Construction of generalized-involutory MDS matrices

Xuting Zhou[1] and Tianshuo Cong[2]

[1]Department of Computer Science and Technology, BNRist, Beijing 100084, Peoples R China

[2]Tsinghua Univ, Institute for Advanced Study, BNRist, Beijing 100084, Peoples R China

Abstract. Maximum Distance Separable (MDS) matrices are usually used to be diffusion layers in cryptographic designs. The main advantage of involutory MDS matrices lies in that both encryption and decryption share the same matrix-vector product. In this paper, we present a new type of MDS matrices called generalized-involutory MDS matrices, implementation of whose inverse matrix-vector products in decryption is the combination of the matrix-vector products in encryption plus a few extra XOR gates. For the purpose of verifying the existence of such matrices, we found 4×4 Hadamard generalized-involutory MDS matrix over GF(2^4) consuming as little as 38 XOR gates with 4 additional XOR gates for inverse matrix, while the best previous single-clock implementation in IWSEC 2019 needs 46 XOR gates with 51 XOR gates for inverse matrix. For GF(2^8), our results also beat the best previous records in ToSC 2017.

KeyWords: MDS matrix, XOR count, Lightweight cryptography, Involutory matrix

1 Introduction

Most symmetric key primitives like block cipher are based on confusion and diffusion components, which are crucial to the security and efficiency of the cryptographic scheme. The major role of the diffusion layer is to spread the internal dependencies as much as possible and hence provide the best resistance to differential and linear attacks. One way to achieve this is to use Maximum Distance Separable (MDS) matrix which theoretically ensures a perfect diffusion. A typical example is AES [1], which uses a 4×4 MDS matrix over GF(2^8) as the diffusion layer. Some other block ciphers such as CLEFIA [2], FOX [3], KHAZAD [4], and ANUBIS [5] also use MDS matrices as diffusion components. As more and more resource-constrained devices such as RFID tags are widely used in Internet of Things (IoT), good hardware efficiency has become an important design factor in lightweight cryptography. As the diffusion layer is a major component of the cryptographic scheme, lowering the hardware cost of MDS matrices becomes a major design principle. One way to design lightweight cryptography is to use involutory MDS matrices. The main advantage of involutory MDS matrices lies in that both encryption and decryption share the same matrix-vector product.

While the cryptographic scheme composed of involutory MDS matrices could reuse encryption circuits in the decryption process, there is no involutory circulant MDS matrix over fields of even characteristic [6–8]. Also, no companion matrix over fields of even characteristic could yield an involutory MDS matrix [9]. Thus Hadamard matrices are widely studied in terms of involutory property. Sim *et al.* [10] provide the construction of Hadamard or Hadamard-Cauchy involutory MDS matrices. However, there is no lightweight involutory MDS matrix in some cases due to the small candidate set. To overcome this problem, some generalizations on involutory are presented.

Victor *et al.* [11] presented quasi-involutory matrices that the matrix-vector product and its inverse can be implemented by clocking the same LFSR-like architecture. Only one additional bit permutation is needed for the implementation of the inverse matrix-vector product.

Victor *et al.* [6] also relaxed circulancy to θ-circulancy to construct θ-circulant involutory MDS matrices for fields of characteristic 2. Also, the involutory definition is furthered and new direct construction of almost involutory θ-circulant MDS matrices is proposed, which could be efficiently implemented in hardware by adding some transformation compared to normal involutory matrix.

Zaghian *et al.* [12] proposed a generalization of involutory called semi involutory. The merit of this generalization is that "the cost of implementation of these matrices and their inverses are *equal*". Cheon *et al.* [13] studied and generalized semi-involutory matrices: a nonsingular matrix $\mathbf{A}$ is semi-involutory if there exists (nonsingular) diagonal matrices $\mathbf{D}$ and $\mathbf{D}'$ such that $\mathbf{A}^{-1} = \mathbf{DAD}'$.

