Matrix-vector product of a new class of quasi-involutory MDS matrices

P. Freyre; E.D. Fiallo; R. Rodríguez

Introduction

- Maximum Distance Separable (MDS) matrices theoretically ensures a perfect diffusion.
- They have great importance in the design of block ciphers and hash functions.

MDS matrices are in general:
not sparse, have a large description $\Rightarrow$ costly implementations

Introduction

To reduce implementation costs:

- circulant matrices.

  Gupta, K. C., Pandey, S. K., Venkateswarlu, A. *On the direct construction of recursive MDS matrices.* Designs, Codes and Cryptography, 2017.

- recursive matrices

  Gupta, K.C., Pandey, S.K., Samanta, S. *Construction of Recursive MDS Matrices Using DLS Matrices.* AFRICACRYPT, 2022.

- methods for transforming an MDS matrix into other ones

  Luong, T. T., Cuong, N. N., *Direct exponent and scalar multiplication transformations of mds matrices: some good cryptographic results for dynamic diffusion layers of block ciphers.* Journal of Computer Science and Cybernetics, 2016.

Introduction

Our interest:

- Diffusion layer **as** MDS matrix-vector product.
- MDS matrix-vector product **based** *on the multiplication of two polynomials modulo a generating polynomial of the cyclic code.*

Arrozarena, P. F., Fiallo, E. D. *Efficient multiplication of a vector by a matrix MDS.* Journal of Science and Technology on Information security, 2022.

no need to store the MDS matrix explicitly

Can be applied to involutory MDS matrices?

Introduction

- Involutory MDS matrices have the main advantage that both encryption and decryption share the same matrix-vector product.
- Finding involutory MDS matrices, in particular **large (involutory) MDS matrices**, is not an easy.

Quasi-involutory MDS matrix?

- Intuitive idea of a **MDS matrix** that is **close to being involutory**.

Our contribution

A new class of quasi-involutive MDS matrices is proposed.

- matrix-vector product through multiplication of two polynomials modulo a generating polynomial of a:
  1. Reed-Solomon (RS) codes
  2. CGMN code

     Couselo, E., Gonzalez, S., Markov, V., Nechaev, A. *Parameters of recursive MDS-codes.* Diskretnaya Matematika, 2000.

- If $p \neq 2$ in $\mathbb{F}_{p^n} \Rightarrow$ the MDS matrix is involutive.
- If $p = 2$ in $\mathbb{F}_{p^n} \Rightarrow$ the MDS matrix is quasi-involutive:
  1. the vector is transformed one step through an LFSR.
  2. the multiplication by the inverse matrix can be performed with the original MDS matrix.

Preliminaries

A *linear code* $\mathcal{C}$ of length n and dimension k over $\mathbb{F}_q$, denoted as $[n, k]_q$, is a linear subspace of dimension k of the linear space $\mathbb{F}_q^n$.

The *minimum distance* d of $\mathcal{C}$ is the minimum weight if its nonzero vectors and we denote the code as $[n, k, d]_q$.

A *generator matrix* for $\mathcal{C}$ is a matrix whose rows form a basis for $\mathcal{C}$ and it is said to be in *standard form* if it has the form $(I_k|R)$ where I_k is a $k \times k$ identity matrix and R is a $k \times (n-k)$ matrix.

Preliminaries

A linear code such that $d = n - k + 1$ (Singleton Bound) is called a *Maximum Distance Separable (MDS) code*.

A *matrix is MDS* if and only if all its minors are non zero.

Cyclic codes

An $[n, k]_q$ code is said to be cyclic if a cyclic shift of any element of the code remains in the code.

$$(c_0, c_1, \ldots, c_{n-1}) \in \mathcal{C} \Rightarrow (c_{n-1}, c_0, \ldots, c_{n-2}) \in \mathcal{C}$$

Can be seen as ideals of $\mathbb{F}_q[x]/(x^n - 1)$ with every monic polynomial $g(x)$ that divides $x^n - 1$ as *generating polynomial*.

The order e of $g(x)$ is the smallest positive integer such that $g(x)$ divides $x^e - 1$ with e divide n.

If $deg(g) = r \Rightarrow$ the code defined by $g(x)$ has dimension $k = n - r$.

