Fully Homomorphic Encryption Using Multivariate Polynomials

Matthew Tamayo-Rios[1], Jean-Charles Faugère[2], Ludovic Perret[2], Peng Hui How[3], and Robin Zhang[4]

[1] Kryptnostic, 201 Marshall St., Redwood City, CA 94063, USA,
matthew@kryptnostic.com
[2] INRIA, Paris-Rocquencourt Center, POLSYS Project, Univ. Paris VI,
jean-charles.faugere@inria.fr
ludovic.perret@lip6.fr
[3] Department of Computer Science, Stanford University,
phow1@stanford.edu
[4] Department of Mathematics, Stanford University,
robinz16@stanford.edu

Abstract. Efficient and secure third party computation has many practical applications in cloud computing. We develop new approach for building fully homomorphic encryption (FHE) schemes, by starting with the intuition of using algebraic descriptions of the encryption and decryption functions to construct a functionally complete set of homomorphic boolean operators. We use this approach to design a compact efficient asymmetric cryptosystem that supports secure third party evaluation of arbitrary boolean functions. In the process, we introduce a new hard problem that is a more difficult variant of of the classical Isomorphism of Polynomials (IP) that we call the Obfuscated-IP.

1 Introduction

Multivariate cryptography is classically defined as the set of cryptographic schemes using the computational hardness of PoSSo, the problem of solving a system of non-linear equations. There is a rather large varieties of basic cryptographic primitives which can be achieved by multivariate cryptosystems : hash-functions [10], stream-cipher [7,8], Zero-Knowledge (ZK) authentication scheme [40,57,56,45], signature (e.g. [49,42,24]), asymmetric encryption (e.g. [43,6,32,49]) and a (somewhat) Fully Homomorphic Encryption (FHE) [2,3].

Homomorphic encryption has long been a subject of great interest in the field of cryptography due to its potential applications in cloud computing for outsourcing analysis and hosting of private data. It was thought infeasible until Gentry's publication of a fully homomorphic encryption algorithm using ideal lattices [34]. FHE [34] is a very powerful cryptographic primitive which allows performing arbitrary computations over encrypted data. In such a scheme, given a function f and a ciphertext $\mathbf{c}$ encrypting a plaintext $\mathbf{m}$, it is possible to transform $\mathbf{c}$ into a new ciphertext $\mathbf{c}'$ which encrypts $f(\mathbf{m})$.

In [2,3], the authors introduced a somewhat FHE scheme based on multivariate polynomials. The idea is to start from a so-called *Polly Cracker cryptosystems* [6,32].The public-key of such systems is a multivariate ideal $\mathcal{I} = \langle f_1, \ldots, f_u \rangle \subset \mathbb{K}[x_1, \ldots, x_n]$, and the secret-key is a Gröbner basis G of $\mathcal{I}$. To encrypt a message $m \in \mathbb{K}[x_1, \ldots, x_n]/\mathcal{I}$, we compute

$$c = \sum h_i f_i + m = f + m, \text{for } f \in \mathcal{I}.$$

The private key is a Gröbner basis G which allows computing efficiently $m = c \bmod \mathcal{I} = c \bmod G$. Despite their simplicity, our confidence in Polly Cracker-style schemes has been shaken as almost all such proposals have been broken [26].
However, due to the multivariate ring structure, it has been noticed [2,3] that a Polly Cracker scheme allows rather naturally homomorphic operations. Indeed:

$$c_1 + c_2 = (f_1 + m_1) + (f_2 + m_2) = (f_1 + f_2) + (m_1 + m_2), \text{for } f_1, f_2 \in \mathcal{I}.$$

So that, $m_1 + m_2 \equiv c_1 + c_2 \bmod \mathcal{I} \equiv c_1 + c_2 \bmod G$. We also have $c_1 \times c_2 \equiv m_1 \times m_2 \bmod G$.

In [2,3], the authors introduced natural noisy variants of classical problems related to Gröbner bases which also generalize previously considered noisy problems such as the Learning With Errors (LWE) problem [55] and the approximate GCD (AGCD) problems of van Dijk et al. [59]. This also leads to present a new somewhat (and doubly) homomorphic encryption scheme – *Noisy Polly-Cracker* – whose security can be provably reduced to these problems.
In this paper, we present a new method for constructing a multivariate FHE.

1.1 State of the Art

FHE is broad and active field of research in cryptography. A good overview of the current state of the art is provided by Armknecht, Boyd, Carr, Gjøsteen, Jaëschke, Reuter, and Strand in [5]. This work also brings much needed canonicalization to various definitions of FHE, which we utilize in this paper. Current approaches to FHE have been mainly based on lattice assumptions, e.g. [34,1,33,59,58] and [18,17,16,20,19,27].
Besides lattice-based, few attempts have been made to design FHE relying on different hardness assumptions. In particular, code-based cryptography [4,12] seem to be an appealing and natural candidate for adapting lattice-based FHE. However, [15] demonstrated that natural code-based analogues of lattice-based FHE can not be secure; including in particular the construction of [4]. Note that [15] don't imply that code-based FHE cannot be constructed, but emphasized that different strategies should be used constructing such schemes. Following the line of impossibility results, the authors of [37] prove several negative results for constructing FHE on several algebraic structures. Let $\mathcal{P}$ be the plaintext space and $\mathcal{C}s$. [37] prove that no secure FHE can be constructed if $\mathcal{P}$ and $\mathcal{C}s$ are vector spaces, and if $\mathcal{P}$ and $\mathcal{C}s$ are fields.
On the more practical side, evaluation of the AES circuit was demonstrated by Gentry et al [35]. Shoup and Halevi's HElib [41] has steadily improved in performance and implemented the BGV [16] scheme with ciphertext packing. SEAL is another librairy [46], released from Microsoft research, which allows to perform somewhat FHE.

1.2 Main Results

Hitherto, most fully homomorphic encryption schemes attempt to preserve the ring structure under various probablistic encryption schemes based on lattices or LWE hardness assumptions. These approaches required the introduction of various techniques such as bootstrapping, squashing, relinearization, and modulus switching for performing multiple evaluations, especially on deeper circuits. Another side-effect was that signifcant machinery was required to transform ring operations to general circuits, which has lead to many classes of homomorphic encryption schemes based on the type of operations they can perform[5].
Rather than attempting to develop more sophisticated techniques for addressing issues with existing approaches, we started with the simpler problem of efficiently constructing homomorphisms in a natural algebraic way. With a set of homomorphisms for a functionally complete set of Boolean operators, we could evaluate arbitrary circuits with as many hops as required as long as ciphertexts stayed compact.
The most trivial approach for construction of homomorphisms is to compose some monomorphism $\mathsf{Enc}_{\mathsf{sk}}$ with some transformation $\mathcal{T}$ composed with the retraction of a monomorphism $\mathsf{Dec}_{\mathsf{sk}}$. This composition will be an algebraic representation of the homomorpism $\mathcal{H}[\mathcal{T}]$. In more cryptographic terms– compose the encryption function with some transformation composed with the decryption function. More formally, given $\mathsf{Enc}_{\mathsf{sk}}(\mathbf{m})$, $\mathsf{Dec}_{\mathsf{sk}}(\mathbf{c})$, and some polynomial tranformation on plaintexts $\mathcal{T}(\mathbf{m}_1, \ldots, \mathbf{m}_n)$, we may construct a homomorphic equivalent of $\mathcal{T}$.

$$\begin{aligned}
\mathsf{Enc}_{\mathsf{sk}}(\mathcal{T}(\mathbf{m}_1, \ldots, \mathbf{m}_n)) &= \mathsf{Enc}_{\mathsf{sk}}(\mathcal{T}(\mathsf{Dec}_{\mathsf{sk}}(\mathbf{c}_1), \ldots, \mathsf{Dec}_{\mathsf{sk}}(\mathbf{c}_n))) \\
\mathcal{H}[\mathcal{T}](\mathbf{c}_1, \ldots, \mathbf{c}_n) &= \mathsf{Enc}_{\mathsf{sk}}(\mathcal{T}(\mathsf{Dec}_{\mathsf{sk}}(\mathbf{c}_1), \ldots, \mathsf{Dec}_{\mathsf{sk}}(\mathbf{c}_n))) \\
\mathsf{Dec}_{\mathsf{sk}}(\mathcal{H}[\mathcal{T}](\mathbf{c}_1, \ldots, \mathbf{c}_n)) &= \mathcal{T}(\mathbf{m}_1, \ldots, \mathbf{m}_n)
\end{aligned} \tag{1}$$

While this provides an intuitive basis for constructing homomorphisms that is correct by construction, it is not clear how to apply this to arbitrary functions. In particular, many possible functions may not have closed

form representations that can be leveraged or even worse– leak information about the decryption function. By leveraging multivariate cryptography we get explicit algebraic representations for the encryption and decryption functions and a set of well studied hard problems related to function composition.
Our contribution is a novel compact ∞-hop fully homomorphic cryptosystem based on a new hardness assumption we are calling Obfuscated Isomorphism of Polymorphism (Obfuscated-IP) related to the well studied Functional Decomposition and Isomorphism of Polynomial problems from multivariate cryptography. By avoiding relinearization, bootstrapping, and squashing we were able to achieve an efficient scheme with practical key sizes.
For convenience, We call the set of functionally complete homomorphic operators that a client can one time provision a server for the purposes of computation the homomorphic public key to distinguish it from the public key.
We avoid the most common pitfalls of algebraic cryptosystems that try decorate easily invertible structures with randomness. We do so by not basing the trapdoor function on the difficulty of calculating the Gröbner basis / determination of ideals. Instead, we introduce a novel problem called `Obfuscated-IP` that is at least as hard as `IP`.

1.3 Organization of the Paper

In section 2 we introduce some basic definitions for the multivariate problems we are relying, some notation for describing our cryptosystem, and composition chains. Next, in section 3 we introduce a new symmetric key multivariate cryptosystem and establish it's security properties. In section 4, we provide a general method for constructing homomorphisms for the cryptosystem described in section 3, provide a few examples for some common circuits, and show how to use homomorphic XOR to create a public key for other parties to encrypt data for owner of the public key.In section 5, we do a basic analysis of its security and estimate levels of security based on cryptosystem parameters. Finally, in section 6 we provide some basic benchmarking of our implementation in C++ and JavaScript.

2 Preliminaries

Historically, the first multivariate public-key encryption scheme – known as C* – has been proposed by Matsumoto and Imai [43]. C* permits to do public-key encryption as well as signature. C* has been completely Patarin [48], but the general principle inspired a a whole generation of researchers that proposed improved variants of the Matsumoto-Imai (MI) principle, e.g. [49,42,51,21,24]. The basic idea of these variants is to construct a public-key $\mathbf{g} \in \mathbb{K}[x_1, \ldots, x_n]^m$ which is equivalent to a set of multivariate polynomials with a specific structure. For instance, derived from some univariate polynomial over an extension, a triangular system, ... [60,61]. Although the scheme presented here differs significantly from known multivariate encryption, its security is deeply related to hard problems that classically arise in multivariate cryptography.

