Practical Functional Encryption for Bilinear Forms

Carmen Elisabetta Zaira Baltico[1], Dario Catalano[1], and Dario Fiore[2]

[1] Dipartimento di Matematica e Informatica, Università di Catania, Italy.
`carmenez@hotmail.it, catalano@dmi.unict.it`
[2] IMDEA Software Institute, Madrid, Spain.
`dario.fiore@imdea.org`

Abstract. We present a practically efficient functional encryption scheme for the class of functionalities that can be expressed via bilinear forms over the integers. Bilinear forms are a general class of quadratic functions that includes, for instance, multivariate quadratic polynomials. Our realization works over asymmetric bilinear groups and is surprisingly simple, efficient and easy to implement. For instance, in our scheme the public key and each ciphertext consist of $2n+1$ and $4n+2$ group elements respectively, where n is the dimension of the encrypted vectors, while secret keys are only two group elements.
The scheme is proved secure under the standard (adaptive) indistinguishability based security notion of Boneh, Sahai and Waters (TCC 2011). The proof is rather convoluted and relies on the so-called generic bilinear group model. Specifically, our proof comes in two main stages. In a preliminary step, we put forward and prove a new master theorem to argue hardness in the generic bilinear group model of a broad family of interactive decisional problems, which includes the indistinguishability-based security game for our functional encryption scheme. Next, the more technically involved part of the proof consists in showing that our scheme actually fits the requirements of our master theorem.

1 Introduction

Traditional public key encryption allows the owner of a secret key sk to decrypt ciphertexts created with respect to a (matching) public key pk. At the same time, without sk, ciphertexts should not reveal any non trivial information about encrypted messages. This all-or-nothing nature of encryption is becoming insufficient in applications where a more fine-grained access to data is required. Functional Encryption (FE) allows to overcome this user-centric access to data of encryption in a very elegant way. Intuitively, given $\mathsf{Encrypt}(m)$ and a key sk_f corresponding to some function f, owner of sk_f learns $f(m)$ and nothing else. Apart from being an interesting theoretical object, Functional Encryption has many natural applications. Think about cloud storage scenarios where users can rely on powerful external servers to store their data. To preserve their privacy users might want to store their files encrypted. At the same time, the users may wish to let the service providers perform basic data mining operations on this data for commercial purposes, without necessarily disclosing the whole data. Functional Encryption allows to reconcile these seemingly contradicting needs as service providers can get secret keys that allow them to perform the desired computations while preserving, as much as possible, the privacy of users.

In terms of security, the standard notion for functional encryption is *indistinguishability*. Informally, this notion states that an adversary who is allowed to see the secret keys for functionalities $f_1, \dots f_n$ should not be able to tell apart which of the challenge messages m_0 or m_1 has been encrypted, under the restriction that $f_i(m_0) = f_i(m_1)$, for all i. This notion was studied in [11,24] and shown inadequate for certain, complex, functionalities[3]. They also explored an alternative,

[3] Here by complex we intend, for instance, functions that are supposed to have some computational hiding properties. In particular, Boneh *et al.* [11] argue that, in applications where security relies on such properties indistinguishability might become problematic.

simulation based, definition, which however cannot be satisfied, in general, without resorting to the random oracle heuristic.

Background on Functional Encryption. The idea of functional encryption originates from Identity Based Encryption (IBE) [26,10] and the closely related concept of Searchable Encryption [9,1]. In IBE, the encrypted messages can be interpreted as a pair (I, m), where I is a, public, string and m is the actual message (often called the "payload"). More in general, the index I can be interpreted as a set of attributes that can be either public or private. Public index schemes are often referred to as attribute based encryption [25,21], a primitive that is by now very well understood [19]. For private index schemes, the situation is more intricate. A first distinction is between *weak* and *strong attribute hiding* schemes [5]. The former notion refers to schemes where the set of secret keys the adversary is allowed to see in the security games is significantly restricted. The adversary is allowed to ask only keys corresponding to functions that cannot be used to decrypt the challenge message. Examples of these schemes are Anonymous Identity based encryption [10,16], Hidden Vector Encryption [12] and (private index) predicate encryption [22,20].

Things are less well established for the setting of private index, strong attribute hiding schemes, a notion that turns out to be equivalent to full fledged functional encryption [11]. Indeed, all known constructions supporting arbitrary circuits, either work for the case of bounded collusions [18,17] or rely on powerful, but poorly understood, assumptions (e.g., [15]). Moreover, they are all terribly inefficient from a practical point of view.

To improve efficiency, a very natural approach is to try to realize schemes using a different, bottom up, perspective. Rather than focusing on generality, one might focus on devising efficient realizations for specific functionalities of practical interest. In 2015, Abdalla *et al.* [2] addressed this question for the case of linear functionalities. In particular they show a construction which is both very simple and relies on standard, well studied assumptions (such as LWE and DDH). The construction was proved secure in the so-called *selective* setting where the adversary is expected to choose the messages on which she wants to be challenged in advance, even before the public key is set up. Not too surprisingly, this result sparkled significant interest in this bottom-up approach, with several results proposing new schemes [6], models [8,4] and improved security [6,3].

Still, none of these results managed to efficiently support more than linear functionalities[4]. This motivates the following question:

Can we construct a practically efficient functional encryption scheme supporting more than linear functionalities?

1.1 Our Contribution

In this paper we answer the previous question in the affirmative. We propose a functional encryption scheme that allows to compute *bilinear forms over the integers*. Using our scheme one can encrypt pairs of n-dimensional vectors $\boldsymbol{a}, \boldsymbol{b}$, while secret keys are associated to $(n \times n)$ matrices $\mathbf{F}$; decryption then allows to compute $\boldsymbol{a}^\top \mathbf{F} \boldsymbol{b} = \sum_{i,j} f_{i,j} a_i b_j$. Bilinear forms represent a very general class of quadratic functions, which includes for example multivariate quadratic polynomials. These functions have several practical applications. For instance, a quadratic polynomial can express many statistical functions (e.g., (weighted) mean, variance, covariance, root-mean-square), the euclidean

[4] We stress that a functional encryption for linear polynomials can be used to support, say, quadratic polynomials, by simply encrypting all the degree two monomials in advance. This however leads to an inefficient solution where the size of the ciphertexts is quadratic in the number of variables.

distance between two vectors, and the application of a linear or quadratic classifier (e.g., linear or quadratic regression).

Our scheme works over asymmetric bilinear groups and is quite efficient. It is essentially optimal in communication size: the public key and the ciphertexts are *linear* in the length of the encrypted vectors (the public key and each ciphertext have $2n+1$ and $4n+2$ group elements respectively), while a secret key consists of *only two* group elements (in addition to the plain function description). In terms of computation, the cost of encryption is linear in the length of the vectors, while the decryption work is linear in the size of the function.

Our FE scheme is proven secure under the standard adaptive indistinguishability-based security notion of Boneh, Sahai and Waters [11]. We build the proof in the generic bilinear group model. In fact, while our scheme is rather simple and easy to understand with respect to correctness, proving its security is much trickier.

In the next paragraphs we give a brief description of our construction, and the proof strategy.

An Overview of Our FE Scheme. The scheme works over asymmetric bilinear groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$. Let us recall that the functionality provided by our FE scheme is that one encrypts pairs of vectors $\boldsymbol{a}, \boldsymbol{b}$, functions are matrices $\mathbf{F}$, and decryption allows to obtain $\boldsymbol{a}^\top \mathbf{F} \boldsymbol{b}$. The initial idea of the construction is to encrypt the two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ à la ElGamal in the two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively, i.e., the ciphertext includes $\boldsymbol{c} = g_1^{r \cdot \boldsymbol{x} + \boldsymbol{a}}$ and $\boldsymbol{d} = g_2^{s \cdot \boldsymbol{y} + \boldsymbol{b}}$ where r, s are randomly chosen and $(g_1^{\boldsymbol{x}}, g_2^{\boldsymbol{y}})$ is the public key. Towards finding a decryption method, we observe that, given $\boldsymbol{c}, \boldsymbol{d}$ and a function $\mathbf{F}$, one can use the bilinear map to compute $U = e(g_1, g_2)^{(r \cdot \boldsymbol{x} + \boldsymbol{a})^\top \mathbf{F}(s \cdot \boldsymbol{y} + \boldsymbol{b})}$. Starting from this, our technique shows how to extend this simple construction with additional structure that enables the extraction from U of the value $e(g_1, g_2)^{\boldsymbol{a}^\top \mathbf{F} \boldsymbol{b}}$, from which the function's result can be eventually obtained by discrete log computation.[5] This basic scheme is extended as follows. First, we let the secret key for function $\mathbf{F}$ be the element $g_1^{\boldsymbol{x}^\top \mathbf{F} \boldsymbol{y}}$. Now, if in the ciphertext we include the element g_2^{rs}, one can extract

$$e(g_1, g_2)^{s \boldsymbol{a}^\top \mathbf{F} \boldsymbol{y} + r \boldsymbol{x}^\top \mathbf{F} \boldsymbol{b} + \boldsymbol{a}^\top \mathbf{F} \boldsymbol{b}} = U \cdot e(g_1^{\boldsymbol{x}^\top \mathbf{F} \boldsymbol{y}}, g_2^{rs})^{-1}.$$

One can see that above the function's result is still "blinded" by cross terms $s(\boldsymbol{a}^\top \mathbf{F} \boldsymbol{y}) + r(\boldsymbol{x}^\top \mathbf{F} \boldsymbol{b})$. Our second idea, to solve this issue and enable full decryption, is to add to the ciphertext the ElGamal encryptions of the vectors $s \cdot \boldsymbol{a}$ and $r \cdot \boldsymbol{b}$. More in detail, we add to the ciphertext the elements $\hat{\boldsymbol{c}} = g_1^{t \cdot \boldsymbol{x} + s \cdot \boldsymbol{a}}$ and $\hat{\boldsymbol{d}} = g_2^{z \cdot \boldsymbol{y} + r \cdot \boldsymbol{b}}$ for random t, z, and the element g_2^{rs-t-z} (instead of g_2^{rs}). With all this information, one can compute the value U in the same way as above, and then use the public key $(g_1^{\boldsymbol{x}}, g_2^{\boldsymbol{y}})$ and the ciphertext components $\hat{\boldsymbol{c}}, \hat{\boldsymbol{d}}$ to compute

$$U' = e(g_1, g_2)^{(t \cdot \boldsymbol{x} + s \cdot \boldsymbol{a})^\top \mathbf{F} \boldsymbol{y} + \boldsymbol{x}^\top \mathbf{F}(z \cdot \boldsymbol{y} + r \cdot \boldsymbol{b})}.$$

By a simple calculation, the function's result can be finally computed as

$$e(g_1, g_2)^{\boldsymbol{a}^\top \mathbf{F} \boldsymbol{b}} = U \cdot U'^{-1} \cdot e(g_1^{\boldsymbol{x}^\top \mathbf{F} \boldsymbol{y}}, g_2^{rs-z-t})^{-1}.$$

As a final note, in the full scheme secret keys are slightly different, we randomize them in order to achieve collusion resistance.

Our Proof Technique. To argue the security of the FE scheme illustrated above we resort to the generic group model. In this model we show that the scheme is secure according to the adaptive

[5] This means that in our scheme messages and functions coefficients are assumed to be sufficiently small integers.

indistinguishability-based notion of Boneh, Sahai and Waters [11]. Our proof technique builds on the generic group framework recently developed by Barthe et al. [7] for the automated analysis of cryptographic assumptions. In this paper we specialize their definitions to bilinear groups, and we extend some of their tools and results in order to deal with interactive decisional problems.

The proof of our functional encryption scheme is developed in two main steps. We first state and prove a master theorem that shows hardness in the generic bilinear group model for a broad family of interactive decisional problems, notably a family which includes the indistinguishability-based experiment for our functional encryption scheme. Slightly more in detail, our master theorem states that these problems are generically hard under a certain algebraic side condition on the distribution of the group elements received by the adversary. These results and techniques are rather general and can be of independent interest.

Second, following the guidelines of our master theorem, we show that our functional encryption scheme meets the algebraic side condition of the master theorem. This is the core part of the proof. Very intuitively, we look at the structure of the scheme's group elements seen by the adversary – public key, ciphertext, secret keys for a bunch of functions – for which the matching of the side condition means that the only information extractable from them is the functions' outputs. So, if the adversary issues only "legitimate" queries (i.e., queries for functions that produce the same results on the two challenge messages), it will not be able to understand which pair of vectors was encrypted.

2 Preliminaries and Definitions

Notation. We denote with $\lambda \in \mathbb{N}$ a security parameter. A *probabilistic polynomial time* (PPT) algorithm $\mathcal{A}$ is a randomized algorithm for which there exists a polynomial $p(\cdot)$ such that for every input x the running time of $\mathcal{A}(x)$ is bounded by $p(|x|)$. We say that a function $\epsilon : \mathbb{N} \rightarrow \mathbb{R}^+$ is *negligible* if for every positive polynomial $p(\lambda)$ there exists $\lambda_0 \in \mathbb{N}$ such that for all $\lambda > \lambda_0$: $\epsilon(\lambda) < 1/p(\lambda)$. If S is a set, $x \xleftarrow{\$} S$ denotes the process of selecting x uniformly at random in S. If $\mathcal{A}$ is a probabilistic algorithm, $y \xleftarrow{\$} \mathcal{A}(\cdot)$ denotes the process of running $\mathcal{A}$ on some appropriate input and assigning its output to y. For a positive integer n, we denote by $[n]$ the set $\{1, \ldots, n\}$. We denote vectors $\boldsymbol{x} = (x_i)$ and matrices $\mathbf{A} = (a_{i,j})$ in bold. For a set S (resp. vector $\boldsymbol{x}$) $|S|$ (resp. $|\boldsymbol{x}|$) denotes its cardinality (resp. number of entries).