Cyclic codes

The generator matrix, in standard form, can be given by $(I_k| - R)$ with

$$R = \begin{pmatrix} x^{n-k} \ mod \ g(x) \\ x^{n-k+1} \ mod \ g(x) \\ \vdots \\ x^{n-1} \ mod \ g(x) \end{pmatrix} \tag{1}$$

Reed–Solomon (RS) codes

A q-ary RS code over $\mathbb{F}_q$ of length $q-1,\ q>2$, is the cyclic code generated by a polynomial of the form

$$g(x) = (x-\alpha^{a+1})(x-\alpha^{a+2})\cdots(x-\alpha^{a+\delta-1})$$

with $a \geq 0$ and $2 \leq \delta \leq q-1$, where α is a primitive element of $\mathbb{F}_q$.

It is an MDS code with parameters $[q-1, q-\delta, \delta]_q$ and the matrix R is a MDS matrix.

Since α is a primitive element, the order of $g(x)$ is $q-1$.

CGMN code

The code is composed of segments of length n of the linear recurring sequences that have characteristic polynomial

$$g(x) = (x - \beta_0) \cdots (x - \beta_{m-1})$$

that is, for $i = 0, 1, \ldots, n - m + 1$ the code has the form

$$\mathcal{K} = \{(u(0), \ldots, u(n)) : u(i+m) = g_0 u(i) + \cdots + g_{n-1} u(i+m-1)\}$$

where $g_0, \ldots, g_{m-1}$ are the coefficients of $g(x)$.

For certain $\beta_0, \ldots, \beta_{m-1}$, it is an MDS code with parameters $[q+1, m, q-m+2]$.

CGMN code

If q is even or m is odd, then the code is cyclic and the order of the polynomial $g(x)$ is conditioned by the order of the elements $\beta_0, \ldots, \beta_{m-1}$.

It is shown that

$$ord(\beta_i) | q+1, \; 0 \leq i \leq m-1$$

and $\beta_i \neq \beta_j, \; i \neq j, \; 0 \leq i, j \leq m-1$.

Then, if $q+1$ is prime, the code is cyclic and the order of $g(x)$ is $q+1$.

It is possible to operate by multiplying polynomials

Quasi-involutive linear transformation

Let $n \in \mathbb{N}$. The biyective linear transformation

$$\Psi : P[x]/g(x) \to P[x]/g(x)$$

defined by

$$\forall p(x) \in P[x]/g(x) : \ \Psi(p(x)) = p(x) \cdot x^n \ mod \ g(x)$$

is quasi-involutive if its inverse Ψ^{-1} is

$$\Psi^{-1}(p(x)) = \Psi(p(x)) \cdot x \ mod \ g(x)$$

Matrix-vector product

Arrozarena, P. F., Fiallo, E. D. *Efficient multiplication of a vector by a matrix MDS.* Journal of Science and Technology on Information security, 2022.

To multiply a vector by any square MDS submatrix of matrix

$$R = \begin{pmatrix} x^{n-k} \ mod \ g(x) \\ x^{n-k+1} \ mod \ g(x) \\ \vdots \\ x^{n-1} \ mod \ g(x) \end{pmatrix}$$

it can be done by multiplying the polynomial that represents the vector by the polynomial corresponding to the first row of the selected submatrix.

Algorithm 1: Generation of involutory and quasi-involutory MDS matrix.

Input :

- The RS or CGMN generating polynomial $g(x) \in \mathbb{F}_q[x]$ of degree $n-k$.
- The canonical polynomials x^i, $0 \leq i \leq n-1$.

Output: involutory or quasi-involutory $n \times n$ MDS matrix M.

Data : $q = p^t$, p prime and $t \in \mathbb{N}$.

if $p > 2$ **then**
    if $g(x)$ *is RS* **then**
        $f(x) \leftarrow \left(-x^{\frac{q-1}{2}} \mod g(x)\right)$;
    if $g(x)$ *is CGMN* **then**
        $f(x) \leftarrow \left(-x^{\frac{q+1}{2}} \mod g(x)\right)$;
    for $i = 1$ **to** n **do**
        $M_i \leftarrow x^{i-1} \cdot f(x) \mod g(x)$;

if $p = 2$ **then**
    if $g(x)$ *is RS* **then**
        $f(x) \leftarrow x^{2^{t-1}-1} \mod g(x)$;
    if $g(x)$ *is CGMN* **then**
        $f(x) \leftarrow x^{2^{t-1}} \mod g(x)$;
    for $i = 1$ **to** n **do**
        $M_i \leftarrow x^{i-1} \cdot f(x) \mod g(x)$;

return M;

Algorithm 2: Generation of involutory and quasi-involutory MDS inverse matrix.