2.1 Functional Decomposition of Polynomials (`FDP`)

`FDP` is a classical problem in computer algebra which asks to decompose – if possible – a set of multivariate polynomials. In cryptography, the problem appeared in the security analysis of 2R/2R$^-$ schemes [52,11]. Let $\mathbf{h} = (h_1, \ldots, h_u) \in \mathbb{K}[x_1, \ldots, x_n]^u$ be a set of multivariate polynomials. We shall say that $(\mathbf{f} = (f_1, \ldots, f_u), \mathbf{g} = (g_1, \ldots, g_n)) \in \mathbb{K}[x_1, \ldots, x_n]^u \times \mathbb{K}[x_1, \ldots, x_n]^n$ is a *decomposition* of $\mathbf{h}$ if:

$$\mathbf{f} \circ \mathbf{g} = (f_1(g_1, \ldots, g_n), \ldots, f_u(g_1, \ldots, g_n)) = \mathbf{h} = (h_1, \ldots, h_u).$$

Given $\mathbf{h} = (h_1, \ldots, h_u) \in \mathbb{K}[x_1, \ldots, x_n]^u$, `FDP` is the problem of recovering a decomposition $(\mathbf{f}, \mathbf{g}) \in \mathbb{K}[x_1, \ldots, x_n]^u \times \mathbb{K}[x_1, \ldots, x_n]^n$. Observe that taking $\mathbf{h} = \mathbf{f}$ and $\mathbf{g} = (g_1, \ldots, g_n) = (x_1, \ldots, x_n)$, or $\mathbf{f} = (x_1, \ldots, x_u)$ and $\mathbf{g} = (h_1, \ldots, h_u, 0, \ldots, 0)$ will lead to a valid, but trivial, decomposition of $\mathbf{h}$. To avoid these cases, we fix the degrees of a decomposition. Thus, let $d_h, d_f, d_g > 1$ be positive integers strictly greater than one. We define:

$\texttt{FDP}(d_h, d_f, d_g)$
Input : $\mathbf{h} = (h_1, \ldots, h_u) \in \mathbb{K}[x_1, \ldots, x_n]^u$, with the h_is all of degree d_h.
Question : Find – if any – polynomials $(\mathbf{f} = (f_1, \ldots, f_u), \mathbf{g} = (g_1, \ldots, g_n)) \in \mathbb{K}[x_1, \ldots, x_n]^u \times \mathbb{K}[x_1, \ldots, x_n]^n$ of degree d_f and d_g respectively such that

$$\mathbf{f} \circ \mathbf{g} = \mathbf{h}.$$

In [62,28], the authors presented algorithms for solving $\texttt{FDP}(4, 2, 2)$, i.e. decomposing quartics polynomials into quadratic polynomials. It appears that the hardness of the problem depends on the ratio u/n. If this ratio is equal to to 1, then [62,28] return a decomposition in polynomial-time. More generally, the algorithm from [28] works in polynomial-time. An algorithm for solving the general $\texttt{FDP}(d_h, d_f, d_g)$ is presented in [31]. The algorithm is also efficient when the ratio u/n is constant. We emphasize that the algorithms [62,28,31] work actually work in the "*tame case*", i.e. when $\text{char}(\mathbb{K}) \nmid d_g$.

In this paper, we consider FDP when $\text{char}(\mathbb{K}) \mid d_g$, that is the "*wild case*". It appears that the wild case has been has much less investigated than the tame case. This presumed hardness of the wild case motivated the design of a variety of schemes [11] based on the hardness of FDP with $\mathbb{K} = \mathbb{F}_2$. In a series of papers [44,36,25], the constructions from [11] have been broken. A key ingredient of [36] is a distinguisher between decomposable polynomials and random polynomials. The idea is to consider the rank of the partial derivatives of an instance $\mathbf{h}(h_1, \ldots, h_u) \in \mathbb{K}[x_1, \ldots, x_n]^u$ of FDP. The distinguisher relies on a heuristic assumption about the rank of partial derivatives of random set of polynomials $\mathbf{h} = (h_1, \ldots, h_u) \in \mathbb{K}[x_1, \ldots, x_n]^u$. In [36], the heuristic has been experimentally verified for $\mathbb{K} = \mathbb{F}_2$.

Note that FDP remains an hard problem since if has been proven NP-Hard by Dickerson [22,23]. Hard instances of FDP appeared when the ratio u/n is not a constant.

2.2 Isomorphism of Polynomials (IP)

The Isomorphism of Polynomials (IP) problem, introduced by Patarin [50], is in some sense a sub-case of FDP in which we try decomposing into linear polynomials. The IP problem is defined is as follows:
Isomorphism of Polynomials (IP)
Input: $((\mathbf{f} = (f_1, \ldots, f_m), \mathbf{g} = (g_1, \ldots, g_m)) \in \mathbb{K}[x_1, \ldots, x_n]^m \times \mathbb{K}[x_1, \ldots, x_n]^m$.
Question : Find – if any – a pair of invertible matrices $(\mathbf{A}, \mathbf{B}) \in \text{GL}_n(\mathbb{K}) \times \text{GL}_m(\mathbb{K})$ such that:

$$\mathbf{g}(\mathbf{x}) = \mathbf{f}(\mathbf{x} \cdot \mathbf{A}) \cdot \mathbf{B}, \text{ with } \mathbf{x} = (x_1, \ldots, x_n)^{\mathrm{T}}.$$

IP is not NP-Hard unless the polynomial-hierarchy collapses, [53,54]. There are quite few algorithms, such as [53,29,13], for solving IP. In particular, [29] proposed to solve IP by reducing it to a system of (overdefined) nonlinear equations whose variables are the unknown coefficients of the matrices. It was conjectured in [29], but never proved, that the corresponding system of nonlinear equations can be solved in polynomial time as soon as the IP instances considered are not homogeneous. Indeed, by slicing of the polynomials degree by degree, one can find equations in the coefficients of the transformation allowing one to recover the transformation. More recently, [14] presented exponential (in the number of variables n) algorithms for solving quadratic homogeneous instances over finite fields. If $\mathbb{K} = \mathbb{F}_2$, [14] described an heuristic algorithm of complexity $2^{n/2}$. For bigger fields $\mathbb{K} = \mathbb{F}_q, q > 2$, [14] gives a probabilistic algorithm of complexity $q^{2n/3}$. In [9], the authors also considered a special case where IP can be solved in polynomial-time (when $\mathbf{f} = (x_1^d, \ldots, x_n^d)$).

2.3 Definitions

We recall below some definitions introduced in [5] that we adopt in this paper. This will allow to state the properties of our fully homomorphic scheme in a general framework.

Definition 2.3.1 ($\mathcal{C}$-Evaluation Scheme)**.** Let $\mathcal{C}$ be a set of circuits. A $\mathcal{C}$-evaluation scheme for $\mathcal{C}$ is a tuple of probabilistic polynomial-time algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ such that:

- $\mathsf{Gen}(1^\lambda, \alpha)$ is the key generation algorithm. It takes two inputs, security parameter λ and auxiliary input α, and outputs a key triple $(\mathsf{pk}, \mathsf{sk}, \mathsf{evk})$, where pk is the key used for encryption, sk is the key used for decryption and evk is the key used for evaluation.
- $\mathsf{Enc}(\mathsf{pk}, m)$ is the encryption algorithm. As input it takes the encryption key pk and a plaintext m. Its output is a ciphertext c.
- $\mathsf{Dec}(\mathsf{sk}, c)$ is the decryption algorithm. As input it takes the decryption key sk and a ciphertext c. Its output is the message m. We have $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m$.
- $\mathsf{Eval}(\mathsf{evk}, C, c_1, \ldots, c_n)$ is the evaluation algorithm. It takes as inputs the evaluation key evk, a circuit $C \in \mathcal{C}$ and a tuple of inputs that can be a mix of ciphertexts and previous evaluation results. It produces an evaluation output.

In what follows, $\mathcal{P}$ denotes the plaintext space. When performing homomorphic computations the image of Eval can be disjoint from the image of Enc. For this reason formal definitions often refer to space of fresh ciphertexts $\mathcal{X}$ to distinguish them from ciphertexts outputs of Eval.

Definition 2.3.2 (Fresh Ciphertext Space)**.** Let $(\mathsf{pk}, \mathsf{sk}, \mathsf{evk}) \leftarrow \mathsf{Gen}(1^\lambda)$. The fresh ciphertext space is defined as

$$\mathcal{X} = \{c \mid \exists m \in \mathcal{P} \text{ such that } \Pr[\mathsf{Enc}(\mathsf{pk}, m) = c > 0]\}.$$

We can formalize correctness of a $\mathcal{C}$-evaluation scheme.

Definition 2.3.3 (Correct Decryption)**.** A $\mathcal{C}$-evaluation scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ is said to *correctly decrypt* if $\forall m \in \mathcal{P}$:

$$\Pr[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}((\mathsf{pk}, m)) = m] = 1,$$

where sk and pk are outputs of $\mathsf{Gen}(1^\lambda, \alpha)$.

Definition 2.3.4 (Correct Evaluation)**.** A $\mathcal{C}$-evaluation scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ is said to *correctly evaluate* all circuits in $\mathcal{C}$ if for all $c_i \in \mathcal{X}$, where $m_i \leftarrow \mathsf{Dec}(\mathsf{evk}, c_i)$, for every $C \in \mathcal{C}$ and some negligible function ϵ,

$$\Pr[\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{evk}, C, c_1, \ldots, c_n)) = C(m_1, \ldots, m_n)] = 1 - \varepsilon(\lambda)$$

where sk and pk are outputs of $\mathsf{Gen}(1^\lambda, \alpha)$.

The next definition formalize the requirement that homomorphic operations do not result in much ciphertext expansion, and the output length depends only on the security parameter.

Definition 2.3.5 (Compactness)**.** A $\mathcal{C}$-evaluation scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ is said to be *compact* if there is a polynomial p, such that for any key-triple $(\mathsf{pk}, \mathsf{sk}, \mathsf{evk})$ output by $\mathsf{Gen}(1^\lambda, \alpha)$, any circuit $C \in \mathcal{C}$ and all ciphertexts $c_i \in \mathcal{X}$, the size of the output $\mathsf{Eval}(\mathsf{evk}, C, c_1, \ldots, c_n)$ is not more than $p(\lambda)$ bits, independent of the size of the circuit.

Definition 2.3.6 (Compactly Evaluate)**.** A $\mathcal{C}$-evaluation scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ is said to *compactly evaluate* all circuits in $\mathcal{C}$ if scheme is compact and correct.

Sometimes it is desirable to perform computation in two or more stages, where the results from one stage could be fed in as inputs for a later stage. This requires that outputs of Eval also be valid inputs to Eval, something that is not required by Definition 2.3.4.