Bilinear Groups. Let $\mathcal{G}(1^\lambda)$ be an algorithm (that we call a *bilinear group generator*) which takes as input the security parameter and outputs the description of a bilinear group setting $\mathsf{bgp} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$, where $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are groups of the same prime order $p > 2^\lambda$, $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ are two generators, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is an efficiently computable, non-degenerate, bilinear map. We define $g_T = e(g_1, g_2)$ as the canonical generator of $\mathbb{G}_T$. In the case $\mathbb{G}_1 = \mathbb{G}_2$, the groups are said *symmetric*, else they are said *asymmetric*. In this paper we work with *asymmetric* bilinear groups in which there is no efficiently computable isomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$ (these are also known as Type-III groups [14]).

We use implicit representation of group elements as introduced in [13]. For $s \in \{1, 2, T\}$ and $x \in \mathbb{Z}_p$, we let $[x]_s = g_s^x \in \mathbb{G}_s$. This notation is extended to matrices (and vectors) as follows. For any $\mathbf{A} = (a_{i,j}) \in \mathbb{Z}_p^{m \times n}$ we define

$$[\mathbf{A}]_s = \begin{pmatrix} g_s^{a_{1,1}} & \dots & g_s^{a_{1,n}} \\ \\ g_s^{a_{m,1}} & \dots & g_s^{a_{m,n}} \end{pmatrix} \in \mathbb{G}_s^{m \times n}$$

Note that from an element $[x]_s \in \mathbb{G}_s$ and a scalar a it is possible to efficiently compute $[ax] \in \mathbb{G}_s$. Also, given group elements $[a]_1 \in \mathbb{G}_1$ and $[b]_2 \in \mathbb{G}_2$, one can efficiently compute $[ab]_T = e([a]_1, [b]_2)$. Furthermore, given a matrix of scalars $\mathbf{F} = (f_{i,j}) \in \mathbb{Z}_p^{n \times n}$ and two n-dimensional vectors of group elements $[\boldsymbol{a}]_1, [\boldsymbol{b}]_2$, one can efficiently compute

$$[\boldsymbol{a}^\top \, \mathbf{F} \, \boldsymbol{b}]_T = \left[\sum_{i,j \in [n]} f_{i,j} \cdot a_i \cdot b_j \right]_T = \sum_{i,j \in [n]} f_{i,j} \cdot e([a_i]_1, [b_j]_2)$$

As above, for an easier and more compact presentation, in our work we slightly abuse notation and treat all groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ as additive groups.

2.1 Functional Encryption

We recall the definitions of Functional Encryption as given by Boneh, Sahai and Waters [11].

Definition 1 (Functionality). *A functionality F defined over $(\mathcal{K}, \mathcal{M})$ is a function $F : \mathcal{K} \times \mathcal{M} \to \mathcal{Y} \cup \{\perp\}$ where $\mathcal{K}$ is a key space, $\mathcal{X}$ is a message space and $\mathcal{Y}$ is an output space which does not contain the special symbol $\perp$.*

Definition 2 (Functional Encryption). *A* functional encryption *scheme $\mathcal{FE}$ for a functionality F is defined by a tuple of algorithms $\mathcal{FE} = (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Encrypt}, \mathsf{Decrypt})$ that work as follows.*

$\mathsf{Setup}(1^\lambda)$ *takes as input a security parameter 1^λ and outputs a master secret key* msk *and a master public key* mpk.

$\mathsf{KeyDer}(\mathsf{msk}, K)$ *takes as input the master secret key and a key $K \in \mathcal{K}$ of the functionality (i.e., a function), and outputs a secret key* sk_K.

$\mathsf{Encrypt}(\mathsf{mpk}, M)$ *takes as input the master public key* mpk *and a message $M \in \mathcal{M}$, and outputs a ciphertext* Ct.

$\mathsf{Decrypt}(\mathsf{sk}_K, \mathsf{Ct})$ *takes as input a secret key* sk_K *and a ciphertext* Ct*, and returns an output $Y \in \mathcal{Y} \cup \{\perp\}$.*

For correctness*, it is required that for all* $(\mathsf{mpk}, \mathsf{msk}) \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$*, all keys $K \in \mathcal{K}$ and all messages $M \in \mathcal{M}$, if* $\mathsf{sk}_K \xleftarrow{\$} \mathsf{KeyDer}(\mathsf{msk}, K)$ *and* $\mathsf{Ct} \xleftarrow{\$} \mathsf{Encrypt}(\mathsf{mpk}, M)$*, then it holds with overwhelming probability that* $\mathsf{Decrypt}(\mathsf{sk}_K, \mathsf{Ct}) = F(K, M)$ *whenever $F(K, M) \neq \perp$.*

Indistinguishability-Based Security. For a functional encryption scheme $\mathcal{FE}$ for a functionality F over $(\mathcal{K}, \mathcal{M})$, security against chosen-plaintext attacks (IND-FE-CPA, for short) is defined via the following experiment, denoted $\mathbf{Exp}^{\mathsf{ind\text{-}fe\text{-}cpa\text{-}}\beta}_{\mathcal{FE},\mathcal{A}}(\lambda)$, which is parametrized by an adversary $\mathcal{A}$, a bit $\beta \in \{0, 1\}$, and a security parameter λ.

Setup: run $(\mathsf{mpk}, \mathsf{msk}) \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$ and give mpk to $\mathcal{A}$.

Query: $\mathcal{A}$ adaptively makes secret key queries. At each query, $\mathcal{A}$ specifies a key K and obtains $\mathsf{sk}_K \xleftarrow{\$} \mathsf{KeyDer}(\mathsf{msk}, K)$ from the challenger.

Challenge: $\mathcal{A}$ chooses a pair of messages $M_0, M_1 \in \mathcal{M}$ such that $F(K, M_0) = F(K, M_1)$ holds for all keys K queried in the previous phase. The challenger computes $\mathsf{Ct}^* \xleftarrow{\$} \mathsf{Encrypt}(\mathsf{mpk}, M_\beta)$ and returns Ct^* to $\mathcal{A}$.

Query: $\mathcal{A}$ makes more secret key queries. At each query $\mathcal{A}$ can adaptively choose a key $K \in \mathcal{K}$, but under the requirement that $F(K, M_0) = F(K, M_1)$.

Guess: $\mathcal{A}$ eventually outputs a bit $\beta' \in \{0, 1\}$, and the experiment outputs the same bit.

We define the advantage of $\mathcal{A}$ as

$$\mathbf{Adv}^{\mathsf{ind\text{-}fe\text{-}cpa}}_{\mathcal{FE},\mathcal{A}}(\lambda) = \left|\Pr[\mathbf{Exp}^{\mathsf{ind\text{-}fe\text{-}cpa\text{-}0}}_{\mathcal{FE},\mathcal{A}}(\lambda) = 1] - \Pr[\mathbf{Exp}^{\mathsf{ind\text{-}fe\text{-}cpa\text{-}1}}_{\mathcal{FE},\mathcal{A}}(\lambda) = 1]\right|$$

Definition 3 (Indistinguishability-Based Security). *A functional encryption scheme $\mathcal{FE}$ is secure against chosen-plaintext attacks if for every PPT algorithm $\mathcal{A}$, $\mathbf{Adv}^{\mathsf{ind\text{-}fe\text{-}cpa}}_{\mathcal{FE},\mathcal{A}}(\lambda)$ is negligible.*

Bilinear Forms Functionality. In this work we consider functional encryption schemes for the *bilinear form functionality* over the integers. This is defined as follows. For a positive integer $n \in \mathbb{N}^+$, we let the message space $\mathcal{M} = \mathbb{Z}_p^n \times \mathbb{Z}_p^n$ – every message M is a pair of vectors $(\boldsymbol{a}, \boldsymbol{b})$ – the key space $\mathcal{K} = \mathbb{Z}_p^{n\times n}$ consists of matrices – every key $K \in \mathcal{K}$ is a matrix $\mathbf{F} = (f_{i,j})$ – and the output space is $\mathcal{Y} = \mathbb{Z}_p$. The functionality $F(K, M)$ is the one that computes the value $\boldsymbol{a}^\top \mathbf{F}\, \boldsymbol{b} \in \mathbb{Z}_p$.

We note that bilinear forms capture an interesting class of quadratic functions. As an interesting application, bilinear forms allow one to compute multivariate quadratic polynomials

$$p(\boldsymbol{m}) = p_0 + \sum_i p_i \cdot m_i + \sum_{i,j} p_{i,j} \cdot m_i \cdot m_j$$

by setting $\boldsymbol{a} = \boldsymbol{b} = (1, \boldsymbol{m}) \in \mathbb{Z}_p^{n+1}$ and by encoding p's coefficients in an upper triangular matrix $\mathbf{F} = (f_{i,j}) \in \mathbb{Z}_p^{(n+1)\times(n+1)}$ where: $f_{1,1} = p_0$, $f_{1,i} = p_{i-1}$ for all $i \in [2, n+1]$, $f_{i,j} = 0$ for all $i > j$, and $f_{i,j} = p_{i-1,j-1}$ for all $i \in [2, n+1]$ and $j \geq i$.

3 Our Functional Encryption for Bilinear Forms

In this section we present our construction of a functional encryption scheme that supports the *bilinear form functionality* over the integers.

$\mathsf{Setup}(1^\lambda, n, B_1, B_2)$ runs the bilinear group generator $\mathsf{bgp} \xleftarrow{\$} \mathcal{G}(1^\lambda)$ to obtain parameters $\mathsf{bgp} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$. Next, the algorithm sets the message space $\mathcal{M} = \{0, \ldots, B_1\}^n \times \{0, \ldots, B_1\}^n \subseteq \mathbb{Z}_p^n \times \mathbb{Z}_p^n$ and samples a scalar $w \xleftarrow{\$} \mathbb{Z}_p$ and two vectors $\boldsymbol{x}, \boldsymbol{y} \xleftarrow{\$} \mathbb{Z}_p^n$ uniformly at random. The key space for the bilinear form functionality is set as the set of matrices $\mathcal{K} = \{0, \ldots, B_2\}^{n\times n} \subseteq \mathbb{Z}_p^{n\times n}$. Essentially, B_1 and B_2 represent integer bounds on the entries of the encrypted vectors and function matrices respectively. The way to set these bounds is discussed at the end of the construction.

It returns the master secret key $\mathsf{msk} := (w, \boldsymbol{x}, \boldsymbol{y})$, and the master public key
$\mathsf{mpk} := (\mathsf{bgp}, [\boldsymbol{x}]_1, [\boldsymbol{y}]_2, [w]_2, B_1, B_2)$.

$\mathsf{KeyDer}(\mathsf{msk}, \mathbf{F})$ takes as input the master secret key msk and a matrix $\mathbf{F} \in \mathcal{K}$ and it returns a secret key $\mathsf{sk}_{\mathbf{F}} := (S_1, S_2, \mathbf{F}) \in \mathbb{G}_1^2 \times \mathcal{K}$ where S_1, S_2 are computed as follows. It samples a random $\gamma \xleftarrow{\$} \mathbb{Z}_p$ and computes

$$(S_1, S_2) := ([\boldsymbol{x}^\top \mathbf{F}\, \boldsymbol{y} + \gamma \cdot w]_1, [\gamma]_1).$$

$\mathsf{Encrypt}(\mathsf{mpk}, (\boldsymbol{a}, \boldsymbol{b}))$ takes as input the master public key and a message consisting of two vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathcal{M}$, and returns a ciphertext $\mathsf{Ct} := (\boldsymbol{c}, \hat{\boldsymbol{c}}, \boldsymbol{d}, \hat{\boldsymbol{d}}, E, \hat{E})$ computed as follows.
Choose $r, s, t, z \in \mathbb{Z}_p$ uniformly at random and compute

$$\begin{aligned} \boldsymbol{c} &:= [r \cdot \boldsymbol{x} + \boldsymbol{a}]_1, & \hat{\boldsymbol{c}} &:= [t \cdot \boldsymbol{x} + s \cdot \boldsymbol{a}]_1 \\ \boldsymbol{d} &:= [s \cdot \boldsymbol{y} + \boldsymbol{b}]_2, & \hat{\boldsymbol{d}} &:= [z \cdot \boldsymbol{y} + r \cdot \boldsymbol{b}]_2 \\ E &:= [rs - z - t]_2 & \hat{E} &:= [w(rs - z - t)]_2 \end{aligned}$$

$\mathsf{Decrypt}(\mathsf{sk}_{\mathbf{F}}, \mathsf{Ct})$ parsing $\mathsf{sk}_{\mathbf{F}} := (S_1, S_2, \mathbf{F})$ and $\mathsf{Ct} := (\boldsymbol{c}, \hat{\boldsymbol{c}}, \boldsymbol{d}, \hat{\boldsymbol{d}}, E, \hat{E})$, it first computes

$$V := \boldsymbol{c}^\top \mathbf{F}\, \boldsymbol{d} - [\boldsymbol{x}]_1{}^\top \mathbf{F}\, \hat{\boldsymbol{d}} - \hat{\boldsymbol{c}}^\top \mathbf{F}\, [\boldsymbol{y}]_2 - e(S_1, E) + e(S_2, \hat{E}) \in \mathbb{G}_T$$

and then extracts the discrete logarithm $v \in \mathbb{Z}_p$ of V in base g_T.
Note that, in order for the decryption algorithm to be efficient we work under the assumption that the encrypted vectors $\boldsymbol{a}, \boldsymbol{b}$ and the function $\mathbf{F}$ all have small entries so that $\boldsymbol{a}^\top \mathbf{F} \boldsymbol{b}$ is small as well. In particular, we require that the quantity $B = n^2 B_1^2 B_2 < p$ is small enough to allow for efficient discrete logarithm computation.

