsf{pk}_k}(m; \boldsymbol{s}) := (\mathfrak{h}_k^{s_1}, (\mathfrak{g}_k \mathfrak{h}_k)^{s_2}, \mathfrak{g}_k^m \mathfrak{g}_k^{s_1 + s_2})$$

for $\boldsymbol{s} \leftarrow_r \mathbb{Z}_q^{1 \times 2}$. Denote $\mathfrak{P}_{k1} := (\mathfrak{h}_k, \mathfrak{1}_k, \mathfrak{g}_k)$ and $\mathfrak{P}_{k2} := (\mathfrak{1}_k, \mathfrak{g}_k \mathfrak{h}_k, \mathfrak{g}_k)$, thus $\mathsf{enc}_{\mathsf{pk}_k}(m; \boldsymbol{s}) = (\mathfrak{1}_k, \mathfrak{1}_k, \mathfrak{g}_k^m) \cdot \mathfrak{P}_{k1}^{s_1} \mathfrak{P}_{k_2}^{s_2}$. Given $\mathfrak{v} \in \mathbb{G}_k^3$, the decryption sets

$$\mathsf{dec}_{\mathsf{sk}}(\mathfrak{v}) := \log_{\mathfrak{g}_k}(\mathfrak{v}_3 \mathfrak{v}_2^{-1/(\gamma+1)} \mathfrak{v}_1^{-1/\gamma}) \enspace ,$$

Decryption succeeds since $\mathfrak{v}_3 \mathfrak{v}_2^{-1/(\gamma+1)} \mathfrak{v}_1^{-1/\gamma} = \mathfrak{g}_k^m \mathfrak{g}_k^{s_1+s_2} \cdot (\mathfrak{g}_k \mathfrak{h}_k)^{-s_2/(\gamma+1)} \cdot \mathfrak{h}_k^{-s_1/\gamma} = \mathfrak{g}_k^m \mathfrak{g}_k^{s_1+s_2} \cdot \mathfrak{g}_k^{-s_2/(\gamma+1)} \mathfrak{g}_k^{-s_2 \cdot \gamma/(\gamma+1)} \cdot \mathfrak{g}_k^{-s_1} = \mathfrak{g}_k^m$. This cryptosystem is CPA-secure under the 2-Incremental Linear (2-ILin) assumption, see [18]. The ILin cryptosystem is *blindable*, $\mathsf{enc}_{\mathsf{pk}_k}(m; \boldsymbol{s}) \cdot \mathsf{enc}_{\mathsf{pk}_k}(0; \boldsymbol{s}') = \mathsf{enc}_{\mathsf{pk}}(m; \boldsymbol{s} + \boldsymbol{s}')$.

We use a variant of the ILin cryptosystem where each plaintext is encrypted twice, in group $\mathbb{G}_1$ and in $\mathbb{G}_2$ (but by using the same secret key an the same randomizer $\boldsymbol{s}$ in both). For technical reasons (relevant to the shuffle argument but not to the ILin cryptosystem), in group $\mathbb{G}_1$ we will use an auxiliary generator $\hat{\mathfrak{g}}_1 = \mathfrak{g}_1^{\varrho/\beta}$ instead of $\mathfrak{g}_1$, for $(\varrho, \beta) \leftarrow_r (\mathbb{Z}_q \setminus \{0\})^2$; both encryption and decryption are done as before but just using the secret key $\mathsf{sk} = (\varrho, \beta, \gamma)$ and the public key $\mathsf{pk}_1 = (\hat{\mathfrak{g}}_1, \mathfrak{h}_1 = \hat{\mathfrak{g}}_1^{\gamma})$; this also redefines $\mathfrak{P}_{k1}$. That is, $\mathsf{enc}_{\mathsf{pk}}(m; \boldsymbol{s}) = (\mathsf{enc}_{\mathsf{pk}_1}(m; \boldsymbol{s}), \mathsf{enc}_{\mathsf{pk}_2}(m; \boldsymbol{s}))$, where $\mathsf{pk}_1 = (\hat{\mathfrak{g}}_1, \mathfrak{h}_1 = \hat{\mathfrak{g}}_1^{\gamma})$, and $\mathsf{pk}_2 = (\mathfrak{g}_2, \mathfrak{h}_2 = \mathfrak{g}_2^{\gamma})$, and $\mathsf{dec}_{\mathsf{sk}}(\mathfrak{v}) := \log_{\hat{\mathfrak{g}}_1}(\mathfrak{v}_3 \mathfrak{v}_2^{-1/(\gamma+1)} \mathfrak{v}_1^{-1/\gamma}) = \log_{\mathfrak{g}_1}(\mathfrak{v}_3 \mathfrak{v}_2^{-1/(\gamma+1)} \mathfrak{v}_1^{-1/\gamma})/(\varrho/\beta)$ for $\mathfrak{v} \in \mathbb{G}_1^3$. We call this the *validity-enhanced* ILin cryptosystem.