Definition 2.3.7 (Staged computation)**.** A computation $\mathbf{C}_{i,n}$ in i-stages of width n is defined by a set of circuits $C_{k\ell}$ index by $1 \leq k \leq i, i \leq \ell \leq n$, where $C_{k\ell}$ has kn inputs. Starting with initial plaintexts $m_{01}, m_{02}, \ldots, m_{0n}$ we compute in stages

$$m_{k\ell} = C_{k\ell}(m_{01}, m_{02}, \ldots, m_{0n}, m_{k-1,1}, \ldots, m_{k-1,n})$$

for each stage $1 \leq k \leq i$ an1 $\leq \ell \leq n$. The output of the staged computation is m_{i1}, m_{i2}, m_{in}.

It is possible to stage homomorphic comptuation in using the same approach as describe above. Let $(\mathsf{pk}, \mathsf{evk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ and let $c_{01}, c_{02}, \ldots, c_{0n}$ be a sequence of ciphertexts from $\mathcal{X}$. The ciphertexts $c_{k\ell}$ for $1 \leq k \leq i, i \leq \ell \leq n$ can be computed by applying Eval in stages.

$$c_{k\ell} = \mathsf{Eval}(\mathsf{evk}, c_{01}, C_{k\ell}, c_{02}, \ldots, c_{0n}, c_{k-1,1}, \ldots, c_{k-1,n})$$

This homomorphic evaluation in stages is known i-hop homomorphic encryption. We can now proceed to define formal notions of correctness for i-Hop and ∞-hop homomorphic schemes.

Definition 2.3.8 (i-Hop correctness)**.** Let $(\mathsf{pk}, \mathsf{evk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda, \alpha)$ and let $\mathbf{C}_{i,n} = C_{k\ell}$ be any staged computation where n is polynomial in λ. and $c_0 = (c_{01}, ..., c_{0n}) in X$ n. A $\mathcal{C}$-evaluation scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ is i-hop correct if

$$\Pr[\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{evk}, \mathbf{C}_{i,n}, c_0) = C_{i,n}(\mathsf{Dec}(sk, c_0))] = 1 - \epsilon(\lambda),$$

where ϵ is a negligible function and the probability is taken over the distribution of possible outcomes of the Eval algorithm outputs.

Definition 2.3.9 (i-Hop)**.** Let $i \in N$. We say that a $\mathcal{C}$-evaluation scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ is i-hop if j-hop correctness holds for all j with $1 \leq j \leq i$.

Definition 2.3.10 (∞-Hop)**.** A $\mathcal{C}$-evaluation scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ is said to be ∞-*hop* if j-hop correctness holds for all j.

∞-*hop* is a natural extension of existing definitions (e.g., i-Hop, multi-hop, poly-hop), and allows for an unlimited number of hops.

3 Symmetric Multivariate Encryption Scheme

We describe here a simple randomized secret-key encryption scheme that will serve as a primitive of our FHE scheme (described in Section 4). The symmetric scheme is a variant of the CCA-secure scheme of [38, Section 5] instantiated with a *composition chain* as defined below:

Definition 3.0.11. Let $\ell > 1$ be an integer and $\mathbf{f}_1, \ldots, \mathbf{f}_\ell \in \mathbb{F}_2[y_1, \ldots, y_n]^n$ be multivariate quadratic polynomials. We shall call composition chain:

$$\mathbf{f} := \mathbf{f}_\ell \circ \cdots \circ \mathbf{f}_1 \in \mathbb{F}_2[y_1, \ldots, y_n]^n.$$

We shall call length of composition chain the integer ℓ.

It can be mentioned that multivariate composition chains have been already considered in the literature as a basic building block for pseudo-random number generators as well as hash functions, e.g. [47,39]. Note that these papers use composition chains of structured polynomials. Here, we consider composition chains of random polynomials.

3.1 Description

The cryptosystem is parametrized with two parameters $n, \ell \in \mathbb{N}$, where n is the plaintext length in bits and $\ell \geq 2$ is the length of composition chain. The secret-key algorithm, that will be denoted by MQSE, is defined by a a set of three polynomial time algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ which are defined as follows:

- $\mathsf{sk} = (\mathbf{M}, \mathbf{f}) \leftarrow \mathsf{Gen}(1^{(n,\ell)})$ which returns a matrix chosen uniformly at random $\mathbf{M} \in \mathrm{GL}_{2n}(\mathbb{F}_2)$ and a composition chain $\mathbf{f}$ with n-tuples of multivariate quadratic polynomials $\mathbf{f}_1, \ldots, \mathbf{f}_\ell \in \mathbb{F}_2[y_1, \ldots, y_n]^n$ chosen uniformly at random.

– $\mathsf{Enc}(\mathsf{sk}, \mathbf{m})$ is a probabilistic algorithm that takes a message $\mathbf{m} \in \mathbb{F}_2^n$ selects an $\mathbf{r} \in \mathbb{F}_2^n$ uniformly at random and returns an encryption of $\mathbf{m}$. It is defined as:

$$\mathsf{Enc}_{\mathsf{sk}}(\mathbf{m}, \mathbf{r}) := \mathbf{M} \begin{bmatrix} \mathbf{m} + \mathbf{f}(\mathbf{r}) \\ \mathbf{r} \end{bmatrix}. \tag{2}$$

– $\mathsf{Dec}(\mathsf{sk}, \mathbf{c})$ is deterministic algorithm that uses the secret key sk to return a message $\mathbf{m} \in \mathbb{F}_2$ decrypted from ciphertext $\mathbf{c} \in \mathbb{F}_2$. It defined as follows:

$$\mathbf{m} = \mathsf{Dec}_{\mathsf{sk}}(\mathbf{c}) := \Big[\pi_{1,2}(\mathbf{M}^{-1}\mathbf{c}) + \mathbf{f}(\pi_{2,2}(\mathbf{M}^{-1}\mathbf{c}))\Big]. \tag{3}$$

Theorem 3.1.1. *The* MQSE *scheme is correct.*

Proof. Let $\mathbf{c} = \mathsf{Enc}_{\mathsf{sk}}(\mathbf{m}, \mathbf{r})$ be as in (2). The correctness of the algorithm is simple to show from expanding $\mathbf{M}^{-1}\mathbf{c}$.

$$\mathbf{M}^{-1}\mathbf{c} = \mathbf{M}^{-1}\mathsf{Enc}_{\mathsf{sk}}(\mathbf{m}, \mathbf{r}) = \begin{bmatrix} \mathbf{m} + \mathbf{f}(\mathbf{r}) \\ \mathbf{r} \end{bmatrix}$$

Substituting back in and evaluating the projection functions completes the proof.

$$\mathsf{Dec}_{\mathsf{sk}}(\mathbf{c}) = [(\mathbf{m} + \mathbf{f}(\mathbf{r})) + \mathbf{f}(\mathbf{r})] = \mathbf{m}$$

□

3.2 Security Analysis

The difference between MQSE and the scheme of [38, Section 5] is first on the use of the matrix $\mathbf{M}$ in MQSE. However, the fundamental difference is that the scheme of [38, Section 5] must be instantiated with a pseudo-random function to achieve CCA-security.
An adversary with an Oracle capable of generating k plaintext-ciphertext pairs $\{(\mathbf{m}_i, \mathbf{c}_i) \in \mathbb{F}_2^{2n} \times \mathbb{F}_2^{2n} \mid 1 \leq i \leq k\}$ will be able to interpolate the decryption polynomials by solving for their coefficients.

$$[\mathbf{m}_1 \dots \mathbf{m}_k] = \mathbf{D} \begin{bmatrix} (\mathbf{c}_1)_1 & \dots & (\mathbf{c}_k)_1 \\ \vdots & \dots & \vdots \\ (\mathbf{c}_1)_{2n} & \dots & (\mathbf{c}_k)_{2n} \\ \vdots & \dots & \vdots \\ (\mathbf{c}_1)_{2n}(\mathbf{c}_1)_{2n-1} & \dots & (\mathbf{c}_k)_{2n}(\mathbf{c}_k)_{2n-1} \\ \vdots & \dots & \vdots \\ (\mathbf{c}_1)_{2n} \dots (\mathbf{c}_1)_1 & \dots & (\mathbf{c}_k)_{2n} \dots (\mathbf{c}_k)_1 \end{bmatrix} \tag{4}$$

Thus the adversary can learn the decryption polynomial in $\theta(\left[\sum_{i=1}^{2^\ell} \binom{i}{2n}\right]^3)$ time/space, which approaches exponential as $2^\ell \to 2n$. The above technique will also apply to any symmetric scheme in code book mode.

4 Multivariate Fully Homomorphic Encryption

In this section we extend the MQSE scheme described in Section 3 to an ∞-hop $\mathcal{C}$-evaluation homomorphic scheme by providing a general method for constructing arbitrary homomorphisms of sets of multivariate polynomial functions. We also providing specific constructions for multiplication by a matrix, XOR, and AND. For brevity, we skip NOT, which can be represented using XOR and a constant. Finally, we define homomorphic public key and show that it has a functionally complete set of homomorphic operators that can be used by an untrusted server to evaluate arbitrarily deep circuits.

4.1 Generic Approach

Let $\mathbf{g} : (\mathbb{F}_2^n)^k \to \mathbb{F}_2^n$ be a k-ary multivariate polynomial function. We adapt the approach described in (1) (Section 1.2) to provide a general algorithm for constructing the homomorphic equivalent $\mathcal{H}[\mathbf{g}]$ of $\mathbf{g}$ for ciphertexts. To do so, we consider the secret-key scheme of Section 3. Let then $S = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an instantiated $\mathsf{MQSE}(n, \ell)$ scheme with secret key $\mathsf{sk} = (\mathbf{M}, \mathbf{f}) \leftarrow \mathsf{Gen}(1^{(n,\ell)})$. The idea of the `FHE` scheme is to derive a multivariate function $\mathcal{H}[\mathbf{g}]$ such that

$$\mathcal{H}[\mathbf{g}](\mathbf{x}_1, \ldots, \mathbf{x}_k) = \mathsf{Enc}_{\mathsf{sk}} \circ \mathbf{g}(\mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_1), \ldots, \mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_k)), \text{with } \mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathbb{F}_2^{2n}.$$

To do so, we introduce the concept of *obfuscated composition chain* and *re-randomization transform*. Before that, we define some projection operators. These operators will allow to simplify the description of our `FHE` scheme.

Definition 4.1.1. Let $k \geq 1$ be an integer. For all $i, 1 \leq i \leq k$, we define the vector projection $\pi_{i,k} : \mathbb{F}_2^{kn} \to \mathbb{F}_2^n$ as follows:

$$\pi_{i,k} : \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{kn} \end{bmatrix} \mapsto \begin{bmatrix} x_{(i-1)n+1} \\ x_{(i-1)n+2} \\ \vdots \\ x_{in} \end{bmatrix} .$$

The next function will be useful for generating a new random vector from several input ciphertexts.