3.1 Correctness

To see the correctness of our scheme, let

$$\begin{aligned} A &= \boldsymbol{c}^\top \mathbf{F}\, \boldsymbol{d} = [r \cdot \boldsymbol{x} + \boldsymbol{a}]_1^\top \mathbf{F}\, [s \cdot \boldsymbol{y} + \boldsymbol{b}]_2 = [(rs) \cdot \boldsymbol{x}^\top \mathbf{F}\, \boldsymbol{y} + r \cdot \boldsymbol{x}^\top \mathbf{F}\, \boldsymbol{b} + s \cdot \boldsymbol{a}^\top \mathbf{F}\, \boldsymbol{y} + \boldsymbol{a}^\top \mathbf{F}\, \boldsymbol{b}]_T \\ B &= [\boldsymbol{x}]_1{}^\top \mathbf{F}\, \hat{\boldsymbol{d}} + \hat{\boldsymbol{c}}^\top \mathbf{F}\, [\boldsymbol{y}]_2 = [\boldsymbol{x}]_1^\top \mathbf{F}\, [z \cdot \boldsymbol{y} + r \cdot \boldsymbol{b}]_2 + [t \cdot \boldsymbol{x} + s \cdot \boldsymbol{a}]_1^\top \mathbf{F}\, [\boldsymbol{y}]_2 \\ &= [z \cdot \boldsymbol{x}^\top \mathbf{F}\, \boldsymbol{y} + r \cdot \boldsymbol{x}^\top \mathbf{F}\, \boldsymbol{b} + t \cdot \boldsymbol{x}^\top \mathbf{F}\, \boldsymbol{y} + s \cdot \boldsymbol{a}^\top \mathbf{F}\, \boldsymbol{y}]_T \end{aligned}$$

and note that

$$\begin{aligned} A - B &= [(rs - t - z) \cdot \boldsymbol{x}^\top \mathbf{F}\, \boldsymbol{y} + \boldsymbol{a}^\top \mathbf{F}\, \boldsymbol{b}]_T = e(S_1 - [w \cdot \gamma]_1, E) + [\boldsymbol{a}^\top \mathbf{F}\, \boldsymbol{b}]_T \\ &= e(S_1, E) - e(S_2, \hat{E}) + [\boldsymbol{a}^\top \mathbf{F}\, \boldsymbol{b}]_T \end{aligned}$$

Since $V = A - B - e(S_1, E) + e(S_2, \hat{E})$ it is easy to see that $V = [\boldsymbol{a}^\top \mathbf{F}\, \boldsymbol{b}]_T$.

4 Proof of Security

In this section we state and prove the security of our functional encryption scheme in the generic group model. As an interesting note, in appendix A we show that, when treated as a standard encryption scheme (i.e., no key derivation queries are allowed), the scheme can be proven semantically secure under the Decisional External Diffie-Hellman assumption (i.e., assuming DDH in both groups $\mathbb{G}_1$ and $\mathbb{G}_2$).

Theorem 1. *The functional encryption scheme described in Section 3 satisfies security against chosen-plaintext attacks (i.e., indistinguishability-based security) in the generic bilinear group model. Precisely, for every adversary $\mathcal{A}$ which makes at most Q key derivation oracle queries and $\tilde{Q}$ generic group oracle queries its advantage is*

$$\mathbf{Adv}^{\mathsf{ind\text{-}fe\text{-}cpa}}_{\mathcal{FE},\mathcal{A}}(\lambda) \leq \frac{5(6n+6+\tilde{Q}+2Q)^2}{p}$$

The proof consists of two main steps. We first state and prove a master theorem that shows hardness in the generic bilinear group model for a broad family of interactive decisional problems, notably a family which includes the indistinguishability-based experiment for our functional encryption scheme. Slightly more in detail, our master theorem states that these problems are generically hard under a certain algebraic side condition on the distribution of the elements received by the adversary. Then, following the guidelines of our master theorem, the second step of the proof consists in showing that our functional encryption scheme meets the algebraic side condition of our master theorem.

4.1 Generic Bilinear Group Model for Interactive Problems

In this section we introduce the generic group model framework that we use to prove the security of our functional encryption scheme. We adopt the framework of Barthe et al. [7] for analyzing assumptions in generic k-linear groups, and specialize their definitions to our case of interest, that are asymmetric (Type-III) bilinear groups. In addition, since the results in [7] for interactive assumptions can only model *computational* problems, we provide extensions that allow us to deal with interactive *decisional* problems.

Generic Bilinear Group Model. Let $\mathsf{bgp} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$ be a bilinear group setting, L_1, L_2, L_T be lists of group elements in $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ respectively, and let $\mathcal{D}$ be a distribution over L_1, L_2, L_T. The generic model for a bilinear group setting bgp and a distribution $\mathcal{D}$ is described compactly in Figure 4.1. In this model, the challenger first initializes the lists L_1, L_2, L_T by sampling the group elements according to $\mathcal{D}$, and the adversary receives handles for the elements in the lists. For $s \in \{1, 2, T\}$, $L_s[h]$ denotes the h-th element in the list L_s. The handle to this element is simply the pair (s, h). An adversary running in the generic bilinear group model can apply group operations and bilinear maps to the elements in the lists. To do this, the adversary has to call the appropriate oracle specifying handles for the input elements. The challenger computes the result of a query, stores it in the corresponding list, and returns to the adversary its (newly created) handle. Handles are not unique (i.e., the same group element may appear more than once in a list under different handles), but the adversary is provided with an equality oracle to check if two handles refer to the same group element. This generic group model follows closely that of Maurer [23] (which slightly differs in presentation, although it is equivalent, to that of Shoup [27]) in that the adversary has access to the state of the challenger via handles, and equality queries have "free" cost in the sense that they are not counted for measuring the adversary's computational complexity.

Below we recall a specific class of distributions on lists of group elements that is used in our work. Intuitively, it considers group elements that are generated by sampling random values $x_1, \ldots, x_n \xleftarrow{\$} \mathbb{Z}_p$ and by computing $[p(x_1, \ldots, x_n)]_s \in \mathbb{G}_s$ for some multivariate polynomial p.

Definition 4 (Polynomially Induced Distributions [7]). *Let $\boldsymbol{P} = (P_1, P_2, P_T)$ be three lists of polynomials in $\mathbb{Z}_p[X_1, \ldots, X_n]$ such that each list contains the constant polynomial* 1*. We define*

State: Lists L_1, L_2, L_T over $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ respectively.

Initialization: Lists L_1, L_2, L_T sampled according to distribution $\mathcal{D}$.

Oracles: The oracles provide black-box access to the group operations, the bilinear map and equalities:

- For all $s \in \{1, 2, T\}$: $\mathsf{add}_s(h_1, h_2)$ appends $L_s[h_1] + L_s[h_2]$ to L_s and returns its handle $(s, |L_s|)$.
- For all $s \in \{1, 2, T\}$: $\mathsf{neg}_s(h)$ appends $-L_s[h]$ to L_s and returns its handle $(s, |L_s|)$.
- $\mathsf{map}_e(h_1, h_2)$ appends $e(L_1[h_1], L_2[h_2])$ to L_T and returns its handle $(T, |L_T|)$.
- For all $s \in \{1, 2, T\}$: $\mathsf{eq}_s(h_1, h_2)$ returns 1 if $L_s[h_1] = L_s[h_2]$ and 0 otherwise.

All queries return $\perp$ when given invalid indices.

Fig. 1. Generic group model $\mathsf{GGM}^{\mathsf{bgp}}_{\mathcal{D}}$ for bilinear group setting $\mathsf{bgp} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$ and distribution $\mathcal{D}$.

the distribution $\mathcal{D}_{\boldsymbol{P}}$ as follows: uniformly sample a vector $\boldsymbol{x} \overset{\$}{\leftarrow} \mathbb{Z}_p^n$ and return three lists $\boldsymbol{L} = (L_1, L_2, L_T)$ where, for every $s \in \{1, 2, T\}$, $L_s = \{[p_1(\boldsymbol{x})]_s, \ldots, [p_{|P_s|}(\boldsymbol{x})]_s\}$ with $p_j(\boldsymbol{X})$ being the j-th polynomial in the list P_s. We compactly denote this process as $\boldsymbol{L} \leftarrow \mathcal{D}_{\boldsymbol{P}}$. A distribution $\mathcal{D}$ is called polynomially induced if $\mathcal{D} = \mathcal{D}_{\boldsymbol{P}}$ for some $\boldsymbol{P}$.

To give an example, the input to an adversary for the computational Diffie-Hellman assumption (in $\mathbb{G}_1$) can be described as a polynomially induced distribution where $P_1 = (1, X_1, X_2)$ contains three polynomials in $\mathbb{Z}_p[X_1, X_2]$.

Definition 5 (Completion). *Given lists of polynomials $\boldsymbol{P} = (P_1, P_2, P_T)$, we define their completion $\mathcal{C}(\boldsymbol{P})$ as*

$$\mathcal{C}(\boldsymbol{P}) := P_T \cup \{p_{1,i}(\boldsymbol{X}) \cdot p_{2,j}(\boldsymbol{X}) : \forall p_{1,i} \in P_1, p_{2,j} \in P_2\}$$

Intuitively speaking, for lists of polynomials $\boldsymbol{P}$ their completion represents the list of all polynomials that can be computed by the adversary by applying bilinear maps (i.e., multiplications) to the polynomials in $\boldsymbol{P}$. Our definition given above is a specialization (which gets somewhat simplified) of the completion definition for k-linear groups given in [7].

To give an example, if $\boldsymbol{P} = (P_1, P_2, P_T)$ with $P_1 = \{1, X_1, X_2\}$, $P_2 = \{1, X_2\}$, $P_T = \{1\}$, then its completion is the list $\{1, X_1, X_2, X_1X_2, X_2^2\}$.

Symbolic Group Model. The *symbolic group model* for a bilinear group setting bgp and a polynomially induced distribution $\mathcal{D}_{\boldsymbol{P}}$, denoted as $\mathsf{SGM}^{\mathsf{bgp}}_{\mathcal{D}_{\boldsymbol{P}}}$, gives to the adversary the same interface as the corresponding generic group model, except that internally the challenger stores lists of polynomials in $\mathbb{Z}_p[X_1, \ldots, X_n]$ instead of lists of group elements. The oracles add_s, neg_s, map and eq_s compute addition, negation, multiplication, and equality in the polynomial ring. For any event $\mathcal{E}$ in the generic group model, we define a symbolic version of it, $S(\mathcal{E})$, where equalities over group elements are replaced by equalities over polynomials. In the case where $\mathcal{E}$ is an event which does not involve equality tests on group elements (e.g., in decisional problems where the finalization event can be a simple check $\beta \overset{?}{=} 1$ on the adversary's output bit) it holds $S(\mathcal{E}) = \mathcal{E}$.

Generic and Symbolic Group Model for Simple Interactive Problems. The definitions given so far work for adversaries that receive statically defined lists at the beginning of the game, and then can interact through the oracles to compute group operations and bilinear maps over them. In what follows we generalize the generic and symbolic group models in order to capture a family of interactive decisional problems which includes the indistinguishability security experiment of our functional encryption scheme. The difference in modeling interactive problems in the generic (and symbolic) group model is that the adversary is provided with access to additional oracles that compute further operations on the elements stored in the lists maintained by the challenger in its state. To formalize this setting, we build on the notion of oracles given by Barthe et al. [7] to model interactive assumptions. One difference, though, is that in our work we consider oracles that do *not* take as inputs group elements (i.e., handles to elements in the challenger's lists) from the adversary – we call these problems "*simple* interactive problems". In other words, we consider oracles that take as inputs scalar parameters in $\mathbb{Z}_p$ and return handles to group elements that are computed from these scalar parameters, values randomly sampled by $\mathcal{D}_{\boldsymbol{P}}$ and other $\mathbb{Z}_p$ values freshly sampled by the oracle itself. This restriction on the type of oracles simplifies the presentation, and allows us to state a master theorem which deals with interactive *decisional* problems, whereas the master theorem for interactive assumptions given in [7] can only deal with computational problems.

We begin by defining the notion of an oracle in the generic bilinear group model.

Definition 6 (Oracles in the generic bilinear group model). *An oracle is a tuple $\mathcal{O} = (Q', \ell, m, \boldsymbol{p}, \boldsymbol{v})$ where:*

- *Q' is the number of oracle queries that are allowed;*
- *ℓ is the number of variables $\delta_1, \ldots, \delta_\ell$ in $\mathbb{Z}_p$ that are taken as scalar parameters;*
- *m is the number of values $\omega_1, \ldots, \omega_m$ randomly sampled by the oracle in $\mathbb{Z}_p$;*
- *$\boldsymbol{p} = (p_1, \ldots, p_c)$ is a vector of polynomials in $\mathbb{Z}_p[X_1, \ldots, X_n, \Delta_1, \ldots, \Delta_\ell, \Omega_1, \ldots, \Omega_m]$ that describes the c values returned by the oracle;*
- *$\boldsymbol{v} = (v_1, \ldots, v_c)$ is a vector of indices such that every $v_i \in \{1, 2, T\}$ describes in which group the polynomial p_i belongs to.*

Basically, in the generic bilinear group model, the oracle takes as input a vector $\boldsymbol{\delta} \in \mathbb{Z}_p^\ell$ from the adversary and returns handles to group elements $[p_1(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{\omega})]_{v_1}, \ldots, [p_c(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{\omega})]_{v_c}$ computed by sampling $\boldsymbol{\omega} \overset{\$}{\leftarrow} \mathbb{Z}_p^m$. In the symbolic group model the oracle has the same interface, except that: instead of sampling new values ω_i, it creates new formal variables Ω_i; instead of returning handle to group elements, it returns handles to formal polynomials in the polynomial ring augmented with the newly created formal variables, i.e., $p_j(\boldsymbol{X}, \boldsymbol{\delta}, \boldsymbol{\Omega}) \in \mathbb{Z}_p[\boldsymbol{X}, \boldsymbol{\Omega}]$.

