Designing Efficient Dyadic Operations for Cryptographic Applications

Gustavo Banegas[1], Paulo S. L. M. Barreto[2], Edoardo Persichetti[3] and Paolo Santini[4]

August 19, 2018

[1] Technische Universiteit Eindhoven, Netherlands

[2] University of Washington at Tacoma, USA

[3] Florida Atlantic University, USA

[4] Università Politecnica delle Marche, Italy

Post-Quantum Cryptography

NIST proposals

November 2017: NIST posts 82 submissions from 260 people.

	Signatures	KEM/Encryption	Overall
Lattice-based	4	24	28
Code-based	5	19	24
MQ-based	7	6	13
Hash-based	4		4
Other	3	10	13
Total	**23**	**59**	**82**

DAGS

DAGS: Key Encapsulation from Dyadic GS Codes

- It is a code-based KEM;
- It uses Generalized Srivastava codes;
- It has short keys

DAGS

DAGS: Key Encapsulation from Dyadic GS Codes

- It is a code-based KEM;
- It uses Generalized Srivastava codes;
- It has short keys, **much smaller than Classic McEliece;**
- As the name suggest it uses dyadic operations.

Introduction

What did we do?

- Improve code-based cryptographic schemes that use Quasi-Dyadic (QD) operations;
- Analyze the multiplication of dyadic matrices using: "Standard", Karatsuba and Fast Walsh-Hadamard Transformation;
- Apply LUP decomposition to dyadic case.

Preliminaries & Notations

What are dyadic matrices?

Given a ring $\mathcal{R}$ and a vector $h = (h_0, h_1, \dots h_{n-1}) \in \mathcal{R}$ with $n = 2^r, r \in \mathbb{N}$, called the order.

A dyadic matrix is the symmetric matrix with components $\mathbf{\Delta}_{ij} = h_{i \oplus j}$, where $\oplus$ stands for bitwise exclusive-or.

We use $\mathbf{\Delta}(h)$ to denote dyadic matrix.

The product of two dyadic matrices is a dyadic matrix.

Quasi-dyadic matrix

A quasi-dyadic matrix is a block matrix whose blocks are dyadic.

Preliminaries & Notations

What are dyadic matrices?

Given a ring $\mathcal{R}$ and a vector $h = (h_0, h_1, \dots h_{n-1}) \in \mathcal{R}$ with $n = 2^r, r \in \mathbb{N}$, called the order.

A dyadic matrix is the symmetric matrix with components $\mathbf{\Delta}_{ij} = h_{i \oplus j}$, where $\oplus$ stands for bitwise exclusive-or.

We use $\mathbf{\Delta}(h)$ to denote dyadic matrix.

The product of two dyadic matrices is a dyadic matrix.

Quasi-dyadic matrix

A quasi-dyadic matrix is a block matrix whose blocks are dyadic.

In particular, we focus on the special case of quasi-dyadic matrices with elements belonging to a field $\mathbb{F}$ of characteristic 2.

Preliminaries & Notations

A dyadic permutation

A *dyadic permutation* is a dyadic matrix $\mathbf{\Pi}^i \in \mathbf{\Delta}(\{0,1\}^n)$ given by $\mathbf{\Pi}^i = \mathbf{\Delta}(\pi^i)$ where π^i is the i-th unit vector.

Example

Suppose $n = 4$, and $i = 1$. So, we have $\pi^1 = (0,1,0,0)$ and $\mathbf{\Pi}^1$ is equal to:

$$\mathbf{\Pi}^1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Standard Multiplication

The element of a matrix **C** in position (i, j) is obtained as the multiplication between the i-th row of **A** and the j-th column of **B**.

Standard Multiplication

The element of a matrix $\mathbf{C}$ in position (i, j) is obtained as the multiplication between the i-th row of $\mathbf{A}$ and the j-th column of $\mathbf{B}$.
However, A and B are **dyadic matrices** and they are **symmetric**.
So, the product is equivalent to the inner product between i-th row of A and the j-th of B.