Definition 4.1.2. Let $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathbb{F}_2^{2n}$ and $\mathbf{R}_1, \ldots, \mathbf{R}_k \in \mathrm{GL}_n(\mathbb{F}_2)$ be matrices chosen uniformly at random. We define a *re-randomization transform* $\mathcal{R}_{[1,\ldots,k]} : (\mathbb{F}_2^{2n})^k \to \mathbb{F}_2^n$ as follows:

$$\mathcal{R}_{[1,\ldots,k]}(\mathbf{x}_1, \ldots, \mathbf{x}_k) := \sum_{j=1}^{k} \mathbf{R}_j \pi_{2,2}(\mathbf{M}^{-1}\mathbf{x}_j).$$

Definition 4.1.3. Let $\ell > 1$ be an integer, $\mathbf{f}_1, \ldots, \mathbf{f}_\ell \in \mathbb{F}_2[y_1, \ldots, y_n]^n$ be multivariate quadratic polynomials and a composition chain:

$$\mathbf{f} := \mathbf{f}_\ell \circ \cdots \circ \mathbf{f}_1 \in \mathbb{F}_2[y_1, \ldots, y_n]^n.$$

Let also $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathbb{F}_2^{2n}$ and $\mathbf{K}_1, \ldots, \mathbf{K}_\ell \in \mathrm{GL}_{kn}(\mathbb{F}_2)$ be matrices chosen uniformly at random. We shall call the sequence $\mathbf{f}' := \mathbf{f}'_1 \circ \cdots \circ \mathbf{f}'_\ell$ an *obfuscated composition chain* where:

$$\mathbf{f}'_1(\mathbf{x}_1, \ldots, \mathbf{x}_k) := \mathbf{K}_1 \begin{bmatrix} \mathbf{f}_1\Big(\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x}_1)\Big) \\ \vdots \\ \mathbf{f}_1\Big(\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x}_k)\Big) \\ \mathbf{f}_1\Big(\mathcal{R}_{[1,\ldots,k]}(\mathbf{x}_1, \ldots, \mathbf{x}_k)\Big) \end{bmatrix}, \tag{5}$$

and for $i > 1$:

$$\mathbf{f}'_i(\mathbf{x}) := \mathbf{K}_i \begin{bmatrix} \mathbf{f}_i\Big(\pi_{1,k+1}(\mathbf{K}_{i-1}^{-1}\mathbf{x})\Big) \\ \vdots \\ \mathbf{f}_i\Big(\pi_{k,k+1}(\mathbf{K}_{i-1}^{-1}\mathbf{x})\Big) \\ \mathbf{f}_i\Big(\pi_{k+1,k+1}(\mathbf{K}_{i-1}^{-1}\mathbf{x})\Big) \end{bmatrix} \text{ for all } \mathbf{x} \in \mathbb{F}_2^{(k+1)n}.$$

Remark that the function computed by an obfuscated composition chain $\mathbf{f}'$ is a linear transformation of the original composition chain $\mathbf{f}$, i.e.:

$$\mathbf{f}'(\mathbf{x}_1,\ldots,\mathbf{x}_k) = \mathbf{K}_\ell \begin{bmatrix} \mathbf{f}\Big(\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x}_1)\Big) \\ \vdots \\ \mathbf{f}\Big(\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x}_k)\Big) \\ \mathbf{f}\Big(\mathcal{R}_{[1,\ldots,k]}(\mathbf{x}_1,\ldots,\mathbf{x}_k)\Big) \end{bmatrix} \text{ for all } \mathbf{x}_1,\ldots,\mathbf{x}_k \in \mathbb{F}_2^{2n}. \tag{6}$$

With the machinery for an obfuscated composition chain out the way, we can express the $\mathcal{H}[\mathbf{g}](\mathbf{x}_1,\ldots,\mathbf{x}_k)$ in terms of obfuscated composition chain $\mathbf{f}'$ of $\mathbf{f}$. This function is defined such that:

$$\mathcal{H}[\mathbf{g}](\mathbf{x}_1,\ldots,\mathbf{x}_k) = \mathsf{Enc}_{\mathsf{sk}} \circ \mathbf{g}\Big(\mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_1),\ldots,\mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_k)\Big). \tag{7}$$

We will see that $\mathcal{H}[\mathbf{g}](\mathbf{x}_1,\ldots,\mathbf{x}_k) =$

$$\mathbf{M}\begin{bmatrix} \mathbf{g}\Big(\pi_{1,2}(\mathbf{M}^{-1}\mathbf{x}_1)+\pi_{1,k+1}(\mathbf{K}^{-1}\mathbf{f}'),\ldots,\pi_{1,2}(\mathbf{M}^{-1}\mathbf{x}_k)+\pi_{k,k+1}(\mathbf{K}^{-1}\mathbf{f}')\Big)+\pi_{k,k+1}(\mathbf{K}^{-1}\mathbf{f}') \\ \mathcal{R}_{[1,\ldots,k]}(\mathbf{x}_1,\ldots,\mathbf{x}_k) \end{bmatrix} =$$

$$\mathbf{M}\begin{bmatrix} \mathbf{g}\Big(\pi_{1,2}(\mathbf{M}^{-1}\mathbf{x}_1)+\mathbf{f}(\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x}_1)),\ldots,\pi_{1,2}(\mathbf{M}^{-1}\mathbf{x}_k)+\mathbf{f}(\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x}_k))\Big)+\mathbf{f}\Big(\mathcal{R}_{[1,\ldots,k]}(\mathbf{x}_1,\ldots,\mathbf{x}_k)\Big) \\ \mathcal{R}_{[1,\ldots,k]}(\mathbf{x}_1,\ldots,\mathbf{x}_k) \end{bmatrix}$$

From (7) we see the order of $\mathbf{g}$ will play a big role in the complexity of the multivariate polynomials for homomorphic evaluation.

4.2 Matrix Multiplication

We apply the generic construction from Section 4.1 to the simple case of homomorphic matrix multiplication. Let $\mathbf{T} \in \mathrm{GL}_n(\mathbb{F}_2)$ and $S = (\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ be an instantiated $\mathsf{MQSE}(n,2)$ scheme with secret-key $\mathsf{sk} = (\mathbf{M},\mathbf{f}) \leftarrow \mathsf{Gen}(1^{(n,2)})$. We construct a function $\mathcal{H}[\mathbf{T}]$ for the homomorpic evaluation of a matrix multiplication by $\mathbf{T}$ on ciphertext vectors.

Definition 4.2.1. Let $\mathsf{sk} = (\mathbf{M},\mathbf{f}) \leftarrow \mathsf{Gen}(1^{(n,2)})$ and $\mathbf{f}_1,\mathbf{f}_2 \in \mathbb{F}_2[y_1,\ldots,y_n]^n$ be the functions in the composition chain $\mathbf{f}$. Also, let $\mathbf{R} \in \mathrm{GL}_n(\mathbb{F}_2)$, and $\mathbf{K}_1,\mathbf{K}_2 \in \mathrm{GL}_{2n}(\mathbb{F}_2)$ be chosen as described in Section 4.1. Then the obfuscated composition chain $\mathbf{f}' : \mathbb{F}_2^{2n} \to \mathbb{F}_2^{2n}$ for $\mathbf{f}$ is as follows:

$$\mathbf{f}'_1(\mathbf{x}) := \mathbf{K}_1 \begin{bmatrix} \mathbf{f}_1(\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x})) \\ \mathbf{f}_1(\mathbf{R}\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x})) \end{bmatrix}, \forall \mathbf{x} \in \mathbb{F}_2^{2n}$$

$$\mathbf{f}'_2(\mathbf{x}) := \mathbf{K}_2 \begin{bmatrix} \mathbf{f}_2(\pi_{1,2}(\mathbf{K}_1^{-1}\mathbf{x})) \\ \mathbf{f}_2(\pi_{2,2}(\mathbf{K}_1^{-1}\mathbf{x})) \end{bmatrix}, \forall \mathbf{x} \in \mathbb{F}_2^{2n}$$

$$\mathbf{f}'(\mathbf{x}) := \mathbf{f}'_2(\mathbf{f}'_1(\mathbf{x})), \forall \mathbf{x} \in \mathbb{F}_2^{2n}.$$

It can be verified that this is the obfuscated composition chain of Definition 4.1.3 with $k = 2$.

We can now define our homomorphic function for matrix multiplication:

Proposition 4.2.1. *Let the notations be as in Definition 4.2.1. For $\mathbf{T} \in \mathrm{GL}_n(\mathbb{F}_2)$, we define $\mathcal{H}[\mathbf{T}] : \mathbb{F}_2^{2n} \to \mathbb{F}_2^{2n}$ as $\mathcal{H}[\mathbf{T}](\mathbf{x}) :=$*

$$\mathbf{M}\begin{bmatrix} \mathbf{T}[\pi_{1,2}(\mathbf{M}^{-1}\mathbf{x})+\pi_{1,2}(\mathbf{K}_2^{-1}\mathbf{f}'(\mathbf{x}))]+\pi_{2,2}(\mathbf{K}_2^{-1}\mathbf{f}'(\mathbf{x})) \\ \mathbf{R}\cdot\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x}) \end{bmatrix}. \tag{8}$$

Then, it holds that:

$$\mathsf{Dec}_{\mathsf{sk}}(\mathcal{H}[\mathbf{T}](\mathbf{x})) = \mathbf{T}\cdot\mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbb{F}_2^{2n}.$$

Proof. Let $\mathbf{f}(y_1,\ldots,y_n) := \mathbf{f}_2(\mathbf{f}_1(y_1,\ldots,y_n))$ and $\mathbf{x} \in \mathbb{F}_2^{2n}$. From Definition 4.2.1 and according to (6), we have that

$$\mathbf{f}'(\mathbf{x}) = \mathbf{K}_2 \begin{bmatrix} \mathbf{f}(\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x})) \\ \mathbf{f}(\mathbf{R}\cdot\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x})) \end{bmatrix}.$$

Substitution into the definition of $\mathcal{H}[\mathbf{T}]$ yields that

$$\begin{aligned}\mathcal{H}[\mathbf{T}](\mathbf{x}) &= \mathbf{M}\begin{bmatrix} \mathbf{T}\cdot\mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}) + \mathbf{f}(\mathbf{R}\cdot\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x})) \\ \mathbf{R}\cdot\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x}) \end{bmatrix} \\ &= \mathsf{Enc}_{\mathsf{sk}}(\mathbf{T}\cdot\mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}), \mathbf{R}\cdot\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x})).\end{aligned}$$

Therefore:

$$\mathsf{Dec}_{\mathsf{sk}}(\mathcal{H}[\mathbf{T}](\mathbf{x})) = \mathbf{T}\cdot\mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}).$$

□

An alternate formulation of the homomorphic function (8) is

$$\mathcal{H}[\mathbf{T}]\Big(\mathsf{Enc}_{\mathsf{sk}}(\mathbf{m},\mathbf{r})\Big) = \mathsf{Enc}_{\mathsf{sk}}\Big(\mathbf{T}\cdot\mathbf{m}, \mathbf{R}\cdot\pi_{2,2}(\mathbf{M}^{-1}\mathsf{Enc}_{\mathsf{sk}}(\mathbf{m},\mathbf{r}))\Big).$$

That is, the homomorphic function is equivalent to re-encryption of the linearly transformed decrypted input with a modified random component $\mathbf{r}$.