As an example, the reader may consider the key derivation oracle corresponding to our functional encryption scheme. It takes as input $\ell = n^2$ values $\delta_1, \ldots, \delta_{n^2}$ which are the coefficients of the bilinear form $\mathbf{F}$; it samples $m = 1$ random value $\omega_1 = \gamma$; returns $c = 2$ elements of $\mathbb{G}_1$ which can be described by polynomials $\boldsymbol{X}^\top \mathbf{F} \boldsymbol{Y} + \Omega_1 W$ and Ω_1 in $\mathbb{Z}_q[\boldsymbol{X}, \boldsymbol{Y}, W, \mathbf{F}, \Omega_1]$.

Now we state and prove a theorem which shows that one can switch from a generic group model experiment to a corresponding symbolic group model experiment. This theorem extends to interactive problems of Theorem 1 in [7].

Theorem 2 (From Generic to Symbol Group Model with Oracles). *Let* bgp *be a bilinear group setting, where p is prime, $\mathcal{D}_{\boldsymbol{P}}$ a polynomially induced distribution, $\mathcal{O} = (Q', \ell, m, \boldsymbol{p}, \boldsymbol{v})$ an oracle such that $c = |\boldsymbol{p}|$, $\mathcal{A}$ an adversary performing at most Q queries, and $\mathcal{E}$ an event without group equality tests. If d is an upper bound on the degree of the polynomials occurring in the internal state of $\mathsf{SGM}^{\mathsf{bgp}}_{\mathcal{D}_{\boldsymbol{P}}}$, and $N = |P_1| + |P_2| + |P_T|$ is the sum of the lists cardinalities, then*

$$\left|\Pr[\mathsf{GGM}^{\mathsf{bgp}}_{\mathcal{D}_{\boldsymbol{P}},\mathcal{O}}(\mathcal{A}) : \mathcal{E}] - \Pr[\mathsf{SGM}^{\mathsf{bgp}}_{\mathcal{D}_{\boldsymbol{P}},\mathcal{O}}(\mathcal{A}) : S(\mathcal{E})]\right| \leq \frac{d \cdot (N + Q + c \cdot Q')^2}{2p}$$

where the probability is taken over the coins of $\mathsf{GGM}^{\mathsf{bgp}}_{\mathcal{D}_{\boldsymbol{P}}}$ and $\mathcal{A}$.

Proof. The proof of this theorem is essentially the same as that of Theorem 1 in [7], which however does not consider oracles. Given the similarity to [7], we only provide an intuition here. The basic idea is that the adversary, who only sees handles and the outcome of equality queries, can notice a difference between the two games only if an equality query would be answered differently. For a single equality check, the probability of seeing a difference (that occurs when two polynomials $f_1 \neq f_2$ are different in SGM, but $f_1(\tilde{\boldsymbol{x}}) = f_2(\tilde{\boldsymbol{x}})$ for a random $\tilde{\boldsymbol{x}}$ in GGM) is bounded using the Schwartz-Zippel lemma, and is $\leq d/p$. The final bound is then obtained by a union bound on the maximum number of equality checks between group elements (resp. polynomials) in the lists. This number is upper bounded by $T^2/2$, where T is the maximal length of the lists, which is $T = N + Q + c \cdot Q'$ for an adversary that makes at most Q queries to the generic group oracles, and has additional access to $\mathcal{O}$ which can be queried at most Q' times, each time returning c polynomials. □

Looking ahead to defining our master theorem for simple interactive decisional problems, we introduce a notion of parametric completion which works in this interactive setting where the adversary gets access to more polynomials in addition to those statically defined in the initial lists $\boldsymbol{P}$.

Definition 7 (Parametric Completion). *Given lists of polynomials $\boldsymbol{P} = (P_1, P_2, P_T)$ and an oracle $\mathcal{O} = (Q', \ell, m, \boldsymbol{p}, \boldsymbol{v})$, we define their parametric completion $\mathcal{C}^{\mathcal{O}}(\boldsymbol{P})$ as follows. Assuming that the c polynomials in $\boldsymbol{p}$ are in $\mathbb{Z}_p[\boldsymbol{X}, \boldsymbol{\Delta}, \boldsymbol{\Omega}]$, we define an extended set of formal variables*

$$\hat{\boldsymbol{\Delta}} = \{\Delta_{i,j}\}_{i \in [\ell], j \in [Q']}, \quad \hat{\boldsymbol{\Omega}} = \{\Omega_{i,j}\}_{i \in [m], j \in [Q']}$$

The parametric completion $\mathcal{C}^{\mathcal{O}}(\boldsymbol{P})$ consists of polynomials in $\mathbb{Z}_p[\boldsymbol{X}, \hat{\boldsymbol{\Delta}}, \hat{\boldsymbol{\Omega}}]$, and is computed as follows:

```
1. $\boldsymbol{P}' := \boldsymbol{P}$
2. foreach $i \in [Q']$:
3.   foreach $j \in [c]$:
4.     $p'_j := p_j(\Delta_1 := \Delta_{1,i}, \ldots, \Delta_\ell := \Delta_{\ell,i}, \Omega_1 := \Omega_{1,i}, \ldots, \Omega_m := \Omega_{m,i})$
5.     $P'_{v_j} := P'_{v_j} \cup \{p'_j\}$
6. $\mathcal{C}^{\mathcal{O}}(\boldsymbol{P}) := \mathcal{C}(\boldsymbol{P}')$
```

Basically, for every query and every polynomial which is to be returned by the oracle, line 4 redefines the polynomial by making a change of variables (so that the newly introduced variables are unique in the game instead of being only locally unique in the query), while line 5 simply adds this polynomial to the corresponding list (i.e., group) according to the index v_j. Finally, the parametric completion is just a completion (as per Definition 5) computed on the lists of polynomials $\boldsymbol{P}'$ which include the initial lists $\boldsymbol{P}$ plus all the polynomials returned by the oracle. We also note that the notion extends naturally to be parametrized by more than one oracle.

4.2 A Master Theorem for Simple Interactive Decisional Problems

Equipped with the framework and the tools introduced in the previous section, we are now ready to state our master theorem. First, we define what we call simple interactive decisional problems.

Definition 8 (Simple Interactive Decisional Problems). *A simple interactive decisional problem in the generic and symbolic bilinear group model for oracles $\mathcal{O} = (Q', \ell, m, \boldsymbol{p}, \boldsymbol{v})$, $\mathcal{O}_{ch} = (1, \ell^*, m^*, \boldsymbol{f}, \boldsymbol{v}^*)$, and $\mathcal{O}'_{ch} = (1, \ell^*, m^*, \boldsymbol{f}', \boldsymbol{v}^*)$, and a legitimacy predicate H is an experiment where:*

- *The adversary $\mathcal{A}$ gets the same input and the same oracles as in Figure 4.1.*
- *$\mathcal{A}$ can interact with two more oracles, either $\mathcal{O}$ and $\mathcal{O}_{ch}$, or $\mathcal{O}$ and $\mathcal{O}'_{ch}$, such that $\mathcal{O}_{ch}$ (resp. $\mathcal{O}'_{ch}$) can be queried only once.*
- *$\mathcal{A}$ can make (adaptive) queries to its oracles under the restriction that $\mathcal{A}$ is "legitimate", where legitimacy is defined by some predicate H over its oracle queries. Specifically, if $\hat{\boldsymbol{\delta}}^* \in \mathbb{Z}_p^{\ell^*}$ is $\mathcal{A}$'s query to oracle $\mathcal{O}_{ch}$ (or $\mathcal{O}'_{ch}$), and $\hat{\boldsymbol{\delta}} = (\hat{\boldsymbol{\delta}}_1, \dots, \hat{\boldsymbol{\delta}}_{Q'}) \in \mathbb{Z}_p^{\ell \cdot Q'}$ are the Q' queries of $\mathcal{A}$ to oracle $\mathcal{O}$, then H is defined as a predicate $H(\hat{\boldsymbol{\delta}}, \hat{\boldsymbol{\delta}}^*) \in \{0,1\}$.*
- *$\mathcal{A}$ returns a bit β, and the finalization event $\mathcal{E}$ is "$\beta \overset{?}{=} 1$".*

Note that the two oracles $\mathcal{O}_{ch}, \mathcal{O}'_{ch}$ differ only in their output polynomials. Namely, it can be $\boldsymbol{f} \neq \boldsymbol{f}'$ (while their length is clearly the same).

Below we state and prove our master theorem whose goal is to bound the difference between the probabilities of the winning event $\mathcal{E}$ in the two executions of the experiment, provided that a certain algebraic condition on the parametric completions is met.

Theorem 3 (Master Theorem for Simple Interactive Decisional Problems). *Let* bgp *be a bilinear group setting, $\mathcal{D}_{\boldsymbol{P}}$ be a polynomially-induced distribution, and $\mathcal{O} = (Q', \ell, m, \boldsymbol{p}, \boldsymbol{v})$ be an oracle. Furthermore, let $\mathcal{O}_{ch} = (1, \ell^*, m^*, \boldsymbol{f}, \boldsymbol{v}^*)$ and $\mathcal{O}'_{ch} = (1, \ell^*, m^*, \boldsymbol{f}', \boldsymbol{v}^*)$ be two other oracles. Let $N = |P_1| + |P_2| + |P_T|$, $c = |\boldsymbol{p}|$, $c^* = |\boldsymbol{f}| = |\boldsymbol{f}'|$, $r = |\mathcal{C}^{\mathcal{O},\mathcal{O}_{ch}}(\boldsymbol{P})|$, and let d denote an upper bound on the total degrees of the polynomials in the parametric completions. If for all vectors $\hat{\boldsymbol{\delta}} \in \mathbb{Z}_p^{\ell Q'}, \hat{\boldsymbol{\delta}}^* \in \mathbb{Z}_p^{\ell^*}$ such that $H(\hat{\boldsymbol{\delta}}, \hat{\boldsymbol{\delta}}^*) = 1$ it holds*

$$\{\boldsymbol{k} \in \mathbb{Z}_p^r \mid \boldsymbol{k} \cdot C = 0\} = \{\boldsymbol{k} \in \mathbb{Z}_p^r \mid \boldsymbol{k} \cdot C' = 0\}, \tag{1}$$

where $C = \mathcal{C}^{\mathcal{O},\mathcal{O}_{ch}}(\boldsymbol{P})(\hat{\boldsymbol{\Delta}} = \hat{\boldsymbol{\delta}}, \hat{\boldsymbol{\Delta}}^* = \hat{\boldsymbol{\delta}}^*)$ and $C' = \mathcal{C}^{\mathcal{O},\mathcal{O}'_{ch}}(\boldsymbol{P})(\hat{\boldsymbol{\Delta}} = \hat{\boldsymbol{\delta}}, \hat{\boldsymbol{\Delta}}^* = \hat{\boldsymbol{\delta}}^*)$, then

$$\left|\Pr[\mathsf{GGM}^{\mathsf{bgp}}_{\mathcal{D}_{\mathcal{P}},\mathcal{O},\mathcal{O}_{ch}}(\mathcal{A}) : \mathcal{E}] - \Pr[\mathsf{GGM}^{\mathsf{bgp}}_{\mathcal{D}_{\mathcal{P}},\mathcal{O},\mathcal{O}'_{ch}}(\mathcal{A}) : \mathcal{E}]\right| \leq \frac{(N + Q + cQ' + c^*)^2 \cdot d}{p}$$

holds for all legitimate adversaries $\mathcal{A}$ that perform at most Q group operations.

Proof. To prove the theorem we first apply Theorem 2 in order to switch the two experiments from the generic to the symbolic group model.

$$\left|\Pr[\mathsf{GGM}^{\mathsf{bgp}}_{\mathcal{D}_{\boldsymbol{P}},\mathcal{O},\mathcal{O}_{ch}}(\mathcal{A}) : \mathcal{E}] - \Pr[\mathsf{SGM}^{\mathsf{bgp}}_{\mathcal{D}_{\boldsymbol{P}},\mathcal{O},\mathcal{O}_{ch}}(\mathcal{A}) : S(\mathcal{E})]\right| \leq \frac{(N + Q + cQ' + c^*)^2 \cdot d}{2p}$$

$$\left|\Pr[\mathsf{SGM}^{\mathsf{bgp}}_{\mathcal{D}_{\boldsymbol{P}},\mathcal{O},\mathcal{O}'_{ch}}(\mathcal{A}) : S(\mathcal{E})] - \Pr[\mathsf{GGM}^{\mathsf{bgp}}_{\mathcal{D}_{\boldsymbol{P}},\mathcal{O},\mathcal{O}'_{ch}}(\mathcal{A}) : \mathcal{E}]\right| \leq \frac{(N + Q + cQ' + c^*)^2 \cdot d}{2p}$$

To complete the proof we claim that

$$\left|\Pr[\mathsf{SGM}^{\mathsf{bgp}}_{\mathcal{D}_{\mathcal{P}},\mathcal{O},\mathcal{O}_{ch}}(\mathcal{A}) : S(\mathcal{E})] - \Pr[\mathsf{SGM}^{\mathsf{bgp}}_{\mathcal{D}_{\mathcal{P}},\mathcal{O},\mathcal{O}'_{ch}}(\mathcal{A}) : S(\mathcal{E})]\right| = 0$$

Since we are quantifying over legitimate adversaries, we take for granted that $\mathcal{A}$'s queries are such that $H(\hat{\boldsymbol{\delta}}, \hat{\boldsymbol{\delta}}^*) = 1$. $\mathcal{A}$'s view in the symbolic game depends only on the outcome of the equality checks which are performed on the polynomials appearing in the lists stored by the challenger. At this point, the key observation is that the parametric completion $C = \mathcal{C}^{\mathcal{O},\mathcal{O}_{ch}}(\boldsymbol{P})(\hat{\boldsymbol{\Delta}} = \hat{\boldsymbol{\delta}}, \hat{\boldsymbol{\Delta}}^* = \hat{\boldsymbol{\delta}}^*)$ can be viewed as the generating set of a vector space V which describes all the polynomials computable by the adversary starting from the polynomials in $\boldsymbol{P}$ and the polynomials returned by the oracles. So, every polynomial $v \in V$ can be expressed as a linear combination of polynomials in C (i.e., $v = \boldsymbol{\lambda} \cdot C$ for some $\boldsymbol{\lambda}$) and $K = \{\boldsymbol{k} \in \mathbb{Z}_p^r \mid \boldsymbol{k} \cdot C = 0\}$ is the kernel of this linear map. Moreover, since the lists P_1, P_2 are assumed to contain the constant polynomial 1, we note that the parametric completion C in the target group is sufficient to express all polynomials in V. Therefore, the side condition on the equality of the kernels of the two linear maps (i.e. condition 1) means that the adversary sees exactly the same equalities in the two experiments. To see this, consider an execution of the SGM experiment where the adversary has two handles h_1, h_2, and assume that these point to polynomials v_1, v_2 in the left game (i.e., with oracle $\mathcal{O}_{ch}$) and v_1', v_2' in the right game (i.e., with oracle $\mathcal{O}_{ch}'$), such that $v_1 = v_2$ (i.e., $\mathsf{eq}_s(h_1, h_2) = 1$ in the left game) and $v_1' \neq v_2'$ (i.e., $\mathsf{eq}_s(h_1, h_2) = 0$ in the right game). Notice that in both experiments the polynomial v_l (resp. v_l') can be expressed using the same linear combination of elements in the respective completion, i.e., for $l = 1, 2$, in the left game we have $v_l = \boldsymbol{\lambda}_l \cdot C$ whereas in the right game we have $v_l' = \boldsymbol{\lambda}_l \cdot C'$. However, this means that $(\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2) \cdot C = 0$ whereas $(\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2) \cdot C' \neq 0$, which contradicts our side condition. □

4.3 Security of the Functional Encryption Scheme

In this section we use the generic group framework presented in the previous section to prove Theorem 1. We proceed as follows. First, we show that the indistinguishability security game for our functional encryption scheme is a simple interactive decisional problem as per Definition 8, and thus it fits our Theorem 3. Next, we give the core part of the proof which is to show that the scheme is symbolically hard, in the sense that it satisfies the side condition of equation (1) in Theorem 3.