Input :

- The RS or CGMN generating polynomial $g(x) \in \mathbb{F}_q[x]$ of degree $n - k$.
- The canonical polynomials x^i, $0 \leq i \leq n - 1$.

Output: involutory or quasi-involutory $n \times n$ MDS inverse matrix M^{-1}.

Data : $q = p^t$, p prime and $t \in \mathbb{N}$.

if $p > 2$ **then**
  if $g(x)$ *is RS* **then**
    $f(x) \leftarrow \left(-x^{\frac{q-1}{2}} \mod g(x)\right)$;
  if $g(x)$ *is CGMN* **then**
    $f(x) \leftarrow \left(-x^{\frac{q+1}{2}} \mod g(x)\right)$;
  for $i = 1$ **to** n **do**
    $M_i \leftarrow x^i \cdot f(x) \mod g(x)$;
if $p = 2$ **then**
  if $g(x)$ *is RS* **then**
    $f(x) \leftarrow x^{2^{t-1}} \mod g(x)$;
  if $g(x)$ *is CGMN* **then**
    $f(x) \leftarrow x^{2^{t-1}+1} \mod g(x)$;
  for $i = 1$ **to** n **do**
    $M_i \leftarrow x^{i-1} \cdot f(x) \mod g(x)$;
return M^{-1};

Algorithm 3: Multiplication of a vector by involutory or quasi-involutory MDS matrix

Input :

- The RS or CGMN generating polynomial $g(x) \in \mathbb{F}_q[x]$ of degree $n-k$.
- The vector of coefficients $a = (a_0, a_1, \ldots, a_{n-1})$.

Output: The vector $\hat{a} = a \cdot M$.
Data : $q = p^t$, p prime and $t \in \mathbb{N}$.

if $p > 2$ **then**
  if $g(x)$ *is RS* **then**
    $f(x) \leftarrow \left(-x^{\frac{q-1}{2}} \mod g(x)\right)$;
  if $g(x)$ *is CGMN* **then**
    $f(x) \leftarrow \left(-x^{\frac{q+1}{2}} \mod g(x)\right)$;
  $\hat{a}(x) \leftarrow a(x) \cdot f(x) \mod g(x)$;
if $p = 2$ **then**
  if $g(x)$ *is RS* **then**
    $f(x) \leftarrow x^{2^{t-1}-1} \mod g(x)$;
  if $g(x)$ *is CGMN* **then**
    $f(x) \leftarrow x^{2^{t-1}} \mod g(x)$;
  $\hat{a}(x) \leftarrow a(x) \cdot f(x) \mod g(x)$;
return $\hat{a}$ `//coefficients of` $\hat{a}(x)$

Algorithm 4: Multiplication of a vector by the inverse of involutory or quasi-involutory MDS matrix

Input :

- The RS or CGMN generating polynomial $g(x) \in \mathbb{F}_q[x]$ of degree $n - k$.
- The vector of coefficients $a = (a_0, a_1, \ldots, a_{n-1})$.

Output: The vector $\hat{a} = a \cdot M^{-1}$.
Data : $q = p^t$, p prime and $t \in \mathbb{N}$.

if $p > 2$ **then**
  if $g(x)$ *is RS* **then**
    $f(x) \leftarrow \left(-x^{\frac{q-1}{2}} \mod g(x)\right)$;
  if $g(x)$ *is CGMN* **then**
    $f(x) \leftarrow \left(-x^{\frac{q+1}{2}} \mod g(x)\right)$;
  $\hat{a}(x) \leftarrow a(x) \cdot f(x) \mod g(x)$;
if $p = 2$ **then**
  if $g(x)$ *is RS* **then**
    $f(x) \leftarrow x^{2^{t-1}-1} \mod g(x)$;
  if $g(x)$ *is CGMN* **then**
    $f(x) \leftarrow x^{2^{t-1}} \mod g(x)$;
  $a(x) \leftarrow a(x) \cdot x \mod g(x)$;
  $\hat{a}(x) \leftarrow a(x) \cdot f(x) \mod g(x)$;
return $\hat{a}$ `//coefficients of` $\hat{a}(x)$