4.3 Binary Operations

We apply the generic construction from Section 4.1 to other operations such as bitwise XOR (denoted by $+$) and bitwise AND (denoted by $\times$). As before, let $S = (\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ be an instantiated $\mathsf{MQSE}(n,2)$ scheme with secret key $\mathsf{sk} = (\mathbf{M},\mathbf{f}) \leftarrow \mathsf{Gen}(1^{(n,2)})$. We construct functions $\mathcal{H}[+]$ and $\mathcal{H}[\times]$ for homomorpic evaluation of $+$ and $\times$ on ciphertext vectors $\mathbf{x}_1,\mathbf{x}_2 \in \mathbb{F}_2^{2n}$.

Definition 4.3.1. Let $\mathsf{sk} = (\mathbf{M},\mathbf{f}) \leftarrow \mathsf{Gen}(1^{(n,2)})$ and $\mathbf{f}_1,\mathbf{f}_2 \in \mathbb{F}_2[y_1,\ldots,y_n]^n$ be the functions in the composition chain of $\mathbf{f}$. Also, let $\mathbf{R}_1,\mathbf{R}_2 \in \mathrm{GL}_n(\mathbb{F}_2), \mathbf{K}_1,\mathbf{K}_2 \in \mathrm{GL}_{3n}(\mathbb{F}_2)$ and $\mathcal{R}_{[1,2]}(\mathbf{x}_1,\mathbf{x}_2) = \mathbf{R}_1\cdot\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x}_1) + \mathbf{R}_2\cdot\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x}_2)$ be chosen as described in Section 4.1. Then, the obfuscated composition chain $\mathbf{f}' : \mathbb{F}_2^{4n} \to \mathbb{F}_2^{3n}$ for $\mathbf{f}$ is as follows:

$$\begin{aligned}\mathbf{f}'_1(\mathbf{x}_1,\mathbf{x}_2) &:= \mathbf{K}_1\begin{bmatrix} \mathbf{f}_1(\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x}_1)) \\ \mathbf{f}_1(\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x}_2)) \\ \mathbf{f}_1\Big(\mathcal{R}_{[1,2]}(\mathbf{x}_1,\mathbf{x}_2)\Big) \end{bmatrix}, \forall(\mathbf{x}_1,\mathbf{x}_2) \in \mathbb{F}_2^{2n}\times\mathbb{F}_2^{2n} \\ \mathbf{f}'_2(\mathbf{x}) &:= \mathbf{K}_2\begin{bmatrix} \mathbf{f}_2(\pi_{1,3}(\mathbf{K}_1^{-1}\mathbf{x})) \\ \mathbf{f}_2(\pi_{2,3}(\mathbf{K}_1^{-1}\mathbf{x})) \\ \mathbf{f}_2(\pi_{3,3}(\mathbf{K}_1^{-1}\mathbf{x})) \end{bmatrix}, \forall\mathbf{x} \in \mathbb{F}_2^{3n}, \text{ and} \\ \mathbf{f}'(\mathbf{x}_1,\mathbf{x}_2) &:= \mathbf{f}'_2(\mathbf{f}'_1(\mathbf{x}_1,\mathbf{x}_2)), \forall(\mathbf{x}_1,\mathbf{x}_2) \in \mathbb{F}_2^{2n}\times\mathbb{F}_2^{2n}.\end{aligned}$$

We also define:

$$\phi_i = \mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_i) = \pi_{1,2}(\mathbf{x}_i) + \pi_{i,3}(\mathbf{K}_2^{-1}\mathbf{f}'(\mathbf{x}_1,\mathbf{x}_2)), \text{for } i \in \{1,2\}.$$

It can be verified that this is the obfuscated composition chain of Definition 4.1.3 with $k = 3$.

We are now in position to derive special public function allowing to perform more general homomorphic operations than matrix multiplication (Section 4.2). We first describe the case of component-wise XOR ($+$).

Proposition 4.3.1. *Let the notations be as in Definition 4.3.1. The homomorphic function for component-wise* XOR *($+$) is:*

$$\mathcal{H}[+](\mathbf{x}_1,\mathbf{x}_2) := \mathbf{M}\begin{bmatrix} \phi_1+\phi_2+\pi_{3,3}(\mathbf{K}_2^{-1}\mathbf{f}'(\mathbf{x}_1,\mathbf{x}_1)) \\ \mathcal{R}_{[1,2]}(\mathbf{x}_1,\mathbf{x}_2) \end{bmatrix}$$

Then, for all $(\mathbf{x}_1, \mathbf{x}_2) \in \mathbb{F}_2^{2n} \times \mathbb{F}_2^{2n}$*, it holds that*

$$\mathsf{Dec}_{\mathsf{sk}}(\mathcal{H}[+](\mathbf{x}_1, \mathbf{x}_2)) = \mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_1) + \mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_2).$$

Proof. Let $\mathbf{f} := \mathbf{f}_2 \circ \mathbf{f}_1$. From Definition 4.1.3 and Definition 4.3.1, we see that

$$\mathbf{f}'(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{K}_2 \begin{bmatrix} \mathbf{f}(\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x}_1)) \\ \mathbf{f}(\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x}_2)) \\ \mathbf{f}\Big(\mathcal{R}_{[1,2]}(\mathbf{x}_1, \mathbf{x}_2)\Big) \end{bmatrix}.$$

Substitution into the definition of $\mathcal{H}[+]$ yields that

$$\begin{aligned} \mathcal{H}[+](\mathbf{x}_1, \mathbf{x}_2) &= \mathbf{M} \begin{bmatrix} \mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_1) + \mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_2) + \mathbf{f}(\mathcal{R}_{[1,2]}(\mathbf{x}_1, \mathbf{x}_2)) \\ \mathcal{R}_{[1,2]}(\mathbf{x}_1, \mathbf{x}_2) \end{bmatrix} \\ &= \mathsf{Enc}_{\mathsf{sk}}(\mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_1) + \mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_2), \mathcal{R}_{[1,2]}(\mathbf{x}_1, \mathbf{x}_2)). \end{aligned}$$

Therefore:

$$\mathsf{Dec}_{\mathsf{sk}}(\mathcal{H}[+](\mathbf{x}_1, \mathbf{x}_2)) = \mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_1) + \mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_2).$$

□

The rationale for explicitly describing $\mathcal{H}[+](\mathbf{x}_1, \mathbf{x}_2)$ is that XOR is required to represent boolean circuits in algebraic normal form, which will come in useful later in the paper. For the same reason, we apply the same approach to component-wise AND ($\times$).

Proposition 4.3.2. *Let the notations be as in Definition 4.3.1. The homomorphic function for component-wise* AND *($\times$) is*

$$\mathcal{H}[\times](\mathbf{x}_1, \mathbf{x}_2) := \mathbf{M} \begin{bmatrix} \phi_1\phi_2 + \pi_{3,3}(\mathbf{K}_2^{-1}\mathbf{f}'(\mathbf{x}_1, \mathbf{x}_2)) \\ \mathcal{R}_{[1,2]}(\mathbf{x}_1, \mathbf{x}_2) \end{bmatrix}.$$

Then, for all $(\mathbf{x}_1, \mathbf{x}_2) \in \mathbb{F}_2^{2n} \times \mathbb{F}_2^{2n}$*, it holds that*

$$\mathsf{Dec}_{\mathsf{sk}}(\mathcal{H}[\times](\mathbf{x}_1, \mathbf{x}_2)) = \mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_1) \times \mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_2).$$

Proof. The function $\mathcal{H}[\times](\mathbf{x}_1, \mathbf{x}_2)$ is defined such that:

$$\begin{aligned} \mathcal{H}[\times](\mathbf{x}_1, \mathbf{x}_2) &= \mathbf{M} \begin{bmatrix} (\mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_1) \times \mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_2)) + \mathbf{f}(\mathcal{R}_{[1,2]}(\mathbf{x}_1, \mathbf{x}_2)) \\ \mathcal{R}_{[1,2]}(\mathbf{x}_1, \mathbf{x}_2) \end{bmatrix} \\ &= \mathsf{Enc}_{\mathsf{sk}}(\mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_2) \times \mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_2), \mathcal{R}_{[1,2]}(\mathbf{x}_1, \mathbf{x}_2)). \end{aligned}$$

□

In fact, the same idea can be generalized for any function g on two messages.

Proposition 4.3.3. *Let the notations be as in Definition 4.3.1 and* $\mathbf{g} : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2^n$ *be a function. The homomorphic function* $\mathcal{H}[\mathbf{g}]$ *associated to* g *is*

$$\mathcal{H}[\mathbf{g}](\mathbf{x}_1, \mathbf{x}_2) := \mathbf{M} \begin{bmatrix} \mathbf{g}(\phi_1, \phi_2) + \pi_{3,3}(\mathbf{K}_2^{-1}\mathbf{f}'(\mathbf{x}_1, \mathbf{x}_2)) \\ \mathcal{R}_{[1,2]}(\mathbf{x}_1, \mathbf{x}_2) \end{bmatrix}.$$

For all $(\mathbf{x}_1, \mathbf{x}_2) \in \mathbb{F}_2^{2n} \times \mathbb{F}_2^{2n}$*, it holds that*

$$\mathsf{Dec}_{\mathsf{sk}}(\mathcal{H}[g](\mathbf{x}_1, \mathbf{x}_2)) = \mathbf{g}(\mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_1), \mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_2)).$$

Proof. The function $\mathcal{H}[\mathbf{g}](\mathbf{x}, \mathbf{y})$ is defined such that:

$$\begin{aligned} \mathcal{H}[g](\mathbf{x}, \mathbf{y}) &= \mathbf{M} \begin{bmatrix} \mathbf{g}(\mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_1), \mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_2)) + \mathbf{f}(\nu(\mathbf{x}_1, \mathbf{x}_2)) \\ \mathcal{R}_{[1,2]}(\mathbf{x}_1, \mathbf{x}_2) \end{bmatrix} \\ &= \mathsf{Enc}_{\mathsf{sk}}(g(\mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_1), \mathsf{Dec}_{\mathsf{sk}}(\mathbf{x}_2)),). \end{aligned}$$

□

4.4 Formal Definition and Proofs

The goal of this part is to explain how to evaluate homorphically arbitrary circuits. To do so, we first introduce SHIFT operators, that is LEFT-SHIFT(<<) and RIGHT-SHIFT(>>).