Indistinguishability Security is a Simple Interactive Decisional Problem. Let us consider the indistinguishability security experiment for our functional encryption scheme in the generic bilinear group model. At the beginning the adversary is given handles for the following lists

$$\begin{aligned} L_1 &= \{[1]_1, [x_1]_1, \ldots, [x_n]_1\} \\ L_2 &= \{[1]_2, [w]_2, [y_1]_2, \ldots, [y_n]_2\} \\ L_T &= \{[1]_T\} \end{aligned}$$

which can be seen as output of the polynomially induced distribution $(L_1, L_2, L_T) \leftarrow \mathcal{D}_{\boldsymbol{P}}$, where

$$P_1 = \{1, X_1, \ldots, X_n\},\ P_2 = \{1, W, Y_1, \ldots, Y_n\},\ P_T = \{1\}$$

are lists of polynomials over $\mathbb{Z}_p[\boldsymbol{X}, \boldsymbol{Y}, W]$.

The adversary is also given access to the key derivation oracle that we can write as $\mathcal{O} = (\cdot, n^2, 1, (p_1, p_2), (1, 1))$ since it can be queried an unbounded number of times, it takes as input the description of a quadratic form which is an $(n \times n)$-dimensional matrix $\mathbf{F} = (f_{i,j})$, samples a single value γ, and outputs two elements of $\mathbb{G}_1$ which can be described with polynomials

$$p_1 = \sum_{i,j \in [n]} f_{i,j} \cdot X_i Y_j + \mathit{\Gamma} \cdot W, \quad p_2 = \mathit{\Gamma} \quad \in \mathbb{Z}_p[\boldsymbol{X}, \boldsymbol{Y}, W, \mathbf{F}, \mathit{\Gamma}].$$

Also, $\mathcal{A}$ can query the challenge oracle that is either $\mathcal{O}_{ch}(1, 4n, 4, \boldsymbol{p}^*, \boldsymbol{v}^*)$ or $\mathcal{O}_{ch}'(1, 4n, 4, \boldsymbol{p}^{*\prime}, \boldsymbol{v}^*)$. To see the definition of these oracles, note that they can be queried only once, they take as input two challenge messages $(\boldsymbol{a}, \boldsymbol{b}), (\boldsymbol{a}', \boldsymbol{b}')$, sample four random values r, s, t, z, and output polynomials corresponding to the ciphertexts, that are either:

$$\begin{aligned}\boldsymbol{p}^* = (\ & \{RX_i + A_i\}_{i=1}^n, && \{TX_i + SA_i\}_{i=1}^n, \\ & \{SX_i + B_i\}_{i=1}^n, && \{ZY_i + RB_i\}_{i=1}^n, \\ & RS - Z - T, && W(RS - Z - T)\ )\end{aligned}$$

or

$$\begin{aligned}\boldsymbol{p}^{*\prime} = (\ & \{RX_i + A_i'\}_{i=1}^n, && \{TX_i + SA_i'\}_{i=1}^n, \\ & \{SX_i + B_i'\}_{i=1}^n, && \{ZY_i + RB_i'\}_{i=1}^n, \\ & RS - Z - T, && W(RS - Z - T)\ )\end{aligned}$$

over $\mathbb{Z}_p[\boldsymbol{X}, \boldsymbol{Y}, W, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{A}', \boldsymbol{B}', R, S, T, Z]$.

Moreover, for an adversary $\mathcal{A}$ that makes Q queries $\mathbf{F}_1, \ldots, \mathbf{F}_Q \in \mathbb{Z}_p^{n \times n}$ to the key derivation oracle, one query $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{a}', \boldsymbol{b}' \in \mathbb{Z}_p^n$ to the challenge oracle, and returns a bit β, then by the security definition we have that $\mathcal{A}$ is *legitimate* if "$H(\mathbf{F}_1, \ldots, \mathbf{F}_Q, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{a}', \boldsymbol{b}') = 1$", where the predicate H is true iff $\boldsymbol{a}^\top \mathbf{F}_i \boldsymbol{b} = \boldsymbol{a}'^\top \mathbf{F}_i \boldsymbol{b}'$ for all $i = 1$ to Q.

It is easy to see that the indistinguishability security experiment for our functional encryption scheme is a simple interactive decisional problem as per Definition 8. In order to obtain a proof of Theorem 1, then we invoke our master Theorem 3.

Instantiating the master theorem. Before focusing on the main part of the proof, which is to show the satisfaction of the side condition, we briefly show how the bound of Theorem 1 is obtained. This follows by observing that: the sum of lists cardinalities is $2(n + 1) + 2$, the key derivation and challenge oracles give $2Q$ and $4n + 2$ polynomials respectively, and, as we shall see a bit later, the maximal total degree of polynomials in the parametric completions is $d = 5$.

Satisfaction of the master theorem side condition. The remaining part of the proof focuses on showing that the interactive decisional problem corresponding to the security of our functional encryption scheme satisfies the side condition of equation (1). To this end, our first step is to compute the parametric completions $\mathcal{C}^{\mathcal{O},\mathcal{O}_{ch}}(\boldsymbol{P})$ and $\mathcal{C}^{\mathcal{O},\mathcal{O}'_{ch}}(\boldsymbol{P})$. In the completions computation we consider directly the adversary's queries as scalars instead of formal variables. Namely, we consider the polynomials in the parametric completions evaluated at $\boldsymbol{A} = \boldsymbol{a}, \boldsymbol{B} = \boldsymbol{b}, \boldsymbol{A}' = \boldsymbol{a}', \boldsymbol{B}' = \boldsymbol{b}'$ and $\hat{\boldsymbol{F}}_k = \boldsymbol{f}^{(k)}, \forall k \in [Q]$; this is indeed what we need for analyzing the side condition of equation (1).

Notation. In the rest of the proof, for presentation's convenience we use the following vector notation to express a bilinear form:

$$\langle \boldsymbol{f}, \boldsymbol{a} \otimes \boldsymbol{b} \rangle = \sum_{i,j \in [n]} f_{i,j} a_i b_j$$

Above, $\boldsymbol{f}$ is the n^2-dimensional vector obtained by concatenating all the rows of $\mathbf{F}$, i.e., $\boldsymbol{f} = (f_{1,1}, f_{1,2}, \ldots, f_{1,n}, f_{2,1}, \ldots, f_{n,n-1}, f_{n,n})$ For any n-dimensional vectors $\boldsymbol{a}, \boldsymbol{b}$, we denote by $\boldsymbol{a} \otimes \boldsymbol{b}$ their tensor product that we write as an n^2-dimensional vector $(a_i b_j)_{i,j}$ where the entries i, j are ordered lexicographically, e.g., $\boldsymbol{a} \otimes \boldsymbol{b} = (a_1 b_1, a_1 b_2, \ldots, a_n b_{n-1}, a_n b_n)$.

Parametric Completions. Consider an adversary $\mathcal{A}$ which queries $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{a}', \boldsymbol{b}'$ to the challenge oracle $\mathcal{O}_{ch}$, and $\boldsymbol{f}^{(1)}, \ldots, \boldsymbol{f}^{(Q)}$ to the key derivation oracle. The computation of the parametric completion $\mathcal{C}^{\mathcal{O},\mathcal{O}_{ch}}(\boldsymbol{P})$ (evaluated at $\boldsymbol{A}=\boldsymbol{a},\boldsymbol{B}=\boldsymbol{b},\boldsymbol{A}'=\boldsymbol{a}',\boldsymbol{B}'=\boldsymbol{b}'$ and $\hat{\boldsymbol{F}}_k=\boldsymbol{f}^{(k)},\forall k\in[Q]$) first builds the following lists:

$$\begin{aligned}
&P'_1=\{1\}\cup\{X_i,\ RX_i+a_i,\ TX_i+a_iS\}_{i\in[n]}\cup\{\langle\boldsymbol{f}^{(k)},\boldsymbol{X}\otimes\boldsymbol{Y}\rangle+\varGamma_kW,\ \varGamma_k\}_{k\in[Q]}\\
&P'_2=\{1,\ W,\ RS-Z-T,\ W(RS-Z-T)\}\cup\{Y_i,\ SY_i+b_i,\ ZY_i+b_iR\}_{i\in[n]},\\
&P'_T=\{1\}
\end{aligned}$$

The last step of the parametric completion computation, $\mathcal{C}(\boldsymbol{P}')$, then yields:

$$\begin{aligned}
C=\{&1,\ W,\ RS-Z-T,\ W(RS-Z-T)\}\cup\\
&\{X_i,Y_i,\ WX_i,RSX_i-ZX_i-TX_i,\ RSWX_i-ZWX_i-TWX_i\}_{i\in[n]}\cup\\
&\{X_iY_j\}_{i,j\in[n]}\cup\\
&\{\varGamma_k,\varGamma_kW,RS\varGamma_k-Z\varGamma_k-T\varGamma_k,RSW\varGamma_k-ZW\varGamma_k-TW\varGamma_k\}_{k\in[Q]}\cup\\
&\{\langle\boldsymbol{f}^{(k)},\boldsymbol{X}\otimes\boldsymbol{Y}\rangle+\varGamma_kW,\ \langle\boldsymbol{f}^{(k)},W(\boldsymbol{X}\otimes\boldsymbol{Y})\rangle+\varGamma_kW^2\}_{k\in[Q]}\cup\\
&\{\langle\boldsymbol{f}^{(k)},(RS-Z-T)(\boldsymbol{X}\otimes\boldsymbol{Y})\rangle+(RS-Z-T)W\varGamma_k\}_{k\in[Q]}\cup\\
&\{\langle\boldsymbol{f}^{(k)},(RS-Z-T)W(\boldsymbol{X}\otimes\boldsymbol{Y})\rangle+(RS-Z-T)W^2\varGamma_k\}_{k\in[Q]}\cup\\
&\{\langle\boldsymbol{f}^{(k)},Y_j(\boldsymbol{X}\otimes\boldsymbol{Y})\rangle+Y_j\varGamma_kW,\ Y_j\varGamma_k\}_{j\in[n],k\in[Q]}\cup\\
&\{RX_i+a_i,\ TX_i+a_i\cdot S,\ SY_i+b_i,\ ZY_i+b_i\cdot R\}_{i\in[n]},\cup\\
&\{RWX_i+a_i\cdot W,\ TWX_i+a_i\cdot SW\}_{i\in[n]},\cup\\
&\{R^2SX_i-RZX_i-RTX_i+a_i\cdot(RS-Z-T)\}_{i\in[n]}\cup\\
&\{R^2SWX_i-RWZX_i-RTWX_i+a_i\cdot(RS-Z-T)W\}_{i\in[n]},\cup\\
&\{RSTX_i-TZX_i-T^2X_i+a_i\cdot(RS^2-SZ-ST)\}_{i\in[n]}\cup\\
&\{RSTWX_i-TWZX_i-T^2WX_i+a_i\cdot(RS^2-SZ-ST)W\}_{i\in[n]},\cup\\
&\{SX_iY_j+b_j\cdot X_i,\ ZX_iY_j+b_j\cdot RX_i,\ RX_iY_j+a_i\cdot Y_j,TX_iY_j+a_i\cdot SY_j\}_{i,j\in[n]}\cup\\
&\{RSX_iY_j+a_ib_j+b_j\cdot RX_i+a_i\cdot SY_j\}_{i,j\in[n]}\cup\\
&\{RZX_iY_j+a_ib_j\cdot R+b_j\cdot R^2X_i+a_i\cdot ZY_j\}_{i,j\in[n]}\cup\\
&\{STX_iY_j+a_ib_j\cdot S+b_j\cdot TX_i+a_i\cdot S^2Y_j\}_{i,j\in[n]}\cup\\
&\{TZX_iY_j+a_ib_j\cdot RS+b_j\cdot RTX_i+a_i\cdot SZY_j\}_{i,j\in[n]}\cup\\
&\{\langle\boldsymbol{f}^{(k)},(SY_j+b_j)(\boldsymbol{X}\otimes\boldsymbol{Y})\rangle+SWY_j\varGamma_k+b_j\cdot W\varGamma_k\}_{j\in[n],k\in[Q]}\cup\\
&\{\langle\boldsymbol{f}^{(k)},(ZY_j+b_j\cdot R)(\boldsymbol{X}\otimes\boldsymbol{Y})\rangle+ZWY_j\varGamma_k+b_j\cdot RW\varGamma_k\}_{j\in[n],k\in[Q]}\cup\\
&\{SY_j\varGamma_k+b_j\cdot\varGamma_k,ZY_j\varGamma_k+b_j\cdot R\varGamma_k\}_{j\in[n],k\in[Q]}
\end{aligned}$$

The completion $C'=\mathcal{C}^{\mathcal{O},\mathcal{O}'_{ch}}(\boldsymbol{P})$ is the same as C except for replacing coefficients a_i with a'_i and b_j with b'_j, for all $i,j\in[n]$. In total, both completions consist of $r=|C|=|C'|=4+15n+9n^2+8Q+6nQ$ polynomials in the ring $\mathbb{Z}_p[X_1,\dots,X_n,Y_1,\dots,Y_n,W,\varGamma_1,\dots,\varGamma_Q,R,S,T,Z]$. Also it is possible to see by inspection that the maximal total degree of the polynomials in C and C' is $d=5$ (this is the degree of the monomials R^2SWX_i).