In this case we denote the ciphertext in group k by $\mathfrak{v}_k$, and its jth component by $\mathfrak{v}_{kj}$. In the case when we have many ciphertexts, we denote the ith ciphertext by $\mathfrak{v}_i$ and the jth component of the ith ciphertext in group k by $\mathfrak{v}_{ikj}$.

2 Shuffle Argument

Let $\Pi = (\mathsf{genpkc}, \mathsf{enc}, \mathsf{dec})$ be an additively homomorphic cryptosystem with randomizer space R; we assume henceworth that one uses the validity-enhanced ILin cryptosystem. Assume that $\mathfrak{v}_i$ and $\mathfrak{v}'_i$ are valid ciphertexts of Π. In a shuffle argument, the prover aims to convince the verifier in zero-knowledge that given $(\mathsf{pk}, (\mathfrak{v}_i, \mathfrak{v}'_i)_{i=1}^n)$, he knows a permutation $\sigma \in S_n$ and randomizers s_{ij}, $i \in [1\,..\,n]$ and $j \in [1\,..\,2]$, such that $\mathfrak{v}'_i = \mathfrak{v}_{\sigma(i)} \cdot \mathsf{enc}_{\mathsf{pk}}(0; \boldsymbol{s}_i)$ for $i \in [1\,..\,n]$. More precisely, we define the group-specific binary relation $\mathcal{R}_{sh,n}$ exactly as in [28, 33]:

$$\mathcal{R}_{sh,n} := \left(\begin{array}{l} (\mathsf{gk}, (\mathsf{pk}, \mathfrak{v}_i, \mathfrak{v}'_i)_{i=1}^n), (\sigma, \boldsymbol{s})) : \\ \sigma \in S_n \wedge \boldsymbol{s} \in R^{n \times 2} \wedge \left(\forall i : \mathfrak{v}'_i = \mathfrak{v}_{\sigma(i)} \cdot \mathsf{enc}_{\mathsf{pk}}(0; \boldsymbol{s}_i)\right) \end{array} \right) \ .$$

See Prot. 1 for the full description of the new shuffle argument.

We note that in the real mix-net, (γ, ϱ, β) is handled differently (in particular, γ — and possibly ϱ/β — will be known to the decrypting party while (ϱ, β) does not have to be known to anybody) than the real trapdoor (χ, α) that enables one to simulate the argument and thus cannot be known to anybody. Moreover, $(\mathfrak{g}_1, \mathfrak{g}_2)^{\sum P_i(\chi)}$ is in the CRS only to optimize computation.

3 Permutation Matrix Argument

3.1 New 1-Sparsity Argument

In a 1-sparsity argument [33], the prover aims to convince the verifier that he knows how to open a commitment $\mathfrak{A}_1$ to $(\boldsymbol{a}, r)$, such that *at most* one coefficient a_I is non-zero. If, in addition, $a_I = 1$, then we have a unit vector argument [19]. A 1-sparsity argument can be constructed by using square span programs [16], an especially efficient variant of the quadratic span programs of [22]. We prove its security in the GBGM and therefore use a technique similar to that of [27], and this introduces some complications as we will demonstrate below. While we start using ideas behind the unit vector argument of [19], we only obtain a 1-sparsity argument. Then, in Sect. 3, we show how to obtain an efficient permutation matrix argument from it.

Clearly, $\boldsymbol{a} \in \mathbb{Z}_q^n$ is a unit vector iff the following $n+1$ conditions hold [19]:
- $a_i \in \{0, 1\}$ for $i \in [1\,..\,n]$ (i.e., $\boldsymbol{a}$ is Boolean), and
- $\sum_{i=1}^n a_i = 1$.