Definition 4.4.1. We define the LEFT-SHIFT (<<) and RIGHT-SHIFT(>>) operators as follows:

$$<<: \mathbf{x} = [x_1, x_2, x_3, \ldots, x_n]^T \mapsto [x_2, x_3, \ldots, x_n, 0]^T,$$
$$>>: \mathbf{x} = [x_1, x_2, x_3, \ldots, x_n]^T \mapsto [0, x_1, \ldots, x_{n-1}]^T.$$

We are now in position to define our FHE scheme.

Definition 4.4.2. MQFHE is a $\mathcal{C}$–evaluation scheme consisting of a tuple of probabilistic polynomial time algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{MQSE.Dec})$ defined by:

- Let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{F}_2^{2n}$, we define $\mathcal{H}[\sum_{i=1}^n](\mathbf{x}_1, \ldots, \mathbf{x}_n)$ as the public function which allows to perform the homomorphic evaluation of $\sum_{i=1}^n \mathbf{x}_i$.
- $(\mathsf{pk}, \mathsf{sk}, \mathsf{evk}) \leftarrow \mathsf{Gen}(1^{(n,\ell)})$ is a probablistic polynomial time key generation algorithm. The secret key is generated by evaluating $\mathsf{sk} = \mathsf{MQSE.Gen}(1^{(n,\ell)})$. Next, the evaluation key is generated using the secret key according the methods in sections 4.1, 4.2, and 4.3 by computing $\mathsf{evk} = (\mathcal{H}[\sum_{i=1}^n], \mathcal{H}[\times], \mathcal{H}[<<], \mathcal{H}[>>])$. The public key utilizes the basis vectors $\mathbf{e}_i \in \mathbb{F}_2^n$ of a canonical basis of $\mathbb{F}_2^n$, i.e. such that $(\mathbf{e}_i)_j = \delta_{i,j}$ and zero vectors $\mathbf{0} \in \mathbb{F}_2^n$ combined with $\mathcal{H}[\sum_{i=1}^n]$ to generate the public key

$$\mathsf{pk} = \left(\mathcal{H}\left[\sum_{i=1}^n\right], \{\mathbf{b}_i = \mathsf{MQSE.Enc}(\mathbf{e}_i) \mid 1 \le i \le n\} \cup \{\mathbf{z}_i = \mathsf{MQSE.Enc}(\mathbf{0}) \mid 1 \le i \le n\}\right).$$

  Remark that MQSE.Enc is a randomized encryption, so that each encryption $\mathbf{z}_i$ of zero can be indeed different. $\mathsf{Gen}(1^{(n,\ell)})$ outputs $(\mathsf{pk}, \mathsf{sk}, \mathsf{evk})$.
- $\mathsf{Enc}(\mathsf{pk}, \mathbf{m})$ is a probabilistic encryption algorithm. Encryption of a message $\mathbf{m} \in \mathbb{F}_2^n$ is performed using pk by first choosing an index set $r_1, \ldots, r_n \in \mathbb{F}_2$ to be a set subset of $\{1, \ldots, n\}$ uniformly at random and output the result of the following sum.

$$\mathsf{Enc}(\mathsf{pk}, \mathbf{m}) = \mathsf{pk} = \mathcal{H}\left[\sum_{i=1}^n\right](m_1\mathbf{b}_1 + r_1\mathbf{z}_1, \ldots, m_n\mathbf{b}_n + r_n\mathbf{z}_n). \tag{9}$$

- $\mathsf{Eval}(\mathsf{evk}, C, \mathbf{c}_1, \ldots, \ldots, \mathbf{c}_k)$ is probabilistic polynomial time algorithm for evaluating a circuit $C \in \mathcal{C}$ on ciphertexts $\mathbf{c}_i \in \mathbb{F}_2^{2n}$. As all Boolean circuits can be represented in an equivalent canonical algebraic normal form f_C over $\mathbb{F}_2$ equivalent to C, whose representation is simple to compute from the gate description of C. Evaluating $\mathcal{H}[f_C]$ in terms of evk requires generating evaluation trees for the monomials in the algebraic normal form, which we describe in Theorem 4.4.1. Output $\mathcal{H}[f_C](\mathbf{c}_1, \ldots, \mathbf{c}_2)$.

Theorem 4.4.1. MQFHE *can evaluate all circuits correctly. More formally, let* $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \ldots, x_n]^n$ *be some n-bit polynomial function in algebraic normal form. An untrusted server can compute* $\mathcal{H}[f](\mathbf{c})$*, with* $\mathbf{c} = \mathsf{MQSE.Enc}(\mathbf{x})$*, by evaluating a tree with internal nodes consisting of homomorphic operators and leaf nodes consisting of* $\mathbf{c}$ *or* $\mathbf{c}_i$*.*

Proof. we proceed by demonstrating that evaluating arbitrary circuits on an untrusted server can be performed by evaluating an equivalent tree of functionally complete homomorphic operators: XOR(+), AND(×), LEFT-SHIFT(<<), RIGHT-SHIFT(>>). To handle constant terms we will require n ciphertexts $\mathbf{c}_i = \mathcal{E}_{sk}(\omega_i)$ with $\omega_i \in \mathbb{F}_2^n$ and $(\omega_i)_j = \delta_{i,j}$. In practice RIGHT-SHIFT(>>) is not required for most common circuits, but it simplifies the following proof.
In order to show that we can homomorphically evaluate any arbitrary polynomial function over $\mathbb{F}_2[x_1, \ldots, x_n]$ we will provide a constructive, albeit inefficient, algorithm for building a tree that represents that evaluation.

We start by showing how to build a tree that is the equivalent of a linear monomial x_j at some particular index i. This tree will simply consists of $n-j$ nodes of $\mathcal{H}[<<]$, followed by n nodes of $\mathcal{H}[>>]$, followed by i nodes of $\mathcal{H}[<<]$, with a ciphertext $\mathbf{c}$ as the leaf node.
Next, we consider the set of possible plaintext monomials M for function f.

$$M = \left\{ \prod_{i \in m} i \;\middle|\; m \in \mathcal{P}(\{x_1, \ldots, x_n\})/\emptyset \right\} \tag{10}$$

The homomorphic equivalent of these monomials, $\mathcal{H}[M]$ has a straighforward representation.

$$\mathcal{H}[M] = \left\{ \mathcal{H}\left[\prod_{i \in m} i \right] \;\middle|\; m \in \mathcal{P}(\{x_1, \ldots, x_n\})/\emptyset \right\} \tag{11}$$

As multiplication is associative, we can compute the overall homomorphic product a single term at time. With this observation a tree can be constructed for each possible set index monomial $m \in \mathcal{P}(\{x_1, \ldots, x_n\})/\emptyset$ using the following algorithm. By replacing the leaf nodes in the product tree, with a subtree representing the

Algorithm 1 Build tree for monomial m

```
m' ← m
choose i ∈ m'
m' ← m'/{i}
if |m'|> 0 then
    currentNode ← H[×]
    currentNode.rightChild ← i
else
    currentNode ← i
end if
while |m'|> 0 do
    choose i ∈ m'
    m' ← m'/{i}
    if |m'|> 0 then
        newNode ← H[×]
        newNode.rightChild ← currentNode
        currentNode ← newNode
    end if
    currentNode.leftChild ← i
end while
return currentNode
```

corresponding linear monomial at a desired index, we can now build out trees that represent homomorphic evaluations of functions with monomial m at index j.

$$(f_{m,j})_i = \left\{ \begin{matrix} m \\ 0 \end{matrix} \middle| \begin{matrix} i = j \\ i \neq j \end{matrix} \right. \tag{12}$$

Since $f(\mathbf{x}) = \sum_{i=1}^{n} \omega_i + \sum_{m,j} f_{m,j}(\mathbf{x})$ we can leverage Algorithm 1 with $\mathcal{H}[+]$ insteadof $\mathcal{H}[\times]$ in order chain together the trees corresponding to each $\mathcal{H}[f_{m,j}]$ and $\mathbf{c}_i$ using homomorphic XOR. □

Proposition 4.4.2. MQFHE *has compact ciphertexts and supports ∞-hop evaluations.*

Proof. Ciphertext output from Enc and Eval have a fixed length of twice the corresponding plaintext length. Repeated evaluations of Eval do not accumulate noise and are compact so they can be used for an unlimited number of evaluations without any side-effects.

□

5 Security Analysis

The goal of this part is to analyse the security of the scheme described in Section 4.

We first consider the version of the scheme described in Section 4.2. More precisely, let $\mathbf{R} \in \mathrm{GL}_n(\mathbb{F}_2), \mathbf{M}, \mathbf{K}_1, \mathbf{K}_2 \in \mathrm{GL}_{2n}(\mathbb{F}_2)$ and $\mathbf{f}_1, \mathbf{f}_2 \in \mathbb{F}_2[x_1, \ldots, x_n]^n$. As explained in Section 4.2, we consider the obfuscated composition chain defined as:

$$\begin{aligned}
\mathbf{f}'_1(\mathbf{x}_1) &:= \mathbf{K}_1 \begin{bmatrix} \mathbf{f}_1(\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x}_1)) \\ \mathbf{f}_1(\mathbf{R}\pi_{2,2}(\mathbf{M}^{-1}\mathbf{x}_1)) \end{bmatrix} \\
\mathbf{f}'_2(\mathbf{x}) &:= \mathbf{K}_2 \begin{bmatrix} \mathbf{f}_2(\pi_{1,2}(\mathbf{K}_1^{-1}\mathbf{x})) \\ \mathbf{f}_2(\pi_{2,2}(\mathbf{K}_1^{-1}\mathbf{x})) \end{bmatrix} \\
\mathbf{f}'(\mathbf{x}) &:= \mathbf{f}'_2(\mathbf{f}'_1(\mathbf{x}))
\end{aligned}$$

In our setting, $\mathbf{f}', \mathbf{f}'_2$, and $\mathbf{f}'_1$ are publicly known. The security of the scheme relies on the hardness to recover the matrices $\mathbf{R}, \mathbf{M}, \mathbf{K}_1, \mathbf{K}_2 \in \mathrm{GL}_{2n}(\mathbb{F}_2)$ and/or the polynomials $\mathbf{f}_1, \mathbf{f}_2 \in \mathbb{F}_2[x_1, \ldots, x_n]^n$. Note that $\mathbf{K}_1$ and $\mathbf{K}_2$ allow to reveal $\mathbf{f}_2$.
We can remark that $\mathbf{K}_1$ and $\mathbf{K}_2$ verify a peculiar relation:

$$\mathbf{K}_2^{-1}\mathbf{f}'_2(\mathbf{K}_1\mathbf{x}) := \begin{bmatrix} \mathbf{f}_2(\pi_{1,2}(\mathbf{x})) \\ \mathbf{f}_2(\pi_{2,2}(\mathbf{x})) \end{bmatrix}.$$