Towards showing equality of the two kernels. Let us recall that the goal of the proof is to show that

$$K=\{\boldsymbol{k}\in\mathbb{Z}_p^r\mid\boldsymbol{k}\cdot C=0\}\ =\ \{\boldsymbol{k}\in\mathbb{Z}_p^r\mid\boldsymbol{k}\cdot C'=0\}=K'\tag{2}$$

under the condition that $\langle \boldsymbol{f}^{(k)}, \boldsymbol{a} \otimes \boldsymbol{b} \rangle = \langle \boldsymbol{f}^{(k)}, \boldsymbol{a}' \otimes \boldsymbol{b}' \rangle$ for all $k = 1$ to Q. One way to show this equality is to compute bases for both kernels K and K', and show that these bases generate the same space (or that they are actually the same). This is what we eventually do. However, instead of proceeding straight to computing bases for the two kernels, we first show that showing the equality in (2) is equivalent to showing a similar equality for a much simpler (smaller) vector space.

Lemma 1. *Let C and C' be the parametric completions computed above. There exist two sets of polynomials $\tilde{C} \subset C$ and $\tilde{C}' \subset C'$, both of cardinality $\tilde{r}$, such that if*

$$\tilde{K} = \{\boldsymbol{k} \in \mathbb{Z}_p^{\tilde{r}} \mid \boldsymbol{k} \cdot \tilde{C} = 0\} \; = \; \{\boldsymbol{k} \in \mathbb{Z}_p^{\tilde{r}} \mid \boldsymbol{k} \cdot \tilde{C}' = 0\} = \tilde{K}' \tag{3}$$

is satisfied then equation (2) is satisfied as well.

Proof. As a first step, we show the existence of a set of indices $\mathcal{S} \subseteq [r]$ and a corresponding vector subspace $U = \{\boldsymbol{k} \in \mathbb{Z}_p^r : k_i = 0, \forall i \in \mathcal{S}\} \subset \mathbb{Z}_p^r$ such that both K and K' are contained in U, i.e., $K \subset U$ and $K' \subset U$. This fact implies that the equality of equation (2) is the same as

$$K = \{\boldsymbol{k} \in U \mid \boldsymbol{k} \cdot C = 0\} \; = \; \{\boldsymbol{k} \in U \mid \boldsymbol{k} \cdot C' = 0\} = K' \tag{4}$$

We show the existence of this set $\mathcal{S}$ by observing the specific shapes of the polynomials in C and C'. $\mathcal{S}$ is the set of indices $i \in [r]$ such that the i-th polynomial in both C and C' contains a unique monomial, i.e., a monomial which appears *only* in that polynomial. For every polynomial p_i (resp. p'_i) such that $i \in \mathcal{S}$ it holds that any vector $\boldsymbol{k} \in K$ (resp. K') must have $k_i = 0$.

By inspection, the set of such unique monomials (which implicitly determines $\mathcal{S}$) is

$$\begin{aligned}
&\{WX_i, RSX_i, ZX_i, RSTX_i, RSTWX_i, RSWX_i, ZWX_i, R^2SX_i, RZX_i\}_{i\in[n]} \cup \\
&\{R^2SWX_i, RZWX_i, RTWX_i, RWX_i, TZX_i, T^2X_i, TZWX_i, T^2WX_i\}_{i\in[n]} \cup \\
&\{STX_iY_j, TZX_iY_j, RZX_iY_j, SX_iY_j\}_{i,j\in[n]} \cup \\
&\{RS\Gamma_k, Z\Gamma_k, T\Gamma_k, W^2\Gamma_k, RSW^2\Gamma_k, ZW^2\Gamma_k, TW^2\Gamma_k\}_{k\in[Q]} \cup \\
&\{Y_j\Gamma_k, SY_j\Gamma_k, WY_j\Gamma_k, ZY_j\Gamma_k, SWY_j\Gamma_k, ZWY_j\Gamma_k\}_{j\in[n],k\in[Q]}
\end{aligned}$$

Then we define $\tilde{C}$ (resp. $\tilde{C}'$) as the subset of C (resp. C') including all those polynomials whose index i is not in $\mathcal{S}$, i.e., $\tilde{C} = \{p_i \in C \mid i \notin \mathcal{S}\}$ and $\tilde{C}' = \{p_i \in C' \mid i \notin \mathcal{S}\}$. Let $\tilde{r} = |\tilde{C}| = |\tilde{C}'|$.

By the definitions of U, $\tilde{C}$ and $\tilde{C}'$ given above, it is easy to see that if the following equality

$$\tilde{K} = \{\boldsymbol{k} \in \mathbb{Z}_p^{\tilde{r}} \mid \boldsymbol{k} \cdot \tilde{C} = 0\} \; = \; \{\boldsymbol{k} \in \mathbb{Z}_p^{\tilde{r}} \mid \boldsymbol{k} \cdot \tilde{C}' = 0\} = \tilde{K}'$$

is satisfied, so is the equality of equation (4), and thus that of equation (2). This completes the proof of the lemma.

For convenience, we show explicitly the simplified completion $\tilde{C}$:

$$\begin{aligned}\tilde{C} = \{&1,\ W,\ RS - Z - T,\ W(RS - Z - T)\} \cup \\ &\{X_i, Y_i\}_{i\in[n]} \cup \\ &\{X_iY_j\}_{i,j\in[n]} \cup \\ &\{\mathit{\Gamma}_k, \mathit{\Gamma}_kW, RSW\mathit{\Gamma}_k - ZW\mathit{\Gamma}_k - TW\mathit{\Gamma}_k\}_{k\in[Q]} \cup \\ &\{\langle \boldsymbol{f}^{(k)}, \boldsymbol{X} \otimes \boldsymbol{Y}\rangle + \mathit{\Gamma}_kW\}_{k\in[Q]} \cup \\ &\{\langle \boldsymbol{f}^{(k)}, (RS - Z - T)(\boldsymbol{X} \otimes \boldsymbol{Y})\rangle + (RS - Z - T)W\mathit{\Gamma}_k\}_{k\in[Q]} \cup \\ &\{RX_i + a_i,\ TX_i + a_i \cdot S,\ SY_i + b_i,\ ZY_i + b_i \cdot R\}_{i\in[n]}, \cup \\ &\{TWX_i + a_i \cdot SW\}_{i\in[n]}, \cup \\ &\{RX_iY_j + a_i \cdot Y_j, ZX_iY_j + b_j \cdot RX_i,\ TX_iY_j + a_i \cdot SY_j\}_{i,j\in[n]} \cup \\ &\{RSX_iY_j + a_ib_j + b_j \cdot RX_i + a_i \cdot SY_j\}_{i,j\in[n]}\end{aligned}$$

$\tilde{C}'$ is the same except for having coefficients a'_i, b'_i instead of a_i, b_i. □

By using the result of Lemma 1, we are left with showing the equality of equation (3). To this end, we apply below an analogous simplification.

Lemma 2. *Let $\tilde{C}$ and $\tilde{C}'$ be the sets of polynomials as defined in Lemma 1. There exist two sets of polynomials $\hat{C} \subset \tilde{C}$ and $\hat{C}' \subset \tilde{C}'$, both of cardinality N, such that if*

$$\hat{K} = \{\boldsymbol{k} \in \mathbb{Z}_p^N \mid \boldsymbol{k} \cdot \hat{C} = 0\} \ = \ \{\boldsymbol{k} \in \mathbb{Z}_p^N \mid \boldsymbol{k} \cdot \hat{C}' = 0\} = \hat{K}' \tag{5}$$

is satisfied then equation (3) *is satisfied as well.*

Proof. The proof of this Lemma is quite similar to that of Lemma 1. As a first step, we show the existence of a set of indices $\tilde{\mathcal{S}} \subseteq [\tilde{r}]$ and a corresponding vector subspace $\tilde{U} = \{\boldsymbol{k} \in \mathbb{Z}_p^{\tilde{r}} : k_i = 0, \forall i \in \tilde{\mathcal{S}}\} \subset \mathbb{Z}_p^{\tilde{r}}$ such that both $\tilde{K}$ and $\tilde{K}'$ are contained in $\tilde{U}$, i.e., $\tilde{K} \subset \tilde{U}$ and $\tilde{K}' \subset \tilde{U}$. This fact implies that the equality of equation (3) is the same as

$$\tilde{K} = \{\boldsymbol{k} \in \tilde{U} \mid \boldsymbol{k} \cdot \tilde{C} = 0\} \ = \ \{\boldsymbol{k} \in \tilde{U} \mid \boldsymbol{k} \cdot \tilde{C}' = 0\} = \tilde{K}' \tag{6}$$

To see the existence of this set $\tilde{\mathcal{S}}$ we again look at the specific shapes of the polynomials in $\tilde{C}$ and $\tilde{C}'$. $\tilde{S}$ is the set of indices $i \in [\tilde{r}]$ such that the i-th polynomial in both $\tilde{C}$ and $\tilde{C}'$ contains a unique monomial, i.e., a monomial which appears only in that polynomial. For every such polynomial p_i (resp. p'_i) it holds that any vector $\boldsymbol{k} \in \tilde{K}$ (resp. $\tilde{K}'$) must have the corresponding i-th coefficient $k_i = 0$.

By inspection, the set of such unique monomials is

$$\{W, RS, Z, T, RSW, WZ, TW\} \cup \{\mathit{\Gamma}_k\}_{k\in[Q]} \cup \{X_i, Y_i, TX_i, TWX_i, ZY_i\}_{i\in[n]} \cup \{RX_iY_j\}_{i,j\in[n]}$$

Similarly to the previous lemma, we define $\hat{C}$ (resp. $\hat{C}'$) as the subset of $\tilde{C}$ (resp. $\tilde{C}'$) including all those polynomials whose index i is not in $\tilde{\mathcal{S}}$, i.e., $\hat{C} = \{p_i \in \tilde{C} \mid i \notin \tilde{\mathcal{S}}\}$, $\quad \hat{C}' = \{p_i \in \tilde{C}' \mid i \notin \tilde{\mathcal{S}}\}$. Let $N = |\hat{C}| = |\hat{C}'|$.

By the definitions of $\tilde{U}$, $\hat{C}$ and $\hat{C}'$, it is easy to see that if the following equality

$$\hat{K} = \{\boldsymbol{k} \in \mathbb{Z}_p^N \mid \boldsymbol{k} \cdot \hat{C} = 0\} \ = \ \{\boldsymbol{k} \in \mathbb{Z}_p^N \mid \boldsymbol{k} \cdot \hat{C}' = 0\} = \hat{K}'$$

is satisfied, so is the equality of equation (6), and thus that of equation (3). This completes the proof of the lemma.