Example of MDS matrix of size 8×8 in an RS code

Let's consider the finite field $\mathbb{F}_{2^8}$ with polynomial $x^8 + x^4 + x^3 + x^2 + 1$. We have then that $n = 2^8 - 1 = 255,\ \delta = 9,\ k = 2^8 - 9 = 247$. The generator polynomial is

$$g(x) = x^8 + \alpha^{176}x^7 + \alpha^{240}x^6 + \alpha^{211}x^5 + \alpha^{253}x^4 + \alpha^{220}x^3 + \alpha^3x^2 + \alpha^{203}x + \alpha^{36}$$

The matrix R is as follows

$$R = \begin{pmatrix} x^8 \ mod \ g(x) \\ x^9 \ mod \ g(x) \\ \vdots \\ x^{254} \ mod \ g(x) \end{pmatrix}$$

Example of MDS matrix of size 8×8 in an RS code

Applying algorithm 1, the obtained square MDS matrix is

$$M = \begin{pmatrix} x^{127} \ mod \ g(x) \\ x^{128} \ mod \ g(x) \\ \vdots \\ x^{134} \ mod \ g(x) \end{pmatrix}$$

$$M = \begin{pmatrix} \texttt{0x49} & \texttt{0xe4} & \texttt{0x8e} & \texttt{0xec} & \texttt{0x3a} & \texttt{0x15} & \texttt{0x1d} & \texttt{0xa4} \\ \texttt{0x6d} & \texttt{0xd0} & \texttt{0xdb} & \texttt{0xc0} & \texttt{0xf} & \texttt{0x12} & \texttt{0xea} & \texttt{0x72} \\ \texttt{0xc4} & \texttt{0xa8} & \texttt{0x95} & \texttt{0x3a} & \texttt{0x35} & \texttt{0xdf} & \texttt{0xe6} & \texttt{0x12} \\ \texttt{0x34} & \texttt{0x12} & \texttt{0x6c} & \texttt{0x9f} & \texttt{0x23} & \texttt{0x6b} & \texttt{0x5d} & \texttt{0x9e} \\ \texttt{0x43} & \texttt{0xf3} & \texttt{0xd1} & \texttt{0x7} & \texttt{0xd7} & \texttt{0xab} & \texttt{0x4f} & \texttt{0x93} \\ \texttt{0xe9} & \texttt{0x5d} & \texttt{0x2} & \texttt{0x64} & \texttt{0x92} & \texttt{0xb8} & \texttt{0x6f} & \texttt{0x60} \\ \texttt{0xff} & \texttt{0xf3} & \texttt{0xbd} & \texttt{0xbe} & \texttt{0x96} & \texttt{0x4d} & \texttt{0xc1} & \texttt{0x2c} \\ \texttt{0x3b} & \texttt{0x2b} & \texttt{0xb1} & \texttt{0x3d} & \texttt{0x1a} & \texttt{0x90} & \texttt{0x1f} & \texttt{0x8f} \end{pmatrix}$$

Example of MDS matrix of size 8×8 in an RS code

Let the vector
$a = (\alpha^7, \alpha^{123}, \alpha^{58}, \alpha^{91}, \alpha^{72}, \alpha^{45}, \alpha^{208}, \alpha^{237}) \in \mathbb{F}_{2^8}^8$.

To perform the operation $a \cdot M$ applying algorithm 3, the operation

$$a(x) \cdot \left(x^{2^{8-1}-1} \; mod \; g(x) \right) \; mod \; g(x)$$

must be performed, where

$$a(x) = \alpha^7 + \alpha^{123}x + \alpha^{58}x^2 + \alpha^{91}x^3 + \alpha^{72}x^4 + \alpha^{45}x^5 + \alpha^{208}x^6 + \alpha^{237}x^7$$

Example of MDS matrix of size 8×8 in an RS code

The result is the polynomial

$$\hat{a}(x) = \alpha^{209} + \alpha^{15}x + \alpha^{245}x^2 + \alpha^{90}x^3 + \alpha^{19}x^4 + \alpha^{157}x^5 + \alpha^{52}x^6 + \alpha^{11}x^7$$

which represents the vector
$\hat{a} = (\alpha^{209}, \alpha^{15}, \alpha^{245}, \alpha^{90}, \alpha^{19}, \alpha^{157}, \alpha^{52}, \alpha^{11})$.