Let $\{0, 2\}^{n+1}$ denote the set of $(n+1)$-dimensional vectors where every coefficient is from $\{0, 2\}$, let $\circ$ denote the Hadamard (entry-wise) product of two vectors, let $V := \left(\begin{smallmatrix} 2 \cdot I_{n \times n} \\ \mathbf{1}_n^\top \end{smallmatrix}\right) \in \mathbb{Z}_q^{(n+1) \times n}$ and $\boldsymbol{b} := \left(\begin{smallmatrix} \mathbf{0}_n \\ 1 \end{smallmatrix}\right) \in \mathbb{Z}_q^{n+1}$. Clearly, the above $n+1$ conditions hold iff $V\boldsymbol{a} + \boldsymbol{b} \in \{0, 2\}^{n+1}$, i.e.,

$$(V\boldsymbol{a} + \boldsymbol{b} - \mathbf{1}_{n+1}) \circ (V\boldsymbol{a} + \boldsymbol{b} - \mathbf{1}_{n+1}) = \mathbf{1}_{n+1} \ . \tag{1}$$

Let ω_i, $i \in [1\,..\,n+1]$ be $n+1$ different values. Let

$$Z(X) := \prod_{i=1}^{n+1} (X - \omega_i)$$

$\mathsf{gencrs}(1^\kappa, n \in \mathrm{poly}(\kappa))$**:** Call $\mathsf{gk} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}) \leftarrow \mathsf{genbp}(1^\kappa)$. Let $P_i(X)$ for $i \in [0 \,..\, n]$ be polynomials, chosen in Sect. 3. Set $\boldsymbol{\chi} = (\chi, \alpha, \varrho, \beta, \gamma) \leftarrow_r \mathbb{Z}_q^2 \times (\mathbb{Z}_q \setminus \{0\})^2 \times (\mathbb{Z}_q \setminus \{0, -1\})$. Let enc be the ILin cryptosystem with the secret key γ, and let $(\mathsf{pk}_1, \mathsf{pk}_2)$ be its public key. Set

$$\mathsf{crs} \leftarrow \begin{pmatrix} \mathsf{gk}, (\mathfrak{g}_1^{P_i(\chi)})_{i=1}^n, \mathfrak{g}_1^\varrho, \mathfrak{g}_1^{\alpha+P_0(\chi)}, \mathfrak{g}_1^{P_0(\chi)}, (\mathfrak{g}_1^{((P_i(\chi)+P_0(\chi))^2-1)/\varrho})_{i=1}^n, \mathsf{pk}_1 = (\hat{\mathfrak{g}}_1 = \mathfrak{g}_1^{\varrho/\beta}, \mathfrak{h}_1 = \hat{\mathfrak{g}}_1^\gamma), \\ (\mathfrak{g}_2^{P_i(\chi)})_{i=1}^n, \mathfrak{g}_2^\varrho, \mathfrak{g}_2^{-\alpha+P_0(\chi)}, \mathsf{pk}_2 = (\mathfrak{g}_2, \mathfrak{h}_2 = \mathfrak{g}_2^\gamma), \mathfrak{g}_2^\beta, \hat{e}(\mathfrak{g}_1, \mathfrak{g}_2)^{1-\alpha^2}, (\mathfrak{g}_1, \mathfrak{g}_2)^{\sum_{i=1}^n P_i(\chi)} \end{pmatrix} .$$

and $\mathsf{td} \leftarrow (\chi, \varrho)$. Return $(\mathsf{crs}, \mathsf{td})$.

$\mathsf{pro}(\mathsf{crs}; \mathfrak{v} \in (\mathbb{G}_1 \times \mathbb{G}_2)^{3n}; \sigma \in S_n, \boldsymbol{s} \in \mathbb{Z}_q^{n\times 2})$**:**