Observe that $\mathbf{f}_2(\pi_{1,2}(\mathbf{x})) = \mathbf{f}_2(x_1, \ldots, x_n)$ and $\mathbf{f}_2(\pi_{2,2}(\mathbf{x})) = \mathbf{f}_2(x_{n+1}, \ldots, x_{2n})$.
Let M_1 (rep. M_2) be the set of monomials in $x_1, \ldots, x_n$ (resp. $x_{n+1}, \ldots, x_{2n}$). Given $\mathbf{f}'_2 \in \mathbb{F}_2[x_1, \ldots, x_{2n}]^{2n}$, the problem is then to recover $(\mathbf{S}, \mathbf{U}) \in \mathrm{GL}_{2n}(\mathbb{F}_2) \times \mathrm{GL}_{2n}(\mathbb{F}_2)$ such that $\pi_{1,2}(\mathbf{U} \cdot \mathbf{f}'_2(\mathbf{S}\mathbf{x}))$ only involved monomials from M_1 and $\pi_{2,2}(\mathbf{U} \cdot \mathbf{f}'_2(\mathbf{S}\mathbf{x}))$ only involved monomials from M_2. Stated differently, the instance OBFUSCATED-IP (Obfuscated Isomorphism of Polynomials) we consider is as follows:

OBFUSCATED-IP (oIP)
Input: $\mathbf{g} \in \mathbb{F}_2[x_1, \ldots, x_{2n}]^{2n}$ be quadratic polynomials.
Question: Find $(\mathbf{S}, \mathbf{U}) \in \mathrm{GL}_{2n}(\mathbb{F}_2) \times \mathrm{GL}_{2n}(\mathbb{F}_2)$ such that:

$$\pi_{1,2}(\mathbf{U} \cdot \mathbf{g}(\mathbf{S} \cdot \mathbf{x})) \in \mathbb{F}_2[x_1, \ldots, x_n]^n \text{ and } \pi_{2,2}(\mathbf{U} \cdot \mathbf{g}(\mathbf{S} \cdot \mathbf{x})) \in \mathbb{F}_2[x_{n+1}, \ldots, x_{2n}]^n.$$

OBFUSCATED-IP is a variant of the classical Isomorphism of Polynomials (IP) problem introduced by J. Patarin in [50]. Recall that in IP the problem is to find two matrices between two sets of algebraic equations. Following the approach of [30], we will show that OBFUSCATED-IP can be solved with Gröbner bases techniques.

Proposition 5.0.3. *Let* $\mathbf{g} \in \mathbb{F}_2[x_1, \ldots, x_{2n}]^{2n}$ *and* $(\mathbf{S}, \mathbf{U}) \in \mathrm{GL}_{2n}(\mathbb{F}_2) \times \mathrm{GL}_{2n}(\mathbb{F}_2)$ *be a solution of* oIP *on* $\mathbf{g}$. *Then, the components of* $\mathbf{S}, \mathbf{U}, \mathbf{S}^{-1}$ *and* $\mathbf{U}^{-1}$ *vanish a system of, at most,* $4n^2 + 2\,n\left(\sum_{i=0}^{2}\binom{2n}{i} - \binom{n}{i}\right) \in O(n^3)$ *cubic equations in* $8n^2 \in O(n^2)$ *variables.*

Proof. Let M be the set of monomials in $x_1, \ldots, x_{2n}$. Let also (X, Y) be two formal matrices of size $2n \times 2n$. We define

$$\mathbf{g}^{\pi_{1,2}}(X, Y) = \pi_{1,2}(Y \cdot \mathbf{g}_2(X \cdot \mathbf{x})) = (g_1^{\pi_{1,2}}, \ldots, g_n^{\pi_{1,2}}) \in \mathbb{F}_2[x_1, \ldots, x_n]^n.$$

and

$$\mathbf{g}^{\pi_{2,2}}(X,Y) = \pi_{2,2}(Y \cdot \mathbf{g}_2(X \cdot \mathbf{x})) = (g_1^{\pi_{2,2}}, \ldots, g_n^{\pi_{2,2}}) \in \mathbb{F}_2[x_1, \ldots, x_n]^n.$$

Let $i, 1 \leq i \leq n$ and $m_1 \in \mathrm{M} \setminus \mathrm{M}_1$. We denote by $\mathrm{Coefficient}(m_1, g_i^{\pi_{1,2}})$ the coefficient of m_2 in the polynomial $g_i^{\pi_{1,2}}$. Similarly, for $m_2 \in \mathrm{M} \setminus \mathrm{M}_2$, we denote by $\mathrm{Coefficient}(m_2, g_i^{\pi_{2,2}})$ the coefficient of m_2 in $\mathbf{g}_i^{\pi_{2,2}}$.
We remark that $\mathrm{Coefficient}(m_2, \mathbf{g}^{\pi_{1,2}}(X,Y))$ and $\mathrm{Coefficient}(m_2, \mathbf{g}^{\pi_{1,2}}(X,Y))$ are cubic polynomials in the components of X and Y. We have then $4n^2$ variables. Also, by the very definition of $(S,U) \in \mathrm{GL}_{2n}(\mathbb{F}_2) \times \mathrm{GL}_{2n}(\mathbb{F}_2)$, we have for all $i, 1 \leq i \leq n$:

$$\begin{aligned} \mathrm{Coefficient}(m_1, g_i^{\pi_{1,2}}(S,U)) = 0, \forall m_1 \in \mathrm{M} \setminus \mathrm{M}_2, \\ \mathrm{Coefficient}(m_2, g_i^{\pi_{2,2}}(S,U)) = 0, \forall m_2 \in \mathrm{M} \setminus \mathrm{M}_1. \end{aligned}$$

The number of monomials in M_1 (resp. M_2) is $\sum_{i=0}^{2} \binom{n}{i}$. The number of monomials in M is $\sum_{i=0}^{2} \binom{2n}{i}$ yielding a set of $2\,n \left(\sum_{i=0}^{2} \binom{2n}{i} - \binom{n}{i}\right)$ equations in the components of X and Y.
To conclude, we need to include algebraically the fact that the solutions that we are looking are invertible matrices. To do so, we introduce two new formal matrices X^* and Y^* of size $2n \times 2n$. We can then introduce the following quadratic equations:

$$X^* \cdot X = \mathrm{I}_{2n} \quad X \cdot X^* = \mathrm{I}_{2n} \quad Y \cdot Y^* = \mathrm{I}_{2n} \quad Y^* \cdot Y = \mathrm{I}_{2n},$$

with I_{2n} being the identity matrix of size $2n \times 2n$. □

We now consider the version of the scheme described in Section 4.3. More precisely, let $\mathbf{K}_1, \mathbf{K}_2 \in \mathrm{GL}_{3n}(\mathbb{F}_2)$ and $\mathbf{f}_2 \in \mathbb{F}_2[x_1, \ldots, x_n]^n$. More precisely, we consider the following relation:

$$\mathbf{f}_2'(\mathbf{x}) := \mathbf{K}_2 \begin{bmatrix} \mathbf{f}_2(\pi_{1,3}(\mathbf{K}_1^{-1}\mathbf{x})) \\ \mathbf{f}_2(\pi_{2,3}(\mathbf{K}_1^{-1}\mathbf{x})) \\ \mathbf{f}_2(\pi_{3,3}(\mathbf{K}_1^{-1}\mathbf{x})) \end{bmatrix}$$

As previously, we remark that:

$$\mathbf{K}_2^{-1}\mathbf{f}_2'(\mathbf{K}_1\mathbf{x}) := \begin{bmatrix} \mathbf{f}_2(\pi_{1,3}(\mathbf{x})) \\ \mathbf{f}_2(\pi_{2,3}(\mathbf{x})) \\ \mathbf{f}_2(\pi_{3,3}(\mathbf{x})) \end{bmatrix}$$

We have slightly more general form that the oIP problem introduced before. This motivates to introduce the following parametrized problem:
oIP(k)
Input: $\mathbf{g} \in \mathbb{F}_2[x_1, \ldots, x_{kn}]^{kn}$ be quadratic polynomials.
Question: Find $(\mathbf{S}, \mathbf{U}) \in \mathrm{GL}_{kn}(\mathbb{F}_2) \times \mathrm{GL}_{kn}(\mathbb{F}_2)$ such that:

$$\begin{aligned} \pi_{1,k}(\mathbf{U} \cdot \mathbf{g}(\mathbf{S} \cdot \mathbf{x})) \in \mathbb{F}_2[x_1, \ldots, x_n]^n, \\ \pi_{2,k}(\mathbf{U} \cdot \mathbf{g}(\mathbf{S} \cdot \mathbf{x})) \in \mathbb{F}_2[x_{n+1}, \ldots, x_{2n}]^n, \\ \vdots \\ \pi_{k,k}(\mathbf{U} \cdot \mathbf{g}(\mathbf{S} \cdot \mathbf{x})) \in \mathbb{F}_2[x_{(k-1)n+1}, \ldots, x_{kn}]^n \end{aligned}$$

Following the same approach than Proposition 5.0.3, we have:

Proposition 5.0.4. *Let $\mathbf{g} \in \mathbb{F}_2[x_1, \ldots, x_{3n}]^{3n}$ and $(\mathbf{S}, \mathbf{U}) \in \mathrm{GL}_{3n}(\mathbb{F}_2) \times \mathrm{GL}_{3n}(\mathbb{F}_2)$ be a solution of* oIP *on $\mathbf{g}$. Then, the components of $\mathbf{S}, \mathbf{U}, \mathbf{S}^{-1}$ and $\mathbf{U}^{-1}$ vanish a system of, at most, $4n^2 + 3\,n \left(\sum_{i=0}^{2} \binom{3n}{i} - \binom{n}{i}\right) \in O(n^3)$ cubic equations in $8n^2 \in O(n^2)$ variables.*

6 Benchmark

We implemented MQFHE in C++ and used Emscripten to port it to JavaScript to run tests in the browser. Our current implementation is single threaded and has only had minor optimizations (no manual vectorization) + compiling with gcc -O3. As our scheme has relatively low hardware requirements we were able to run benchmarks on a standard 15" MacBook Pro Retina.

- Processor: 2.5 GHz Intel Core i7 with 6MB shared L3 cache
- RAM: 16 GB of 1600 MHz DDR3 onboard memory

We used the steady clock from std::chrono and averaged runtimes over 100 runs with random initializations with $n = 64, 128$ and $\ell = 2$. Some operations ran more expensively than expected (e.g. LMM) and will be revisited with new highly optimized benchmarking versions.