For convenience, we show the simplified completion $\hat{C}$:

$$\begin{aligned}
\hat{C}_0 &= \{1\} \\
\hat{C}_{1,i} &= \{RX_i + a_i\}_{i\in[n]}, \\
\hat{C}_{2,i} &= \{SY_i + b_i\}_{i\in[n]} \\
\hat{C}_{3,i,j} &= \{X_iY_j\}_{i,j\in[n]} \\
\hat{C}_{4,i,j} &= \{RSX_iY_j + a_ib_j + b_j \cdot RX_i + a_i \cdot SY_j\}_{i,j\in[n]} \\
\hat{C}_{5,i,j} &= \{TX_iY_j + a_i \cdot SY_j\}_{i,j\in[n]} \\
\hat{C}_{6,i,j} &= \{ZX_iY_j + b_j \cdot RX_i\}_{i,j\in[n]} \\
\hat{C}_{7,k} &= \{W\mathit{\Gamma}_k\}_{k\in[Q]} \\
\hat{C}_{8,k} &= \{RSW\mathit{\Gamma}_k - ZW\mathit{\Gamma}_k - TW\mathit{\Gamma}_k\}_{k\in[Q]} \\
\hat{C}_{9,k} &= \{\langle \boldsymbol{f}^{(k)}, \boldsymbol{X} \otimes \boldsymbol{Y}\rangle + W\mathit{\Gamma}_k\}_{k\in[Q]} \\
\hat{C}_{10,k} &= \{\langle \boldsymbol{f}^{(k)}, (RS - Z - T)(\boldsymbol{X} \otimes \boldsymbol{Y})\rangle + (RS - Z - T)W\mathit{\Gamma}_k\}_{k\in[Q]}
\end{aligned}$$

$\hat{C}'$ is defined analogously, except for having values a_i' and b_i' instead of a_i and b_i respectively. □

By using the result of Lemma 2, we are left with showing the equality of equation (5) that we recall below

$$\hat{K} = \{\boldsymbol{k} \in \mathbb{Z}_p^N \mid \boldsymbol{k} \cdot \hat{C} = 0\} \;=\; \{\boldsymbol{k} \in \mathbb{Z}_p^N \mid \boldsymbol{k} \cdot \hat{C}' = 0\} = \hat{K}'$$

All the polynomials in the completions $\hat{C}$ and $\hat{C}'$ can be seen as linear combinations of the following set of monomials, that we call the monomials basis

$$\begin{array}{cccc}
H_0 = 1 & \{H_{1,i} = RX_i\}_{i\in[n]} & \{H_{2,i} = SY_i\}_{i\in[n]} & \\
\{H_{3,i,j} = X_iY_j\}_{i,j\in[n]} & \{H_{4,i,j} = RSX_iY_j\}_{i,j\in[n]} & \{H_{5,i,j} = TX_iY_j\}_{i,j\in[n]} & \{H_{6,i,j} = ZX_iY_j\}_{i,j\in[n]} \\
\{H_{7,k} = W\mathit{\Gamma}_k\}_{k\in[Q]} & \{H_{8,k} = RSW\mathit{\Gamma}_k\}_{k\in[Q]} & \{H_{9,k} = TW\mathit{\Gamma}_k\}_{k\in[Q]} & \{H_{10,k} = ZW\mathit{\Gamma}_k\}_{k\in[Q]}
\end{array}$$

Let us write the above monomials basis as a vector $\boldsymbol{H}$ of N entries. Then, $\boldsymbol{H}$ is a monomial basis in the sense that for every polynomial $p \in \hat{C}$ (resp. $p' \in \hat{C}'$) there exists a vector $\boldsymbol{v} \in \mathbb{Z}_p^N$ (resp. $\boldsymbol{v}'$) such that $p = \langle \boldsymbol{v}, \boldsymbol{H}\rangle$ (resp. $p' = \langle \boldsymbol{v}', \boldsymbol{H}\rangle$). (Precisely, $\boldsymbol{v}$ has coefficients in $\{0,1\} \cup \{a_i, b_i\}_{i\in[n]} \cup \{f_{i,j}^{(k)}\}_{i,j\in[n],k\in[Q]}$ while $\boldsymbol{v}'$ has coefficients in $\{0,1\} \cup \{a_i', b_i'\}_{i\in[n]} \cup \{f_{i,j}^{(k)}\}_{i,j\in[n],k\in[Q]}$.)

Let $\mathbf{M} \in \mathbb{Z}_p^{N\times N}$ be the matrix obtained by concatenating, row after row, all these vectors $\boldsymbol{v}_1, \ldots \boldsymbol{v}_N$, i.e., such that all polynomials in the completion can be compactly expressed as $\hat{C} = \mathbf{M} \cdot \boldsymbol{H}$. And let us define analogously $\mathbf{M}'$ such that $\hat{C}' = \mathbf{M}' \cdot \boldsymbol{H}$.

Using this representation in the monomial basis, then showing the equality in (5) is the same as showing

$$\{\boldsymbol{k} \in \mathbb{Z}_p^N : \boldsymbol{k}^\top \cdot \mathbf{M} = \mathbf{0}\} = \{\boldsymbol{k} \in \mathbb{Z}_p^N : \boldsymbol{k}^\top \cdot \mathbf{M}' = \mathbf{0}\}$$

namely that $\mathbf{M}$ and $\mathbf{M}'$ have the same left kernel.

We finalize the proof of Theorem 1 by proving the following lemma.

Lemma 3. *Let* $\mathbf{M}$ *and* $\mathbf{M}'$ *be the matrices defined above. Then* $\ker(\mathbf{M}^\top) = \ker(\mathbf{M}'^\top)$.

Proof. We prove the lemma by computing bases for the kernels of both transposed matrices $\mathbf{M}^\top$ and $\mathbf{M}'^\top$. Below we write the matrix $\mathbf{M}^\top$ using a "block representation" that we explain slightly below:

$$\mathbf{M}^\top = \begin{array}{lr||ccc|cccc|cccc}
 & & \hat{C}_0 & \hat{\boldsymbol{C}}_1 & \hat{\boldsymbol{C}}_2 & \hat{\boldsymbol{C}}_3 & \hat{\boldsymbol{C}}_4 & \hat{\boldsymbol{C}}_5 & \hat{\boldsymbol{C}}_6 & \hat{\boldsymbol{C}}_7 & \hat{\boldsymbol{C}}_8 & \hat{\boldsymbol{C}}_9 & \hat{\boldsymbol{C}}_{10} \\
\hline\hline
H_0 & 1 & 1 & \boldsymbol{a} & \boldsymbol{b} & \mathbf{0} & \boldsymbol{a}\otimes\boldsymbol{b} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_1 & R\boldsymbol{X} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{I}\otimes\boldsymbol{b} & \mathbf{0} & \mathbf{I}\otimes\boldsymbol{b} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_2 & S\boldsymbol{Y} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \boldsymbol{a}\otimes\mathbf{I} & \boldsymbol{a}\otimes\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\hline
\boldsymbol{H}_3 & \boldsymbol{X}\otimes\boldsymbol{Y} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F} & \mathbf{0} \\
\boldsymbol{H}_4 & RS(\boldsymbol{X}\otimes\boldsymbol{Y}) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F} \\
\boldsymbol{H}_5 & T(\boldsymbol{X}\otimes\boldsymbol{Y}) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{F} \\
\boldsymbol{H}_6 & Z(\boldsymbol{X}\otimes\boldsymbol{Y}) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{F} \\
\hline
\boldsymbol{H}_7 & W\boldsymbol{\Gamma} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\
\boldsymbol{H}_8 & RSW\boldsymbol{\Gamma} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{I} \\
\boldsymbol{H}_9 & TW\boldsymbol{\Gamma} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & -\mathbf{I} \\
\boldsymbol{H}_{10} & ZW\boldsymbol{\Gamma} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & -\mathbf{I}
\end{array}$$

Above, the elements on the left and above the double rules || are intended as labels for the rows and columns of the matrix.

In our "block representation" we have that:

- $\hat{C}_0$ is a single column, $\hat{\boldsymbol{C}}_1, \hat{\boldsymbol{C}}_2$ consist of n columns each, $\hat{\boldsymbol{C}}_3, \ldots, \hat{\boldsymbol{C}}_6$ have n^2 columns each, and $\hat{\boldsymbol{C}}_7, \ldots, \hat{\boldsymbol{C}}_{10}$ have Q columns each.
- Similarly to above, H_0 is a single row, $\boldsymbol{H}_1, \boldsymbol{H}_2$ consist of n rows each, $\boldsymbol{H}_3, \ldots, \boldsymbol{H}_6$ have n^2 rows each, and $\boldsymbol{H}_7, \ldots, \boldsymbol{H}_{10}$ have Q rows each.
- $\boldsymbol{a}, \boldsymbol{b}$ are n-dimensional row vectors.
- $\mathbf{I}$ is the identity matrix of dimension $n \times n$, or $n^2 \times n^2$ or $Q \times Q$.
- $\mathbf{0}$ denotes a vector or a matrix of zeros whose dimension is easily extrapolated from its position.
- $\mathbf{F}$ is the $(n^2 \times Q)$-dimensional matrix $\mathbf{F} = \left[\boldsymbol{f}^{(1)} \mid \cdots \mid \boldsymbol{f}^{(Q)}\right]$, which essentially represents a concatenation, column after column, of all the queried functions, each represented as a column vector.
- Tensoring notation: For any vectors $\boldsymbol{a}, \boldsymbol{b}$ of dimension n, we denote by $\boldsymbol{a} \otimes \boldsymbol{b}$ their tensor product that we write as an n^2-dimensional *row vector* $(a_i b_j)_{i,j}$ where the entries i, j are ordered lexicographically, e.g., $\boldsymbol{a} \otimes \boldsymbol{b} := (a_1 b_1, a_1 b_2, \ldots, a_n b_{n-1}, a_n b_n)$. Clearly, $\otimes$ *is not commutative.*

  Moreover, abusing notation, we define the tensor product between an $(\ell \times n)$-dimensional matrix $\mathbf{A}$ and an n-dimensional vector $\boldsymbol{b}$ as the component-wise tensor product of every row of $\mathbf{A}$ with $\boldsymbol{b}$, i.e., letting $\mathbf{A}_i$ be the i-th row of $\mathbf{A}$, we define

$$\mathbf{A} \otimes \boldsymbol{b} := \begin{bmatrix} \mathbf{A}_1 \otimes \boldsymbol{b} \\ \vdots \\ \mathbf{A}_\ell \otimes \boldsymbol{b} \end{bmatrix} \quad \text{and similarly} \quad \boldsymbol{b} \otimes \mathbf{A} := \begin{bmatrix} \boldsymbol{b} \otimes \mathbf{A}_1 \\ \vdots \\ \boldsymbol{b} \otimes \mathbf{A}_\ell \end{bmatrix}$$

As an example, using the just introduced notation, one can take a block-column such as $\hat{\boldsymbol{C}}_1 \in \mathbb{Z}_p^{N \times n}$ in $\mathbf{M}^\top$, and compactly write

$$\hat{\boldsymbol{C}}_1 \otimes \boldsymbol{b} = \begin{bmatrix} \boldsymbol{a} \otimes \boldsymbol{b} \\ \mathbf{I} \otimes \boldsymbol{b} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \in \mathbb{Z}_p^{N \times n^2}$$

This block representation of $\mathbf{M}^\top$ is convenient as it allows us to perform gaussian elimination on $\mathbf{M}^\top$ by expressing multiple column operations with single block-of-columns operations (i.e., intuitively treating every block as if being of constant size). Namely, we will express operations using blocks and observe that these get easily translated into corresponding column operations as follows:

- swap of column-blocks is translated into component-wise swapping of columns,
- addition/subtraction of two column-blocks becomes a component-wise addition/subtraction of the corresponding columns,
- tensoring of a column-block by a vector is translated into (simultaneously) multiplying several columns by field constants.

Now we proceed to computing a basis for the kernel of $\mathbf{M}^\top$. To this end, we first extend below $\mathbf{M}^\top$ with the identity matrix. This gives us the following matrix $\mathbf{T}_1$:

$$\mathbf{T}_1 \quad = \quad \begin{array}{c||ccc|cccc|cccc}
 & \hat{C}_0 & \hat{\boldsymbol{C}}_1 & \hat{\boldsymbol{C}}_2 & \hat{\boldsymbol{C}}_3 & \hat{\boldsymbol{C}}_4 & \hat{\boldsymbol{C}}_5 & \hat{\boldsymbol{C}}_6 & \hat{\boldsymbol{C}}_7 & \hat{\boldsymbol{C}}_8 & \hat{\boldsymbol{C}}_9 & \hat{\boldsymbol{C}}_{10} \\
\hline
H_0 & 1 & \boldsymbol{a} & \boldsymbol{b} & \mathbf{0} & \boldsymbol{a} \otimes \boldsymbol{b} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_1 & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{I} \otimes \boldsymbol{b} & \mathbf{0} & \mathbf{I} \otimes \boldsymbol{b} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_2 & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \boldsymbol{a} \otimes \mathbf{I} & \boldsymbol{a} \otimes \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\hline
\boldsymbol{H}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F} & \mathbf{0} \\
\boldsymbol{H}_4 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F} \\
\boldsymbol{H}_5 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{F} \\
\boldsymbol{H}_6 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{F} \\
\hline
\boldsymbol{H}_7 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\
\boldsymbol{H}_8 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{I} \\
\boldsymbol{H}_9 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & -\mathbf{I} \\
\boldsymbol{H}_{10} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & -\mathbf{I} \\
\hline
 & 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}
\end{array}$$

In what follows we perform gaussian elimination on the above matrix via a series of column transformations until the upper matrix gets in column echelon form. In order, we apply the following column transformations (expressed in block notation):

1. $\hat{\boldsymbol{C}}_1 = \hat{\boldsymbol{C}}_1 - \boldsymbol{a} \otimes \hat{C}_0$ and $\hat{\boldsymbol{C}}_2 = \hat{\boldsymbol{C}}_2 - \boldsymbol{b} \otimes \hat{C}_0$; this yields matrix $\mathbf{T}_2$.
2. $\hat{\boldsymbol{C}}_4 = \hat{\boldsymbol{C}}_4 - (\boldsymbol{a} \otimes \boldsymbol{b}) \otimes \hat{C}_0 - \hat{\boldsymbol{C}}_5 - \hat{\boldsymbol{C}}_6$; this yields matrix $\mathbf{T}_3$.
3. $\hat{\boldsymbol{C}}_5 = \hat{\boldsymbol{C}}_5 - \boldsymbol{a} \otimes \hat{\boldsymbol{C}}_2$ and $\hat{\boldsymbol{C}}_6 = \hat{\boldsymbol{C}}_6 - \hat{\boldsymbol{C}}_1 \otimes \boldsymbol{b}$; this yields matrix $\mathbf{T}_4$.
4. $\hat{\boldsymbol{C}}_9 = \hat{\boldsymbol{C}}_9 - \hat{\boldsymbol{C}}_3 \cdot \mathbf{F} - \hat{\boldsymbol{C}}_7$ and $\hat{\boldsymbol{C}}_{10} = \hat{\boldsymbol{C}}_{10} - \hat{\boldsymbol{C}}_4 \cdot \mathbf{F}$; this yields matrix $\mathbf{T}_5$.
5. $\hat{\boldsymbol{C}}_{10} = \hat{\boldsymbol{C}}_{10} - \hat{\boldsymbol{C}}_8$ and $\hat{\boldsymbol{C}}_4 = \hat{\boldsymbol{C}}_4 + \hat{\boldsymbol{C}}_5 + \hat{\boldsymbol{C}}_6$; this yields matrix $\mathbf{T}_6$.