The schoolbook matrix multiplication takes 2^{3r} multiplications.
Because $\boldsymbol{\Delta(a)\Delta(b)}$ is dyadic we only need the top row.

Standard Multiplication

input : $r \in \mathbb{N}$, $n = 2^r$ and $a, b \in \mathbb{F}^n$
output: $c \in \mathbb{F}^n$ such that $\boldsymbol{\Delta}(c) = \boldsymbol{\Delta}(a)\boldsymbol{\Delta}(b)$

```
c ← (0, 0, ..., 0);
c_0 ← a_0 b_0;
for i ← 1 to n − 1 do
    c_0 ← c_0 + a_i b_i;
    i^(b) ← binary representation of i;
    for j ← 0 to n − 1 do
        j^(b) ← binary representation of j;
        π^(b) ← i^(b) ⊕ j^(b);
        π ← integer representation of π^(b);
        c_i ← c_i + a_i b_π;
    end
end
return c;
```

Standard Multiplication

input : $r \in \mathbb{N}$, $n = 2^r$ and $a, b \in \mathbb{F}^n$
output: $c \in \mathbb{F}^n$ such that $\boldsymbol{\Delta}(c) = \boldsymbol{\Delta}(a)\boldsymbol{\Delta}(b)$

```
c ← (0, 0, ..., 0);
c_0 ← a_0 b_0;
for i ← 1 to n − 1 do
    c_0 ← c_0 + a_i b_i;
    i^(b) ← binary representation of i;
    for j ← 0 to n − 1 do
        j^(b) ← binary representation of j;
        π^(b) ← i^(b) ⊕ j^(b);
        π ← integer representation of π^(b);
        c_i ← c_i + a_i b_π;
    end
end
return c;
```

Complexity estimated in:

$$C_{std} = r(2^{2r} - 2^r) + 2^{2r} C_{mul} + (2^{2r} - 2^r) C_{sum}$$

Dyadic Convolution

What is dyadic convolution?

The dyadic convolution of two vectors $a, b \in \mathbb{F}$, denoted by $a \vartriangle b$, is the unique vector of $\mathbb{F}$ such that $\boldsymbol{\Delta}(a \vartriangle b) = \boldsymbol{\Delta}(a)\boldsymbol{\Delta}(b)$.

Sylvester-Hadamard Matrices

$$
\begin{aligned}
\mathbf{H}_0 &= \left[\ 1\ \right], \\
\mathbf{H}_r &= \begin{bmatrix} \mathbf{H}_{r-1} & \mathbf{H}_{r-1} \\ \mathbf{H}_{r-1} & -\mathbf{H}_{r-1} \end{bmatrix}, r > 0.
\end{aligned}
$$

Dyadic Convolution

What do we achieve?

Computing $\mathbf{c}$ such that $\boldsymbol{\Delta}(\mathbf{a})\boldsymbol{\Delta}(\mathbf{b}) = \boldsymbol{\Delta}(\mathbf{c})$ involves only three multiplications of vectors by Sylvester-Hadamard matrices.

Dyadic Convolution

What do we achieve?

Computing $\mathbf{c}$ such that $\mathbf{\Delta}(\mathbf{a})\mathbf{\Delta}(\mathbf{b}) = \mathbf{\Delta}(\mathbf{c})$ involves only three multiplications of vectors by Sylvester-Hadamard matrices.

For this we propose two algorithms. First, we need to compute $\mathbf{a}\mathbf{H}_r$ where $\mathbf{a}$ is a vector and $\mathbf{H}_r$ a Sylvester-Hadamard matrix. Second, we perform the multiplication

Dyadic Convolution

```
input  : r ∈ ℕ, n = 2^r and a ∈ 𝔽^n
output: aH_r
v ← 1;
for j ← 1 to n do
    w ← v;
    v ← (v << 1);
    /* left shift by one position                    */
    for i ← 0 to n − 1 by v do
        for l ← 0 to w do
            s ← a_{i+l};
            q ← a_{i+l+w};
            a_{i+l} ← s + q;
            a_{i+l+w} ← s − q;
        end
    end
end
return a;
```