In this paper, we propose definitions of additive and multiplicative generalized-involutory matrices (GIM). Their inverse matrices are the combinations of some lightweight matrices and themselves using a small number of addition and multiplication operations. Therefore, the inverse matrix-vector product of our proposed matrices can be implemented efficiently. We theoretically prove that all Hadamard MDS matrices over a finite field are multiplicative GIM, thus the only work is to judge whether a Hadamard matrix is MDS or not.We exhaustively search 4×4 generalized-involutory Hadamard and circulant MDS matrices over field GF(2^4) and GF(2^8). We apply BP heuristics [14], a competitive global optimization algorithm, to search generalized-involutory MDS matrices.

We compare our results with previous works in Table 1, where $\mathcal{S}$-xor denotes the XOR count needed for the matrix-vector product after optimization. The total numbers of XOR gates for both $\mathbf{M}v$ and $\mathbf{M}^{-1}v$ for Hadamard multiplicative GIMs over GF(2^4) are $38 + 4 = 42$, which is only 43 percent of the best previous result $46 + 51 = 97$ [15]. Due to limited computing resources, we only implement the inner product of the first row of matrix and vector over GF(2^8). Our Hadamard multiplicative GIM over GF(2^8) and its inverse matrix consume $34 + 3 = 37$ XOR gates totally, which is more lightweight than previous best result $35 + 76 = 111$ [16]. Our circulant candidates also beat the best previous records in ToSC 2017 and FSE 2016.

This paper is structured as follows. Section 2 describes the performance metric, i.e. XOR count. Then, we propose the definition of generalized-involutory MDS matrices in Section 3, followed by experiment results in Section 4, and Section 5 concludes the paper.

2 XOR count

Given a matrix, we use XOR count [19], the number of XOR gates needed, as the metric for the hardware cost.

For a $k \times k$ Hadamard matrix $\mathbf{H}$, there are 2 variant metrics of XOR count:

- The XOR count needed for directly computing the multiplication of $\mathbf{M}$ with a vector is called $\mathcal{D}$-xor, denoted as $\mathcal{D}(\mathbf{M})$. $\mathcal{D}(\mathbf{M})$ is an overestimation of the hardware implementation cost.
- We then use BP heuristic method [14] to optimize the circuit. The XOR count needed after optimization is called $\mathcal{S}$-xor, denoted as $\mathcal{S}(\mathbf{M})$ [16].

Table 1: Experiment results

Galois Field	**Type**	$\mathcal{S}$-**xor(M)**	$\mathcal{S}$-**xor($\mathbf{M}^{-1}$)**	Ref.
GF(2^4)	Had. Involution	**44**	0	FSE20 [17]
GF(2^4)	Had. Involution	47	0	IWSEC19 [15]
GF(2^4)	Had. Involution	48	0	ToSC17 [14]
GF(2^8)	Had. Involution	**36**	0	**ours**
GF(2^8)	Had. Involution	38	0	FSE15 [10], ToSC17 [16]
GF(2^4)	Had	**38**	4	**Multiplicative GIM (ours)**
GF(2^4)	Had	48	54	ToSC17 [14]
GF(2^4)	Had	46	51	IWSEC19 [15]
GF(2^8)	Had	**34**	3	**Multiplicative GIM (ours)**
GF(2^8)	Had	35	76	FSE15 [10],ToSC17 [16]
GF(2^8)	Cir	36	77	ToSC17(AES) [16]
GF(2^8)	Cir	31	75	FSE16 [18] ToSC17 [16]
GF(2^8)	Cir	39	**36**	**Additive GIM (ours)**
GF(2^8)	Cir	37	**18**	**Multiplicative GIM (ours)**

Consider the XOR count of a given $k \times k$ Hadamard matrix $\texttt{had}(h_0, \ldots, h_{k-1})$. Let $\mathcal{D}(h_i)$ be the XOR count of the element h_i. If we directly calculate matrix-vector multiplication, we will cost k^2 multiplications and $k(k-1)$ additions. Therefore, the $\mathcal{D}$-xor over $\mathrm{GF}(2^c)/p(x)$ could be calculated as

$$\mathcal{D}(H_{k,k}) = k \times \sum_{i=0}^{k-1} \mathcal{D}(h_i) + k \times (k-1) \times c.$$