1. `Commitment function`
   `Input`: permutation σ, CRS elements $((\mathfrak{g}_1^{P_i(\chi)})_{i=1}^n,\ \mathfrak{g}_1^\varrho,\ (\mathfrak{g}_2^{P_i(\chi)})_{i=1}^n,\ \mathfrak{g}_2^\varrho)$.
   (a) For $i = 1$ to $n-1$:
   Set $r_i \leftarrow_r \mathbb{Z}_q$. Set $(\mathfrak{A}_{i1}, \mathfrak{A}_{i2}) \leftarrow (\mathfrak{g}_1, \mathfrak{g}_2)^{P_{\sigma^{-1}(i)}(\chi) + r_i\varrho}$.
   (b) Set $r_n \leftarrow -\sum_{i=1}^{n-1} r_i$.
   (c) Set $(\mathfrak{A}_{n1}, \mathfrak{A}_{n2}) \leftarrow (\mathfrak{g}_1, \mathfrak{g}_2)^{\sum_{i=1}^n P_i(\chi)} / \prod_{i=1}^{n-1} (\mathfrak{A}_{i1}, \mathfrak{A}_{i2})$.
2. `Sparsity, for permutation matrix`
   `Input`: permutation σ, elements $(\mathfrak{A}_{i1})_{i=1}^n$ from commitment,
   CRS elements $((\mathfrak{g}_1^{P_0(\chi)}),\ \mathfrak{g}_1^\varrho,\ (\mathfrak{g}_1^{((P_i(\chi)+P_0(\chi))^2-1)/\varrho})_{i=1}^n)$.
   (a) For $i = 1$ to n:
   Set $\pi_{\mathsf{1sp}:i} \leftarrow (\mathfrak{A}_{i1}\mathfrak{g}_1^{P_0(\chi)})^{2r_i} (\mathfrak{g}_1^\varrho)^{-r_i^2} \mathfrak{g}_1^{((P_{\sigma^{-1}(i)}(\chi)+P_0(\chi))^2-1)/\varrho}$.
3. `Shuffling function`
   `Input`: permutation σ, original ciphertexts $\mathfrak{v}$, randomness $\boldsymbol{s}$ for shuffling, public keys $\mathsf{pk}_1, \mathsf{pk}_2$.
   (a) For $i = 1$ to n: Set $(\mathfrak{v}'_{i1}, \mathfrak{v}'_{i2}) \leftarrow (\mathfrak{v}_{\sigma(i)1}, \mathfrak{v}_{\sigma(i)2}) \cdot (\mathsf{enc}_{\mathsf{pk}_1}(0; \boldsymbol{s}_i), \mathsf{enc}_{\mathsf{pk}_2}(0; \boldsymbol{s}_i))$.
4. `Consistency function`
   `Input`: original ciphertexts $\mathfrak{v}$, randomness $\boldsymbol{r}$ used in commitment, CRS values $((\mathfrak{g}_2^{P_i(\chi)})_{i=1}^n,\ \mathfrak{g}_2^\varrho)$.
   (a) For $k = 1$ to 2: Set $r_{s:k} \leftarrow_r \mathbb{Z}_q$. Set $\pi_{\mathsf{c1}:k} \leftarrow \mathfrak{g}_2^{\sum_{i=1}^n s_{ik} P_i(\chi) + r_{s:k}\varrho}$.
   (b) $(\boldsymbol{\pi}_{\mathsf{c2}:1}, \boldsymbol{\pi}_{\mathsf{c2}:2}) \leftarrow \prod_{i=1}^n (\mathfrak{v}_{i1}, \mathfrak{v}_{i2})^{r_i} \cdot (\mathsf{enc}_{\mathsf{pk}_1}(0; \boldsymbol{r}_s), \mathsf{enc}_{\mathsf{pk}_2}(0; \boldsymbol{r}_s))$.
5. Return $\pi_{sh} \leftarrow (\mathfrak{v}', (\mathfrak{A}_{i1}, \mathfrak{A}_{i2})_{i=1}^{n-1}, (\pi_{\mathsf{1sp}:i})_{i=1}^n, \pi_{\mathsf{c1}:1}, \pi_{\mathsf{c1}:2}, \boldsymbol{\pi}_{\mathsf{c2}:1}, \boldsymbol{\pi}_{\mathsf{c2}:2})$.

$\mathsf{ver}(\mathsf{crs}; \mathfrak{v}; \mathfrak{v}', (\mathfrak{A}_{i1}, \mathfrak{A}_{i2})_{i=1}^{n-1}, (\pi_{\mathsf{1sp}:i})_{i=1}^n, \pi_{\mathsf{c1}:1}, \pi_{\mathsf{c1}:2}, \boldsymbol{\pi}_{\mathsf{c2}:1}, \boldsymbol{\pi}_{\mathsf{c2}:2})$**:**