Operation	C++	Emscripten JavaScript Port
Generate private and bridge key (64-bit):	0.00866544s	0.1876s
Generate public key (64-bit):	0.153361s	0.86047s
Compute encrypted LMM (64-bit):	0.00515336s	0.048094s
Compute encrypted XOR (64-bit):	0.00003927s	0.000349s
Compute encrypted AND (64-bit):	0.00010069s	0.000815s
Generate private and bridge key (128-bit):	0.0570461s	1.2367s
Generate public key (128-bit):	1.28326s	7.9566s
Compute encrypted LMM (128-bit):	0.0000372425s	0.36909s
Compute encrypted XOR (128-bit):	0.000120598s	0.00207s
Compute encrypted AND (128-bit):	0.000194966s	0.00404s
Compute encrypted ADD (128-bit):	0.021877929s	n/a
Compute encrypted MULT (128-bit):	1.409277131s	n/a

Table 1. Average FHE operation runtime

The sizes of the keys are:

Key	Size (64-bit)	Size (128-bit)
PrivateKey	$<$ 50KB	$<$ 330KB
BridgeKey	$<$ 70KB	$<$ 412KB
PublicKey	$<$ 2.4MB	$<$ 18.5MB

Table 2. Key sizes

References

1. 2009.
2. Martin R. Albrecht, Pooya Farshim, Jean-Charles Faugère, and Ludovic Perret. Polly cracker, revisited. pages 179–196.
3. Martin R. Albrecht, Jean-Charles Faugère, Pooya Farshim, Gottfried Herold, and Ludovic Perret. Polly cracker, revisited. *Des. Codes Cryptography*, 79(2):261–302, 2016.
4. Frederik Armknecht, Daniel Augot, Ludovic Perret, and Ahmad-Reza Sadeghi. On constructing homomorphic encryption schemes from coding theory. pages 23–40.

5. Frederik Armknecht, Colin Boyd, Christopher Carr, Kristian Gjøsteen, Angela Jäschke, Christian A. Reuter, and Martin Strand. A guide to fully homomorphic encryption. Cryptology ePrint Archive, Report 2015/1192, 2015. http://eprint.iacr.org/.
6. Boo Barkee, Deh Cac Can, Julia Ecks, Theo Moriarty, and R. F. Ree. Why you cannot even hope to use Gröbner bases in Public Key Cryptography: An open letter to a scientist who failed and a challenge to those who have not yet failed. *Journal of Symbolic Computations*, 18(6):497–501, 1994.
7. Côme Berbain, Henri Gilbert, and Jacques Patarin. QUAD: A practical stream cipher with provable security. pages 109–128.
8. Côme Berbain, Henri Gilbert, and Jacques Patarin. QUAD: A multivariate stream cipher with provable security. *J. Symb. Comput.*, 44(12):1703–1723, 2009.
9. Jérémy Berthomieu, Jean-Charles Faugère, and Ludovic Perret. Polynomial-time algorithms for quadratic isomorphism of polynomials: The regular case. *J. Complexity*, 31(4):590–616, 2015.
10. Olivier Billet, Matthew J. B. Robshaw, and Thomas Peyrin. On building hash functions from multivariate quadratic equations. pages 82–95.
11. Alex Biryukov, Charles Bouillaguet, and Dmitry Khovratovich. Cryptographic schemes based on the ASASA structure: Black-box, white-box, and public-key (extended abstract). pages 63–84.
12. Andrej Bogdanov and Chin Ho Lee. Homomorphic encryption from codes. Cryptology ePrint Archive, Report 2011/622, 2011. http://eprint.iacr.org/2011/622.
13. Charles Bouillaguet, Jean-Charles Faugère, Pierre-Alain Fouque, and Ludovic Perret. Practical cryptanalysis of the identification scheme based on the isomorphism of polynomial with one secret problem. pages 473–493.
14. Charles Bouillaguet, Pierre-Alain Fouque, and Amandine Véber. Graph-theoretic algorithms for the "isomorphism of polynomials" problem. pages 211–227.
15. Zvika Brakerski. When homomorphism becomes a liability. pages 143–161.
16. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 309–325, New York, NY, USA, 2012. ACM.
17. Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. pages 97–106.
18. Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. pages 505–524.
19. Jung Hee Cheon and Damien Stehlé. Fully homomophic encryption over the integers revisited. pages 513–536.
20. Jean-Sébastien Coron, David Naccache, and Mehdi Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. pages 446–464.
21. Nicolas T. Courtois, Louis Goubin, and Jacques Patarin. SFLASHv3, a fast asymmetric signature scheme. Cryptology ePrint Archive, Report 2003/211, 2003. http://eprint.iacr.org/2003/211.
22. M. Dickerson. *The functional Decomposition of Polynomials*. PhD thesis, Cornell University, July 1989. TR 89-1023.
23. M. Dickerson. General polynomial decomposition and the s-1-decomposition are np-hard. *International Journal of Foundations of Computer Science*, pages 147–156, 1993.
24. Jintai Ding and Dieter Schmidt. Rainbow, a new multivariable polynomial signature scheme. In John Ioannidis, Angelos D. Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security, Third International Conference, ACNS 2005, New York, NY, USA, June 7-10, 2005, Proceedings*, volume 3531 of *Lecture Notes in Computer Science*, pages 164–175, 2005.
25. Itai Dinur, Orr Dunkelman, Thorsten Kranz, and Gregor Leander. Decomposing the ASASA block cipher construction. Cryptology ePrint Archive, Report 2015/507, 2015. http://eprint.iacr.org/2015/507.
26. Franoise Levy dit Vehel, Maria Grazia Marinari, Ludovic Perret, and Carlo Traverso. A survey on Polly Cracker systems. In Massimiliano Sala, Teo Mora, Ludovic Perret, Shojiro Sakata, and Carlo Traverso, editors, *Gröbner Bases. Coding and Cryptography*, pages 285–305. Springer Verlag, Berlin, Heidelberg, New York, 2009.
27. Léo Ducas and Daniele Micciancio. FHEW: Bootstrapping homomorphic encryption in less than a second. pages 617–640.
28. Jean-Charles Faugère and Ludovic Perret. Cryptanalysis of $2R^-$ schemes. pages 357–372.
29. Jean-Charles Faugère and Ludovic Perret. Polynomial equivalence problems: Algorithmic and theoretical aspects. pages 30–47.
30. Jean-Charles Faugère and Ludovic Perret. Polynomial equivalence problems: Algorithmic and theoretical aspects. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 30–47. Springer, 2006.

31. Jean-Charles Faugère and Ludovic Perret. High order derivatives and decomposition of multivariate polynomials. In Jeremy R. Johnson, Hyungju Park, and Erich Kaltofen, editors, *Symbolic and Algebraic Computation, International Symposium, ISSAC 2009, Seoul, Republic of Korea, July 29-31, 2009, Proceedings*, pages 207–214. ACM, 2009.
32. Mike Fellows and Neal Koblitz. Combinatorial cryptosystems galore! In G. L. Mullen and P. J.-S. Shiue, editors, *Finite Fields: Theory, Applications, and Algorithms*, volume 168 of *Contemporary Mathematics*, pages 51–61. AMS, 1994.
33. Craig Gentry. Toward basing fully homomorphic encryption on worst-case hardness. pages 116–137.
34. Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. crypto.stanford.edu/craig.
35. Craig Gentry, Shai Halevi, and NigelP. Smart. Homomorphic evaluation of the aes circuit. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 850–867. Springer Berlin Heidelberg, 2012.
36. Henri Gilbert, Jérôme Plût, and Joana Treger. Key-recovery attack on the ASASA cryptosystem with expanding S-boxes. pages 475–490.
37. Kristian Gjøsteen and Martin Strand. Fully homomorphic encryption must be fat or ugly? Cryptology ePrint Archive, Report 2016/105, 2016. http://eprint.iacr.org/2016/105.
38. Oded Goldreich. *The Foundations of Cryptography - Volume 2, Basic Applications*. Cambridge University Press, 2004.
39. Domingo Gómez-Pérez, Jaime Gutierrez, and Alina Ostafe. Common composites of triangular polynomial systems and hash functions. *J. Symb. Comput.*, 72:182–195, 2016.
40. Aline Gouget and Jacques Patarin. Probabilistic multivariate cryptography. pages 1–18.
41. Shai Halevi and Victor Shoup. *Advances in Cryptology – EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, chapter Bootstrapping for HElib, pages 641–670. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.
42. Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced oil and vinegar signature schemes. pages 206–222.
43. Tsutomu Matsumoto and Hideki Imai. Public quadratic polynominal-tuples for efficient signature-verification and message-encryption. pages 419–453.
44. Brice Minaud, Patrick Derbez, Pierre-Alain Fouque, and Pierre Karpman. Key-recovery attacks on ASASA. pages 3–27.
45. Valerie Nachef, Jacques Patarin, and Emmanuel Volte. Zero-knowledge for multivariate polynomials. Cryptology ePrint Archive, Report 2012/239, 2012. http://eprint.iacr.org/2012/239.
46. Kim Laine Kristin Lauter Michael Naehrig John Wernsing Nathan Dowlin, Ran Gilad-Bachrach. Manual for using homomorphic encryption for bioinformatics. Technical report, November 2015.
47. Alina Ostafe and Igor E. Shparlinski. Pseudorandom numbers and hash functions from iterations of multivariate polynomials. *Cryptography and Communications*, 2(1):49–67, 2010.
48. Jacques Patarin. Cryptoanalysis of the Matsumoto and Imai public key scheme of eurocrypt'88. pages 248–261.
49. Jacques Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. pages 33–48.
50. Jacques Patarin. Hidden fields equations (hfe) and isomorphisms of polynomials (ip): Two new families of asymmetric algorithms. In *Proceedings of the 15th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'96, pages 33–48, Berlin, Heidelberg, 1996. Springer-Verlag.
51. Jacques Patarin, Nicolas Courtois, and Louis Goubin. QUARTZ, 128–bit long digital signatures. pages 282–297.
52. Jacques Patarin and Louis Goubin. Asymmetric cryptography with s-boxes. In Yongfei Han, Tatsuaki Okamoto, and Sihan Qing, editors, *Information and Communication Security, First International Conference, ICICS'97, Beijing, China, November 11-14, 1997, Proceedings*, volume 1334 of *Lecture Notes in Computer Science*, pages 369–380. Springer, 1997.
53. Jacques Patarin, Louis Goubin, and Nicolas Courtois. Improved algorithms for isomorphisms of polynomials. pages 184–200.
54. L. Perret. On the computational complexity of some equivalence problems of polynomial systems of equations over finite fields. *Electronic Colloquium on Computational Complexity (ECCC)*, 116, 2004.
55. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. pages 84–93.
56. Koichi Sakumoto. Public-key identification schemes based on multivariate cubic polynomials. pages 172–189.
57. Koichi Sakumoto, Taizo Shirai, and Harunaga Hiwatari. Public-key identification schemes based on multivariate quadratic polynomials. pages 706–723.
58. Nigel P. Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. pages 420–443.

59. Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. pages 24–43.
60. Christopher Wolf and Bart Preneel. Large superfluous keys in multivariate quadratic asymmetric systems. In *Public Key Cryptography – PKC 2005*, volume 3386 of *LNCS*, pages 275–287. Springer, 2005.
61. Christopher Wolf and Bart Preneel. Equivalent keys in multivariate quadratic public key systems. *Journal of Mathematical Cryptology*, 4(4):375–415, 2011.
62. Ding-Feng Ye, Zongduo Dai, and Kwok-Yan Lam. Decomposing attacks on asymmetric cryptography based on mapping compositions. 14(2):137–150, 2001.