Take matrix $\mathbf{E} = \texttt{had}(0x1, 0x2, 0x8, 0xa)$ and vector $V = (v_0, v_1, v_2, v_3)^\top$ over $\mathrm{GF}(2^4)/\texttt{0x13}$ as an example. Firstly, we should calculate the XOR count of each element in the matrix. Let $v = \sum_{i=0}^{3} v^i x^i$ for $0 \le k \le 3$. For element $0xa$, $0xa \cdot v$ is equal to

$$\begin{aligned} 0xa \cdot v =& (x^3 + x) \times (v^3x^3 + v^2x^2 + v^1x + v^0) \\ =& (v^0v^2v^3)x^3 + (v^1v^2v^3)x^2 + (v^0v^1v^2v^3)x + (v^1v^3). \end{aligned}$$

As there is no AND operation between vector elements, we omit the XOR mark $\oplus$ in the expression without ambiguity. For example, v^0v^2 here denotes $v^0 \oplus v^2$. So $\mathcal{D}(0xa) = 2 + 2 + 3 + 1 = 8$. Similarly, we could acquire $\mathcal{D}(0x1) = 0$, $\mathcal{D}(0x2) = 1$, $\mathcal{D}(0x8) = 3$, and hence

$$\mathcal{D}(E) = 4 \times (0 + 1 + 3 + 8) + 48 = 96.$$

3 Construct generalized-involutory MDS matrices

For involutory MDS matrices, both encryption and decryption share the same matrix-vector product. However, there is no lightweight involutory MDS matrix in some cases due to the small candidate set. To overcome this problem, We propose generalizations on involutory, i.e. additive and multiplicative generalized-involutory matrices (GIM).

3.1 Multiplicative GIM

Definition 1. *(Multiplicative GIM). A $k \times k$ matrix $\mathbf{M}_k$ over field* $\mathrm{GF}(2^c)$ *is a multiplicative GIM if there exists a lightweight matrix $\mathbf{G}_k$ such that*

$$\mathbf{M}_k^{-1} = \mathbf{G}_k \mathbf{M}_k.$$

From Def.1, we could directly get a property that the inverse matrix-vector product $\mathbf{M}_k^{-1} \cdot v$ are simple linear combinations of matrix-vector product $\mathbf{M}_k \cdot v$. Therefore, only $\mathcal{D}(\mathbf{G}_k)$ additional XOR gates are needed for the implementation of the inverse matrix-vector product, which is often much less than original XOR count $\mathcal{D}(\mathbf{M}_k^{-1})$. Especially, matrices are involutory matrices when $\mathbf{G}_k$ is an identity matrix. $\mathbf{G}_k$ could be calculated by $\mathbf{G}_k = \mathbf{M}_k^{-1}\mathbf{M}_k^{-1}$.

Corollary 1. *Invertible Hadamard matrices over finite field* $\mathrm{GF}(2^c)$ *are all multiplicative GIM.*

Proof. For an invertible Hadamard matrix $\mathbf{H}_k$ =`had` $(h_0, ..., h_{k-1})$ and its inverse $\mathbf{H}_k^{-1}$ =`had` $(h_0^{'}, ..., h_{k-1}^{'})$ over $\mathrm{GF}(2^c)$, we have

$$\mathbf{H}_k \cdot \mathbf{H}_k = \sum_{i=0}^{k-1} h_i^2 \cdot \mathbf{I}_k. \tag{1}$$

Here we define a *multiplier-factor* α as:

$$\alpha = \frac{1}{\sum_{i=0}^{k-1} h_i^2}. \tag{2}$$

Multiplying α on both sides of Eq. (1), we can obtain $\alpha\mathbf{H}_k \times \mathbf{H}_k = \mathbf{I}_k$, thus the inverse matrix $\mathbf{H}_k^{-1}$ equals to $\alpha\mathbf{H}_k$. Then, we could get

$$\mathbf{H}_k^{-1} = \alpha \cdot \mathbf{H}_k = diag(\alpha)\mathbf{H}_k, \tag{3}$$

where $diag(\alpha)$ is a diagonal matrix whose main diagonal elements are α and other elements are zero, so $diag(\alpha)$ is a lightweight matrix. According to Def. 1, $\mathbf{H}_k$ is a multiplicative GIM, and extra XOR gates for inverse matrix-vector product is $\mathcal{D}(diag(\alpha)) = k\mathcal{D}(\alpha)$.