Algorithm 1: The fast Walsh-Hadamard transform (FWHT)

Dyadic Convolution

input : $r \in \mathbb{N}$, $n = 2^r$ and $\mathbf{a}, \mathbf{b} \in \mathbb{F}^n$
output: $\mathbf{a} \triangle \mathbf{b} \in \mathbb{F}^n$ such that $\mathbf{\Delta}(\mathbf{a})\mathbf{\Delta}(\mathbf{b}) = \mathbf{\Delta}(\mathbf{a} \triangle \mathbf{b})$
$c \leftarrow (0, 0, \ldots, 0)$;
$\tilde{c} \leftarrow (0, 0, \ldots, 0)$;
Compute $\tilde{a} \leftarrow \mathbf{aH}_r$ via previous algorithm;
Compute $\tilde{b} \leftarrow \mathbf{bH}_r$ via previous algorithm;
for $j \leftarrow 0$ **to** $n - 1$ **do**
    $\tilde{c} \leftarrow \tilde{a}_j \tilde{b}_j$;
end
Compute $c \leftarrow \tilde{c}\mathbf{H}_r$ via previous algorithm;
$c \leftarrow (c >> r)$;
`/* right shift by` r `positions */`
return c;

Algorithm 2: Dyadic convolution via the FWHT

Karatsuba

Consider a vector $\mathbf{a}$ and its halves defined as:

$$\mathbf{a}_0 = \left[a_0, a_1, \cdots, a_{\frac{n}{2}-1}\right]$$
$$\mathbf{a}_1 = \left[a_{\frac{n}{2}}, a_{\frac{n}{2}+1}, \cdots, a_{n-1}\right].$$

Some straightforward computations show that the following relations hold:

$$\mathbf{c}_0 = \mathbf{a}_0\mathbf{b}_0 + \mathbf{a}_1\mathbf{b}_1$$
$$\mathbf{c}_1 = (\mathbf{a}_0 + \mathbf{a}_1)(\mathbf{b}_0 + \mathbf{b}_1) + \mathbf{c}_0$$

We can summarize the complexity of this method as:

$$C_{\mathtt{Kar}} = 3^r \cdot C_{\mathtt{mul}} + 4 \cdot [3^r - 2^r] \cdot C_{\mathtt{sum}}$$

Dyadic Matrices Inverse

Inverse of dyadic matrices can be defined as:

The inverse $\boldsymbol{\Delta}(\mathbf{a})^{-1}$ is a dyadic matrix $\boldsymbol{\Delta}(\mathbf{b})$. We can compute $\mathbf{b}$ as follows:

1. Compute $\widetilde{\mathbf{b}}$ with $\mathrm{diag}(\widetilde{\mathbf{b}}) = [\mathrm{diag}\,(\mathbf{a}\mathbf{H}_r)]^{-1}$;
2. Compute $\mathbf{b}' = \widetilde{\mathbf{b}}\mathbf{H}_r$;
3. For each entry in $\mathbf{b}'$ shift right r positions, the result is $\mathbf{b}$.

DAGS

Improving DAGS

Table: Cost of Multiplication between Dyadic Matrices

		Standard	Karatsuba	Dyadic Convolution
$\mathbb{F}_{2^5}$	$r = 4$	$4,833$	$2,194$	$3,899$
	$r = 5$	$21,285$	$5,909$	$12,045$
$\mathbb{F}_{2^6}$	$r = 4$	$5,833$	$2,194$	$4,899$
	$r = 5$	$23,231$	$6,223$	$13,568$

DAGS

Improving DAGS

Table: Comparison of Inversion Methods

	Original DAGS	LUP Inversion	LUP + Karatsuba
DAGS 1	$1,318,973,209$	$321,771$	$108,117$
DAGS 3	$2,211,076,311$	$557,822$	$198,199$
DAGS 5	$17,925,330,712$	$654,713$	$431,890$

Questions

Thank you for your attention.
gustavo@cryptme.in