1. Set $(\mathfrak{A}_{n1}, \mathfrak{A}_{n2}) \leftarrow (\mathfrak{g}_1, \mathfrak{g}_2)^{\sum_{i=1}^n P_i(\chi)} / \prod_{i=1}^{n-1} (\mathfrak{A}_{i1}, \mathfrak{A}_{i2})$.
2. Set $(p_{1i}, p_{2j}, p_{3ij}, p_{4j})_{i\in[1\,..\,n], j\in[1\,..\,3]} \leftarrow_r \mathbb{Z}_q^{4n+6}$.
3. Check that `/* Permutation matrix: */`
   $$\prod_{i=1}^n \hat{e}\left((\mathfrak{A}_{i1}\mathfrak{g}_1^{\alpha+P_0(\chi)})^{p_{1i}}, \mathfrak{A}_{i2}\mathfrak{g}_2^{-\alpha+P_0(\chi)}\right) = \hat{e}\left(\prod_{i=1}^n \pi_{\mathsf{1sp}:i}^{p_{1i}}, \mathfrak{g}_2^\varrho\right) \cdot \hat{e}(\mathfrak{g}_1, \mathfrak{g}_2)^{(1-\alpha^2)\sum_{i=1}^n p_{1i}}.$$
4. Check that `/* Validity: */`
   $$\hat{e}\left(\mathfrak{g}_1^\varrho, \prod_{j=1}^3 \pi_{\mathsf{c2}:2j}^{p_{2j}} \cdot \prod_{i=1}^n \prod_{j=1}^3 (\mathfrak{v}'_{i2j})^{p_{3ij}}\right) = \hat{e}\left(\prod_{j=1}^3 \pi_{\mathsf{c2}:1j}^{p_{2j}} \cdot \prod_{i=1}^n \prod_{j=1}^3 (\mathfrak{v}'_{i1j})^{p_{3ij}}, \mathfrak{g}_2^\beta\right).$$
5. Set $\mathfrak{R} \leftarrow \hat{e}\left(\hat{\mathfrak{g}}_1, \pi_{\mathsf{c1}:2}^{p_{42}} (\pi_{\mathsf{c1}:1}\pi_{\mathsf{c1}:2})^{p_{43}}\right) \cdot \hat{e}\left(\mathfrak{h}_1, \pi_{\mathsf{c1}:1}^{p_{41}} \pi_{\mathsf{c1}:2}^{p_{42}}\right) / \hat{e}\left(\prod_{j=1}^3 \pi_{\mathsf{c2}:1j}^{p_{4j}}, \mathfrak{g}_2^\varrho\right)$.
6. Check that `/* Consistency: */`
   $$\prod_{i=1}^n \hat{e}\left(\prod_{j=1}^3 (\mathfrak{v}'_{i1j})^{p_{4j}}, \mathfrak{g}_2^{P_i(\chi)}\right) / \prod_{i=1}^n \hat{e}\left(\prod_{j=1}^3 \mathfrak{v}_{i1j}^{p_{4j}}, \mathfrak{A}_{i2}\right) = \mathfrak{R}.$$

Protocol 1: The new shuffle argument

be the unique degree $n+1$ monic polynomial, such that $Z(\omega_i) = 0$ for all $i \in [1 \mathinner{..} n+1]$. Let the ith Lagrange basis polynomial

$$\ell_i(X) := \prod_{j \in [1 \mathinner{..} n+1], j \neq i} ((X - \omega_j)/(\omega_i - \omega_j))$$

be the unique degree n polynomial, s.t. $\ell_i(\omega_i) = 1$ and $\ell_i(\omega_j) = 0$ for $j \neq i$.

For $i \in [1 \mathinner{..} n]$, let $P_i(X)$ be the polynomial that interpolates the ith column of the matrix V. That is,

$$P_i(X) = 2\ell_i(X) + \ell_{n+1}(X)$$

for $i \in [1 \mathinner{..} n]$. Let

$$P_0(X) = \ell_{n+1}(X) - 1$$

be the polynomial that interpolates $\boldsymbol{b} - \mathbf{1}_{n+1}$. In the rest of this paper, we will heavily use the following simple result.

Lemma 1. *$\{P_i(X)\}_{i=0}^n$ is linearly independent.*

Proof. Assume that $\sum_{i=0}^n b_i P_i(X) = 0$ for some constants b_i. Thus, $\sum_{i=0}^n b_i P_i(\omega_k) = 0$ for each k. Consider any $k \in [1 \mathinner{..} n]$. Then, $0 = b_0 P_0(\omega_k) + \sum_{i=1}^n b_i P_i(\omega_k) = b_0(\ell_{n+1}(\omega_k) - 1) + \sum_{i=1}^n b_i(2\ell_i(\omega_k) + \ell_{n+1}(\omega_k)) = -b_0 + 2b_k$. Thus, $b_k = b_0/2$ for $k \in [1 \mathinner{..} n]$. Consider now the case $k = n+1$, then $0 = b_0 P_0(\omega_{n+1}) + \sum_{i=1}^n b_i P_i(\omega_{n+1}) = b_0(\ell_{n+1}(\omega_{n+1}) - 1) + \sum_{i=1}^n b_i(2\ell_i(\omega_{n+1}) + \ell_{n+1}(\omega_{n+1})) = \sum_{i=1}^n b_i = n/2 \cdot b_0$. Thus $b_k = 0$ for $k \in [0 \mathinner{..} n]$. □