3.2 Additive GIM

Definition 2. *(Additive GIM). A $k \times k$ matrix $\mathbf{M}_k$ over field* $\mathrm{GF}(2^c)$ *is an additive GIM if there exists a lightweight matrix $\mathbf{G}_k$ such that*

$$\mathbf{M}_k^{-1} = \mathbf{G}_k + \mathbf{M}_k.$$

The inverse matrix-vector product $\mathbf{M}_k^{-1}v$ is the sum of matrix-vector products of $\mathbf{G}_k v$ and $\mathbf{M}_k v$. Therefore, we need $\mathcal{D}(\mathbf{G}_k)$ XOR gates for the matrix-vector product $\mathbf{G}_k v$ and $k \times c$ XOR gates for the summation of $\mathbf{G}_k v$ and $\mathbf{M}_k v$ to implement the inverse matrix-vector product $\mathbf{M}_k^{-1}v$. The value of $\mathcal{D}(\mathbf{G}_k) + k \times c$ is often less than original XOR count $\mathcal{D}(\mathbf{M}_k^{-1})$. Specially, matrices are involutory matrices when $\mathbf{G}_k$ is a zero matrix. $\mathbf{G}_k$ could be calculated by $\mathbf{G}_k = \mathbf{M}_k^{-1} + \mathbf{M}_k$.

4 Experiment results

In this part, We exhaustively search 4×4 Hadamard and circulant matrices over GF(2^4) and GF(2^8) and check they are MDS matrices or not using the following property: a matrix $\mathbf{M}$ is MDS if and only if every square submatrix of $\mathbf{M}$ is nonsingular. Then we use BP heuristic method [14] to optimize the circuit.

Hadamard and circulant GIMs and the comparison with previous works are shown in Table 2. **XOR count** denotes the number of XOR gates needed for matrix-vector product after optimization. Due to limited computing resources, we only implement the inner product of the first row of matrix and vector over GF(2^8). Thus **XOR count** of a matrix over GF(2^8) denote the number of XOR gates needed for the inner product of the first row of matrix and vector. We list the first row of each matrix and its inverse in the third column, where elements are in hex. Also the XOR counts of matrix and its inverse are listed. For inverse matrix of multiplicative GIM, we listed only the XOR count of the lightweight matrix $\mathbf{G}$. For inverse matrix of additive GIM, we listed the number of XOR gates used to compute both $\mathbf{G}v$ and $\mathbf{M}v + \mathbf{G}v$.

BP heuristic method is powerful, and we obtain Hadamard involutory matrix over GF(2^8) which is better than the best previous result using this optimization method. The XOR counts of the inverse of GIMs are much smaller than that of previous works. Our Hadamard multiplicative GIM over GF(2^4) merely consumes 4 additional XOR gates for inverse matrix, this number is 51 in IWSEC 2019. The total number of XOR gates for both $\mathbf{M}v$ and $\mathbf{M}^{-1}v$ for our Hadamard multiplicative GIM over GF(2^4) is $42 = 38 + 4$, which is even smaller than previous best involutory result 44 in FSE 2020. The first row of our Hadamard multiplicative GIM over GF(2^8) merely consumes 3 additional XOR gates for the inverse matrix, which is more lightweight than previous result 76 in ToSC 2017. The total numbers of XOR gates for both $\mathbf{M}v$ and $\mathbf{M}^{-1}v$ over GF(2^8) are 75 and 55 for circulant multiplicative and additive GIMs repectively, while the best previous result is $106 = 31 + 75$ in ToSC 2017.