It can be verified by means of the usual multiplication of a vector by a matrix that

$$\hat{a} = a \cdot M$$

Example of MDS matrix of size 8×8 in an RS code

Applying algorithm 2 is obtained M^{-1}

$$M^{-1} = \begin{pmatrix} \texttt{0xe4} & \texttt{0x8e} & \texttt{0xec} & \texttt{0x3a} & \texttt{0x15} & \texttt{0x1d} & \texttt{0xa4} & \texttt{0xb9} \\ \texttt{0xd0} & \texttt{0xdb} & \texttt{0xc0} & \texttt{0xf} & \texttt{0x12} & \texttt{0xea} & \texttt{0x72} & \texttt{0x34} \\ \texttt{0xa8} & \texttt{0x95} & \texttt{0x3a} & \texttt{0x35} & \texttt{0xdf} & \texttt{0xe6} & \texttt{0x12} & \texttt{0x7e} \\ \texttt{0x12} & \texttt{0x6c} & \texttt{0x9f} & \texttt{0x23} & \texttt{0x6b} & \texttt{0x5d} & \texttt{0x9e} & \texttt{0xe8} \\ \texttt{0xf3} & \texttt{0xd1} & \texttt{0x7} & \texttt{0xd7} & \texttt{0xab} & \texttt{0x4f} & \texttt{0x93} & \texttt{0x74} \\ \texttt{0x5d} & \texttt{0x2} & \texttt{0x64} & \texttt{0x92} & \texttt{0xb8} & \texttt{0x6f} & \texttt{0x60} & \texttt{0x78} \\ \texttt{0xf3} & \texttt{0xbd} & \texttt{0xbe} & \texttt{0x96} & \texttt{0x4d} & \texttt{0xc1} & \texttt{0x2c} & \texttt{0x5a} \\ \texttt{0x2b} & \texttt{0xb1} & \texttt{0x3d} & \texttt{0x1a} & \texttt{0x90} & \texttt{0x1f} & \texttt{0x8f} & \texttt{0x30} \end{pmatrix}$$

Example of MDS matrix of size 8×8 in an RS code

Let the vector
$\hat{a} = a \cdot M = (\alpha^{209}, \alpha^{15}, \alpha^{245}, \alpha^{90}, \alpha^{19}, \alpha^{157}, \alpha^{52}, \alpha^{11})$.

To perform the operation $\hat{a} \cdot M^{-1}$ applying algorithm 4, the operations

$$\hat{a}(x) \leftarrow \hat{a}(x) \cdot x \ mod \ g(x) \rightarrow \textbf{one step through an LFSR}$$

$$\hat{a}(x) \cdot \left(x^{2^7-1} \ mod \ g(x)\right) \ mod \ g(x)$$

must be performed, where $\hat{a}(x)$ is the polynomial

$$\hat{a}(x) = \alpha^{209} + \alpha^{15}x + \alpha^{245}x^2 + \alpha^{90}x^3 + \alpha^{19}x^4 + \alpha^{157}x^5 + \alpha^{52}x^6 + \alpha^{11}x^7$$

Example of MDS matrix of size 8×8 in an RS code

The result is, in effect, the polynomial

$$a(x) = \alpha^7 + \alpha^{123}x + \alpha^{58}x^2 + \alpha^{91}x^3 + \alpha^{72}x^4 + \alpha^{45}x^5 + \alpha^{208}x^6 + \alpha^{237}x^7$$

which represents the vector
$a = (\alpha^7, \alpha^{123}, \alpha^{58}, \alpha^{91}, \alpha^{72}, \alpha^{45}, \alpha^{208}, \alpha^{237})$.

Conclusions

- A new class of quasi-involutive MDS matrices has been defined.
- When the characteristic of the finite field is different from 2, the MDS matrix is involutive.
- When the characteristic is 2, the MDS matrix is quasi-involutive.
  - the inverse matrix-vector product is done first by shifting the vector one position to the right using an LFSR.
- All matrix-vector product is expressed through multiplication of two polynomials modulo a generating polynomial of a cyclic code.