We arrive at the polynomial $Q(X) = (\sum_{i=1}^n a_i P_i(X) + P_0(X))^2 - 1 = (P_I(X) + P_0(X))^2 - 1$ (here, we used the fact that $\boldsymbol{a} = \boldsymbol{e_I}$ for some $I \in [1 \mathinner{..} n]$), such that $\boldsymbol{a}$ is a unit vector iff $Z(X) \mid Q(X)$. As in [27], to obtain privacy, we now add randomness $A_\varrho X_\varrho$ to $Q(X)$, arriving at the degree $2n$ polynomial

$$Q_{wi}(X, X_\varrho) = (P_I(X) + P_0(X) + A_\varrho X_\varrho)^2 - 1 \quad . \tag{2}$$

Here, X_ϱ is a special independent random variable, and $A_\varrho \leftarrow_r \mathbb{Z}_q$. This means that we will use an instantiation of the polynomial commitment scheme with $P_i(X)$ defined as in the current subsection.

The new 1-sparsity argument is the subargument of the shuffle argument on Prot. 1, where the verifier only executes verification step Eq. (**??**) for one concrete value of i.

Theorem 1. *Consider $i \in [1 \mathinner{..} n]$. The 1-sparsity argument is perfectly complete. The following holds in the GBGM, given that the generic adversary works in polynomial time. If the honest verifier accepts on Step 3 for this i, then there exists $I \in [1 \mathinner{..} n]$, such that*

$$\mathfrak{A}_{i1} = \mathfrak{g}_1^{a(\chi) + A_\varrho \varrho + A_\alpha(\alpha + P_0(\chi))} \quad , \tag{3}$$

where $a(X) = (1 + A_\alpha) P_I(X)$ for some constant A_α.

Proof. Completeness: For an honest prover, $\mathfrak{A}_{i1} = \mathfrak{g}_1^{A(\boldsymbol{\chi})}$, $\mathfrak{A}_{i2} = \mathfrak{g}_2^{B(\boldsymbol{\chi})}$, and $\pi_{\mathsf{1sp}:i} = \mathfrak{g}_1^{C(\boldsymbol{\chi})}$, where $A(\boldsymbol{X}) = B(\boldsymbol{X}) = P_I(X) + A_\varrho X_\varrho$ and $C(\boldsymbol{X}) = 2A_\varrho \cdot (A(\boldsymbol{X}) + P_0(X)) - A_\varrho^2 X_\varrho + Q_{wi}(X, X_\varrho)/X_\varrho$. Write

$$\mathcal{V}_{1sp}(\boldsymbol{X}) := (A(\boldsymbol{X}) + X_\alpha + P_0(X)) \cdot (B(\boldsymbol{X}) - X_\alpha + P_0(X)) - C(\boldsymbol{X}) \cdot X_\varrho - (1 - X_\alpha^2) \quad . \tag{4}$$

The verification equation Eq. (**??**) assesses that $\mathcal{V}_{1sp}(\boldsymbol{\chi}) = 0$. This simplifies to $\mathcal{V}_{1sp}(\boldsymbol{X}) = (A_\varrho X_\varrho + P_I(X) + P_0(X))^2 - 1 - Q_{wi}(X, X_\varrho)$. Hence for an honest prover, it follows from Eq. (2) that $\mathcal{V}_{1sp}(\boldsymbol{\chi}) = 0$.

3.2 Permutation Matrix Argument

Assume we explicitly compute $\mathfrak{A}_{n1} = \mathfrak{g}_1^{\sum_{i=1}^n P_i(\chi)} / \prod_{j=1}^{n-1} \mathfrak{A}_{j1}$ as in Prot. 1, and then apply the 1-sparsity argument to each $\mathfrak{A}_{i1}$, $i \in [1 \mathinner{..} n]$. Then, as in [33], we get that $(\mathfrak{A}_{11}, \ldots, \mathfrak{A}_{n1})$ commits to a permutation matrix.