Here we use two examples to show the implementation of the inverse of multiplicative and additive GIMs. Given a matrix over GF(2^8), we use $\mathcal{S}(\mathbf{M})$ to denote the number of XOR gates needed for the inner product of the first row of $\mathbf{M}$ and vector after optimization.

Example 1. Given a multiplicative GIM $\mathbf{H} = \mathtt{had}(0x01, 0x02, 0x08, 0x8e)$ over GF(2^8)/$0x11d$, we need 34 XOR gates and $\mathcal{D}(\alpha)$ extra XOR gates according to Corollary 1, where α is 0x8e by Eq. 2, thus $\mathcal{D}(0x8e) = 3$ extra XOR gates are needed. If we directly implement the first row of its inverse matrix $\mathbf{H} = \mathtt{had}(0x8e, 0x01, 0x04, 0x47)$, $\mathcal{S}(\mathbf{H}^{-1}) = 34$ XOR gates are needed. Thus $34 - 3 = 31$ XOR gates are saved.

Example 2. Given an additive GIM $\mathbf{C} = \mathtt{cir}(0x02, 0x8e, 0x53, 0x01)$ over GF(2^8)/$0x14d$, we need 39 XOR gates and $\mathcal{S}(\mathbf{G})+8 = 36$ extra XOR gates, where 8 is the number of XOR gates used to XOR $\mathbf{C}v$ and $\mathbf{G}v$. If we directly implement the first row of its inverse matrix $\mathbf{C} = \mathtt{cir}(0x51, 0x8b, 0x53, 0xa7)$, $\mathcal{S}(\mathbf{C}^{-1}) = 54$ XOR gates are needed. Thus $54 - 36 = 18$ XOR gates are saved.

5 Conclusion

In this paper, we propose the concept of generalized-involutory matrix and provide two types of GIMs, i.e. additive GIM and multiplicative GIM. By applying BP heuristic on

Table 2: XOR counts of 4×4 Hadamard and circulant MDS matrices over GF(2^4) and GF(2^8)

Galois Field	**Type**	**M, $\mathbf{M^{-1}}$**	**XOR count**	**G**	Ref.
GF(2^4)/$0x13$	Had	$(1, 4, 9, d)$	44	-	FSE20 [17]
	Involution	$(1, 4, 9, d)$	0		
GF(2^4)/$0x13$	Had	$(2, 3, 9, d)$	**38**	diag(9)	**Multiplicative GIM (ours)**
		$(1, 8, d, f)$	4		
GF(2^4)/$0x13$	Had	$(1, 2, 8, 9)$	48	-	ToSC17 [14]
		$(d, 9, 2, f)$	54		
GF(2^4)/$0x13$	Had	$(1, 2, 8, 9)$	46	-	IWSEC19 [15]
		$(d, 9, 2, f)$	51		
GF(2^8)/$0x12b$	Had	$(01, 02, 95, 97)$	**36**	-	**ours**
	Involution	$(01, 02, 95, 97)$	0		
GF(2^8)/$0x165$	Had	$(01, 02, b0, b2)$	38	-	FSE15 [10]
	Involution	$(01, 02, b0, b2)$	0		
GF(2^8)/$0x11d$	Had	$(01, 02, 08, 8e)$	**34**	diag($8e$)	**Multiplicative GIM (ours)**
		$(8e, 01, 04, 47)$	**3**		
GF(2^8)/$0x1c3$	Had	$(01, 02, 04, 91)$	35	-	FSE15 [10],ToSC17 [16]
		$(27, 4e, 9c, 79)$	76		
GF(2^8)/$0x11b$	Cir	$(02, 03, 01, 01)$	36	-	ToSC17(AES) [16]
		$(0e, 0b, 0d, 09)$	77		
GF(2^8)/$0x1c3$	Cir	$(01, 01, 02, 91)$	31	-	FSE16 [18] ToSC17 [16]
		$(55, 5a, 71, 41)$	75		
GF(2^8)/$0x14d$	Cir	$(02, 8e, 53, 01)$	39	cir(53, 05, 00, a6)	**Additive GIM (ours)**
		$(51, 8b, 53, a7)$	**36**		
GF(2^8)/$0x187$	Cir	$(01, a2, a3, c3)$	37	cir(01, 00, 05, 00)	**Multiplicative GIM (ours)**
		$(a7, 63, a6, 60)$	**18**		