The matrices $\mathbf{T}_1$–$\mathbf{T}_6$ appear in the following.

$$\mathbf{T}_2 \quad = \quad \begin{array}{r||ccc|cccc|cccc}
 & \hat{C}_0 & \hat{\boldsymbol{C}}_1 & \hat{\boldsymbol{C}}_2 & \hat{\boldsymbol{C}}_3 & \hat{\boldsymbol{C}}_4 & \hat{\boldsymbol{C}}_5 & \hat{\boldsymbol{C}}_6 & \hat{\boldsymbol{C}}_7 & \hat{\boldsymbol{C}}_8 & \hat{\boldsymbol{C}}_9 & \hat{\boldsymbol{C}}_{10} \\
\hline\hline
H_0 & 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{a}\otimes\boldsymbol{b} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_1 & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{I}\otimes\boldsymbol{b} & \mathbf{0} & \mathbf{I}\otimes\boldsymbol{b} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_2 & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \boldsymbol{a}\otimes\mathbf{I} & \boldsymbol{a}\otimes\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\hline
\boldsymbol{H}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F} & \mathbf{0} \\
\boldsymbol{H}_4 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F} \\
\boldsymbol{H}_5 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{F} \\
\boldsymbol{H}_6 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{F} \\
\hline
\boldsymbol{H}_7 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\
\boldsymbol{H}_8 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{I} \\
\boldsymbol{H}_9 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & -\mathbf{I} \\
\boldsymbol{H}_{10} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & -\mathbf{I} \\
\hline
 & 1 & -\boldsymbol{a} & -\boldsymbol{b} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}
\end{array}$$

$$
\mathbf{T}_3 \quad = \quad
\begin{array}{r||ccc|cccc|cccc}
 & \hat{C}_0 & \hat{\boldsymbol{C}}_1 & \hat{\boldsymbol{C}}_2 & \hat{\boldsymbol{C}}_3 & \hat{\boldsymbol{C}}_4 & \hat{\boldsymbol{C}}_5 & \hat{\boldsymbol{C}}_6 & \hat{\boldsymbol{C}}_7 & \hat{\boldsymbol{C}}_8 & \hat{\boldsymbol{C}}_9 & \hat{\boldsymbol{C}}_{10} \\
\hline\hline
H_0 & 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_1 & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}\otimes\boldsymbol{b} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_2 & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \boldsymbol{a}\otimes\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\hline
\boldsymbol{H}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F} & \mathbf{0} \\
\boldsymbol{H}_4 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F} \\
\boldsymbol{H}_5 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{F} \\
\boldsymbol{H}_6 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{F} \\
\hline
\boldsymbol{H}_7 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\
\boldsymbol{H}_8 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{I} \\
\boldsymbol{H}_9 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & -\mathbf{I} \\
\boldsymbol{H}_10 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & -\mathbf{I} \\
\hline
 & 1 & -\boldsymbol{a} & -\boldsymbol{b} & \mathbf{0} & -\boldsymbol{a}\otimes\boldsymbol{b} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}
\end{array}
$$

$$\mathbf{T}_4 \quad = \quad \begin{array}{r||ccc|cccc|cccc}
 & \hat{C}_0 & \hat{\boldsymbol{C}}_1 & \hat{\boldsymbol{C}}_2 & \hat{\boldsymbol{C}}_3 & \hat{\boldsymbol{C}}_4 & \hat{\boldsymbol{C}}_5 & \hat{\boldsymbol{C}}_6 & \hat{\boldsymbol{C}}_7 & \hat{\boldsymbol{C}}_8 & \hat{\boldsymbol{C}}_9 & \hat{\boldsymbol{C}}_{10} \\
\hline\hline
H_0 & 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_1 & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_2 & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\hline
\boldsymbol{H}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F} & \mathbf{0} \\
\boldsymbol{H}_4 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F} \\
\boldsymbol{H}_5 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{F} \\
\boldsymbol{H}_6 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{F} \\
\hline
\boldsymbol{H}_7 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\
\boldsymbol{H}_8 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{I} \\
\boldsymbol{H}_9 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & -\mathbf{I} \\
\boldsymbol{H}_1 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & -\mathbf{I} \\
\hline
 & 1 & -\boldsymbol{a} & -\boldsymbol{b} & \mathbf{0} & -\boldsymbol{a}\otimes\boldsymbol{b} & \boldsymbol{a}\otimes\boldsymbol{b} & \boldsymbol{a}\otimes\boldsymbol{b} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I}\otimes\boldsymbol{b} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & -\boldsymbol{a}\otimes\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}
\end{array}$$

$$\mathbf{T}_5 = \begin{array}{r||ccc|cccc|cccc}
 & \hat{C}_0 & \hat{\boldsymbol{C}}_1 & \hat{\boldsymbol{C}}_2 & \hat{\boldsymbol{C}}_3 & \hat{\boldsymbol{C}}_4 & \hat{\boldsymbol{C}}_5 & \hat{\boldsymbol{C}}_6 & \hat{\boldsymbol{C}}_7 & \hat{\boldsymbol{C}}_8 & \hat{\boldsymbol{C}}_9 & \hat{\boldsymbol{C}}_{10} \\
\hline\hline
H_0 & 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_1 & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_2 & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\hline
\boldsymbol{H}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_4 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_5 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_6 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\hline
\boldsymbol{H}_7 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_8 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{I} \\
\boldsymbol{H}_9 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & -\mathbf{I} \\
\boldsymbol{H}_{10} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & -\mathbf{I} \\
\hline
 & 1 & -\boldsymbol{a} & -\boldsymbol{b} & \mathbf{0} & -\boldsymbol{a}\otimes\boldsymbol{b} & \boldsymbol{a}\otimes\boldsymbol{b} & \boldsymbol{a}\otimes\boldsymbol{b} & \mathbf{0} & \mathbf{0} & \mathbf{0} & (\boldsymbol{a}\otimes\boldsymbol{b})\mathbf{F} \\
 & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I}\otimes\boldsymbol{b} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & -\boldsymbol{a}\otimes\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{F} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{F} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}
\end{array}$$

$$
\mathbf{T}_6 \quad = \quad
\begin{array}{r||ccccccccc||c||c}
 & \hat{C}_0 & \hat{\boldsymbol{C}}_1 & \hat{\boldsymbol{C}}_2 & \hat{\boldsymbol{C}}_3 & \hat{\boldsymbol{C}}_4 & \hat{\boldsymbol{C}}_5 & \hat{\boldsymbol{C}}_6 & \hat{\boldsymbol{C}}_7 & \hat{\boldsymbol{C}}_8 & \hat{\boldsymbol{C}}_9 & \hat{\boldsymbol{C}}_{10} \\
\hline\hline
H_0 & 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_1 & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_2 & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_4 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_5 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_6 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_7 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_8 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_9 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_{10} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{0} \\
\hline
 & 1 & -\boldsymbol{a} & -\boldsymbol{b} & \mathbf{0} & \boldsymbol{a}\otimes\boldsymbol{b} & \boldsymbol{a}\otimes\boldsymbol{b} & \boldsymbol{a}\otimes\boldsymbol{b} & \mathbf{0} & \mathbf{0} & \mathbf{0} & (\boldsymbol{a}\otimes\boldsymbol{b})\mathbf{F} \\
 & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & -\mathbf{I}\otimes\boldsymbol{b} & \mathbf{0} & -\mathbf{I}\otimes\boldsymbol{b} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & -\boldsymbol{a}\otimes\mathbf{I} & -\boldsymbol{a}\otimes\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{F} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{F} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & -\mathbf{I} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\
 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}
\end{array}
$$

As one can see, in the above matrix $\mathbf{T}_6$ the upper part is in column echelon form. Hence, the basis $\mathcal{K}$ of the kernel of $\mathbf{M}^\top$ is represented by the two rightmost block-columns of the lower matrix. These columns are a collection of $2Q$ N-dimensional vectors as follows

$$
\mathcal{K} = \left\{ \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ -\mathbf{F} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ -\mathbf{I} \\ \mathbf{0} \\ \mathbf{I} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} (\boldsymbol{a}\otimes\boldsymbol{b})\mathbf{F} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ -\mathbf{F} \\ \mathbf{F} \\ \mathbf{F} \\ \mathbf{0} \\ -\mathbf{I} \\ \mathbf{0} \\ \mathbf{I} \end{bmatrix} \right\} \in \mathbb{Z}_p^{N\times 2Q}
$$

It is easy to see that when applying the analogous set of transformations on $\mathbf{M'}^\top$ (where $\boldsymbol{a}$ and $\boldsymbol{b}$ are replaced by $\boldsymbol{a}'$ and $\boldsymbol{b}'$ respectively) one obtains the *same* basis $\mathcal{K}$. Precisely, the analogous transformations lead to the same vectors of the kernel except for having $(\boldsymbol{a}'\otimes\boldsymbol{b}')\mathbf{F}$ instead of $(\boldsymbol{a}\otimes\boldsymbol{b})\mathbf{F}$. However, by the legitimacy condition of the security game it holds $(\boldsymbol{a}\otimes\boldsymbol{b})\mathbf{F} = (\boldsymbol{a}'\otimes\boldsymbol{b}')\mathbf{F}$. Hence, $\mathbf{M}$ and $\mathbf{M}'$ have the same basis for their left kernels, which completes the proof. □

References

1. Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 205–222. Springer, August 2005.
2. Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. Simple functional encryption schemes for inner products. In *PKC 2015*, LNCS, pages 733–751. Springer, 2015.
3. Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. Better security for functional encryption for inner product evaluations. Cryptology ePrint Archive, Report 2016/011, 2016. `http://eprint.iacr.org/2016/011`.
4. Michel Abdalla, Mariana Raykova, and Hoeteck Wee. Multi-input inner-product functional encryption from pairings. *IACR Cryptology ePrint Archive*, 2016:425, 2016.
5. Shweta Agrawal, David Mandell Freeman, and Vinod Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 21–40. Springer, December 2011.
6. Shweta Agrawal, Benoît Libert, and Damien Stehlé. Fully secure functional encryption for inner products, from standard assumptions. LNCS, pages 333–362. Springer, August 2016.
7. Gilles Barthe, Edvard Fagerholm, Dario Fiore, John C. Mitchell, Andre Scedrov, and Benedikt Schmidt. Automated analysis of cryptographic assumptions in generic group models. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 95–112. Springer, August 2014.
8. Allison Bishop, Abhishek Jain, and Lucas Kowalczyk. Function-hiding inner product encryption. LNCS, pages 470–491. Springer, December 2015.
9. Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 506–522. Springer, May 2004.
10. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, August 2001.
11. Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, March 2011.
12. Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 535–554. Springer, February 2007.
13. Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, August 2013.
14. Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Appl. Math.*, 156(16):3113–3121, September 2008.
15. Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013.
16. Craig Gentry. Practical identity-based encryption without random oracles. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 445–464. Springer, May / June 2006.
17. Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 555–564. ACM Press, June 2013.
18. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 162–179. Springer, August 2012.
19. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 545–554. ACM Press, June 2013.
20. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In *CRYPTO 2015, Part II*, LNCS, pages 503–523. Springer, August 2015.
21. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 06*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309.

22. Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 146–162. Springer, April 2008.
23. Ueli M. Maurer. Abstract models of computation in cryptography (invited paper). In Nigel P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *LNCS*, pages 1–12. Springer, December 2005.
24. Adam O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010. `http://eprint.iacr.org/2010/556`.
25. Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, May 2005.
26. Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53. Springer, August 1984.
27. Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, May 1997.

A Security of the basic scheme

In this section we give a proof sketch for showing that if one does not have to simulate key derivation queries (i.e., the scheme is treated as a plain asymmetric encryption scheme) then semantic security can be proved under the (symmetric) external decisional Diffie-Hellman assumption (SXDH).

Precisely, we use the following assumption, which can be proved equivalent to SXDH via a standard hybrid argument.

Assumption 1 *Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$ be the output of the bilinear group generator algorithm. For $i = 1, 2$, the following distributions are computationally indistinguishable*

$$\{(g_1, g_2, g_i^a, g_i^{b_0}, \ldots, g_i^{b_n}, g_i^{c_0}, \ldots, g_i^{c_n}) \mid r, s, a, t, b_0, \ldots b_n \leftarrow \mathbb{Z}_p,\ c_j = ab_j \bmod p\}$$

$$\{(g_1, g_2, g_i^a, g_i^{b_0}, \ldots, g_i^{b_n}, g_i^{c_0}, \ldots, g_i^{c_n}) \mid r, s, a, t, b_0, \ldots b_n, c_0, \ldots, c_n \leftarrow \mathbb{Z}_p\}$$

Now we prove the semantic security of the scheme via the following hybrid argument.

Game 0 This represents the security game when played using the original scheme.

Game 1 Here, in the challenge ciphertext, we replace the component $z \cdot \boldsymbol{y}$ in $\hat{\boldsymbol{d}}$ with a random vector $\boldsymbol{\alpha}$. Indistinguishability from the previous game comes from assumption 1 (in $\mathbb{G}_2$). Notice that now $\hat{\boldsymbol{d}}$ and e are independent random variables.

Game 2 Here we continue changing the way the challenge ciphertext is generated, and replace $t \cdot \boldsymbol{x}$ in $\hat{\boldsymbol{c}}$ with a random vector $\boldsymbol{\beta}$. Indistinguishability from the previous game comes from assumption 1 in $\mathbb{G}_1$. Notice that now also $\hat{\boldsymbol{c}}$ is random and independent from the rest of equations.

Game 3 Here we choose to replace the component $r \cdot \boldsymbol{x}$ in $\hat{\boldsymbol{c}}$ with random vector $\boldsymbol{\eta}$. Indistinguishability from previous game comes from assumption 1 (in $\mathbb{G}_1$).

Game 4 Here we use similar ideas to replace $s \cdot \boldsymbol{y}$ with a random vector. Indistinguishability from previous game comes from assumption 1 in $\mathbb{G}_2$. In this game it is clear that the challenge ciphertext does not contain any information about the challenge messages and thus the adversary has zero advantage in winning into this game.