4 Validity Argument

The shuffle argument employs validity arguments for $(\boldsymbol{\pi}_{\mathsf{c}2:1}, \boldsymbol{\pi}_{\mathsf{c}2:2})$ and for each $(\mathfrak{v}'_{i1}, \mathfrak{v}'_{i2})$. We outline this argument for $(\boldsymbol{\pi}_{\mathsf{c}2:1}, \boldsymbol{\pi}_{\mathsf{c}2:2})$, the argument is the same for $(\mathfrak{v}'_{i1}, \mathfrak{v}'_{i2})$. More precisely, in the validity argument for $(\boldsymbol{\pi}_{\mathsf{c}2:1}, \boldsymbol{\pi}_{\mathsf{c}2:2})$, the verifier checks that $\hat{e}(\mathfrak{g}_1^{\varrho}, \pi_{\mathsf{c}2:2j}) = \hat{e}(\pi_{\mathsf{c}2:1j}, \mathfrak{g}_2^{\beta})$ for $j \in [1\,..\,3]$. Thus, for

$$\mathcal{V}_{val:j}(\boldsymbol{X}) = E_{1j}(\boldsymbol{X}) X_\beta - X_\varrho E_{2j}(\boldsymbol{X}) \ ,$$

this argument guarantees that in the GBGM, $\mathcal{V}_{val:j}(\boldsymbol{X}) = 0$ for $j \in [1\,..\,3]$.

5 Consistency Argument

We call the subargument of Prot. 1, where the verifier only executes the last verification (namely, Eq. (**??**)), the *consistency argument*. Intuitively, the consistency argument guarantees that the ciphertexts have been permuted by using the same permutation according to which the elements $\mathfrak{g}_k^{P_i(\chi)}$ were permuted inside the commitments $\mathfrak{A}_{i1}$.

References

1. Ambrona, M., Barthe, G., Schmidt, B.: Automated Unbounded Analysis of Cryptographic Constructions in the Generic Group Model. In: EUROCRYPT 2016. LNCS, vol. 9666, pp. 822–851
2. Aranha, D.F., Barreto, P.S.L.M., Longa, P., Ricardini, J.E.: The Realm of the Pairings. In: SAC 2013. LNCS, vol. 8282, pp. 3–25
3. Barthe, G., Fagerholm, E., Fiore, D., Scedrov, A., Schmidt, B., Tibouchi, M.: Strongly-Optimal Structure Preserving Signatures from Type II Pairings: Synthesis and Lower Bounds. In: PKC 2015. LNCS, vol. 9020, pp. 355–376
4. Bellare, M., Garay, J.A., Rabin, T.: Batch Verification with Applications to Cryptography and Checking. In: LATIN 1998. LNCS, vol. 1380, pp. 170–191
5. Bitansky, N., Canetti, R., Paneth, O., Rosen, A.: On the Existence of Extractable One-Way Functions. In: STOC 2014, pp. 505–514
6. Bitansky, N., Dachman-Soled, D., Garg, S., Jain, A., Kalai, Y.T., Lopez-Alt, A., Wichs, D.: Why "Fiat-Shamir for Proofs" Lacks a Proof. In: TCC 2013. LNCS, vol. 7785, pp. 182–201
7. Blum, M., Feldman, P., Micali, S.: Non-Interactive Zero-Knowledge and Its Applications. In: STOC 1988, pp. 103–112
8. Boneh, D., Boyen, X., Goh, E.J.: Hierarchical Identity Based Encryption with Constant Size Ciphertext. In: EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456
9. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: CRYPTO 2004. LNCS, vol. 3152, pp. 41–55
10. Bos, J.W., Costello, C., Naehrig, M.: Exponentiating in Pairing Groups. In: SAC 2013. LNCS, vol. 8282, pp. 438–455
11. Buchberger, B.: An Algorithm for Finding the Basis Elements of the Residue Class Ring of a Zero Dimensional Polynomial Ideal. PhD thesis, University of Innsbruck (1965)
12. Canetti, R., Goldreich, O., Halevi, S.: The Random Oracle Methodology, Revisited. In: STOC 1998, pp. 209–218
13. Chaabouni, R., Lipmaa, H., Zhang, B.: A Non-Interactive Range Proof with Constant Communication. In: FC 2012. LNCS, vol. 7397, pp. 179–199
14. Ciampi, M., Persiano, G., Siniscalchi, L., Visconti, I.: A Transform for NIZK Almost as Efficient and General as the Fiat-Shamir Transform Without Programmable Random Oracles. In: TCC 2016-A (2). LNCS, vol. 9563, pp. 83–111
15. Damgård, I.: Towards Practical Public Key Systems Secure against Chosen Ciphertext Attacks. In: CRYPTO 1991. LNCS, vol. 576, pp. 445–456
16. Danezis, G., Fournet, C., Groth, J., Kohlweiss, M.: Square Span Programs with Applications to Succinct NIZK Arguments. In: ASIACRYPT 2014 (1). LNCS, vol. 8873, pp. 532–550
17. Dent, A.W.: Adapting the Weaknesses of the Random Oracle Model to the Generic Group Model. In: ASIACRYPT 2002. LNCS, vol. 2501, pp. 100–109

18. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.L.: An Algebraic Framework for Diffie-Hellman Assumptions. In: CRYPTO (2) 2013. LNCS, vol. 8043, pp. 129–147
19. Fauzi, P., Lipmaa, H.: Efficient Culpably Sound NIZK Shuffle Argument without Random Oracles. In: CT-RSA 2016. LNCS, vol. 9610, pp. 200–216
20. Fauzi, P., Lipmaa, H., Zhang, B.: Efficient Modular NIZK Arguments from Shift and Product. In: CANS 2013. LNCS, vol. 8257, pp. 92–121
21. Fischlin, M.: A Note on Security Proofs in the Generic Model. In: ASIACRYPT 2000. LNCS, vol. 1976, pp. 458–469
22. Gennaro, R., Gentry, C., Parno, B., Raykova, M.: Quadratic Span Programs and NIZKs without PCPs. In: EUROCRYPT 2013. LNCS, vol. 7881, pp. 626–645
23. Goldwasser, S., Kalai, Y.T.: On the (In)security of the Fiat-Shamir Paradigm. In: FOCS 2003, pp. 102–113
24. Goldwasser, S., Micali, S., Rackoff, C.: The Knowledge Complexity of Interactive Proof-Systems. In: STOC 1985, pp. 291–304
25. Groth, J.: A Verifiable Secret Shuffle of Homomorphic Encryptions. J. Cryptology **23**(4) (2010) pp. 546–579
26. Groth, J.: Short Pairing-Based Non-interactive Zero-Knowledge Arguments. In: ASIACRYPT 2010. LNCS, vol. 6477, pp. 321–340
27. Groth, J.: On the Size of Pairing-based Non-interactive Arguments. In: EUROCRYPT 2016. LNCS, vol. 9666, pp. 305–326
28. Groth, J., Lu, S.: A Non-interactive Shuffle with Pairing Based Verifiability. In: ASIACRYPT 2007. LNCS, vol. 4833, pp. 51–67
29. Groth, J., Ostrovsky, R., Sahai, A.: New Techniques for Noninteractive Zero-Knowledge. Journal of the ACM **59**(3) (2012)
30. Lipmaa, H.: Progression-Free Sets and Sublinear Pairing-Based Non-Interactive Zero-Knowledge Arguments. In: TCC 2012. LNCS, vol. 7194, pp. 169–189
31. Lipmaa, H.: Prover-Efficient Commit-And-Prove Zero-Knowledge SNARKs. In: AFRICACRYPT 2016. LNCS, vol. 9646, pp. 185–206
32. Lipmaa, H., Zhang, B.: A More Efficient Computationally Sound Non-Interactive Zero-Knowledge Shuffle Argument. In: SCN 2012. LNCS, vol. 7485, pp. 477–502
33. Lipmaa, H., Zhang, B.: A More Efficient Computationally Sound Non-Interactive Zero-Knowledge Shuffle Argument. Journal of Computer Security **21**(5) (2013) pp. 685–719
34. Maurer, U.M.: Abstract Models of Computation in Cryptography. In: Cryptography and Coding 2005, pp. 1–12
35. Naor, M.: On Cryptographic Assumptions and Challenges. In: CRYPTO 2003. LNCS, vol. 2729, pp. 96–109
36. Nielsen, J.B.: Separating Random Oracle Proofs from Complexity Theoretic Proofs: The Non-committing Encryption Case. In: CRYPTO 2002. LNCS, vol. 2442, pp. 111–126
37. Schwartz, J.T.: Fast Probabilistic Algorithms for Verification of Polynomial Identities. Journal of the ACM **27**(4) (1980) pp. 701–717
38. Shoup, V.: Lower Bounds for Discrete Logarithms and Related Problems. In: EUROCRYPT 1997. LNCS, vol. 1233, pp. 256–266
39. Zippel, R.: Probabilistic Algorithms for Sparse Polynomials. In: EUROSM 1979. LNCS, vol. 72, pp. 216–226