4×4 Hadamard and circulant matrix over $\mathrm{GF}(2^4)$ and $\mathrm{GF}(2^8)$, we obtain state-of-the-art XOR counts. Our work can be extended to a larger scope, such as 8×8 matrix over $\mathrm{GL}(m, \mathbb{F}_q)$. Also, we can extend the concept of generalized-involutory to find more lightweight candidates.

References

1. D. Joan and R. Vincent, "The design of Rijndael: AES-the advanced encryption standard," in Information Security and Cryptography, springer, 2002.
2. T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, "The 128-bit blockcipher CLEFIA," in International workshop on fast software encryption, pp. 181–195, Springer, 2007.
3. P. Junod and S. Vaudenay, "FOX: a new family of block ciphers," in International Workshop on Selected Areas in Cryptography, pp. 114–129, Springer, 2004.
4. P. Barreto and V. Rijmen, "The Khazad legacy-level block cipher," Primitive submitted to NESSIE, vol. 97, p. 106, 2000.
5. P. S. Barreto, "The Anubis block cipher," NESSIE, 2000.
6. V. Cauchois and P. Loidreau, "On circulant involutory MDS matrices," Designs, Codes and Cryptography, vol. 87, no. 2, pp. 249–260, 2019.
7. A. Kesarwani, S. Sarkar, and A. Venkateswarlu, "Exhaustive Search for Various Types of MDS Matrices," IACR Transactions on Symmetric Cryptology, pp. 231–256, 2019.
8. S. Sarkar and H. Syed, "Lightweight diffusion layer: Importance of Toeplitz matrices," IACR Transactions on Symmetric Cryptology, pp. 95–113, 2016.
9. K. C. Gupta, S. K. Pandey, and A. Venkateswarlu, "Almost involutory recursive MDS diffusion layers," Designs, Codes and Cryptography, vol. 87, no. 2-3, pp. 609–626, 2019.
10. S. M. Sim, K. Khoo, F. Oggier, and T. Peyrin, "Lightweight MDS involution matrices," in International Workshop on Fast Software Encryption, pp. 471–493, Springer, 2015.
11. V. Cauchois, P. Loidreau, and N. Merkiche, "Direct construction of quasi-involutory recursive-like MDS matrices from 2-cyclic codes," 2016.
12. A. Zaghian and M. Mousavi, "Design and Construction of Lightweight MDS Semi Involutory Matrices Based on the Recursive Structures and Binary Sparse Matrices," Journal of Advanced Defense Science and Technology, vol. 10, no. 4, pp. 407–417, 2020.
13. G.-S. Cheon, B. Curtis, and H. Kim, "Semi-involutory matrices and signed self-inverse," Linear Algebra and its Applications, vol. 622, pp. 294–315, 2021.
14. T. Kranz, G. Leander, K. Stoffelen, and F. Wiemer, "Shorter linear straight-line programs for MDS matrices," IACR Transactions on Symmetric Cryptology, pp. 188–211, 2017.
15. S. Banik, Y. Funabiki, and T. Isobe, "More results on shortest linear programs," in International Workshop on Security, pp. 109–128, Springer, 2019.
16. J. Jean, T. Peyrin, S. M. Sim, and J. Tourteaux, "Optimizing implementations of lightweight building blocks," IACR Transactions on Symmetric Cryptology, pp. 130–168, 2017.
17. Z. Xiang, X. Zeng, D. Lin, Z. Bao, and S. Zhang, "Optimizing implementations of linear layers," IACR Transactions on Symmetric Cryptology, pp. 120–145, 2020.
18. M. Liu and S. M. Sim, "Lightweight MDS generalized circulant matrices," in International Conference on Fast Software Encryption, pp. 101–120, Springer, 2016.
19. K. Khoo, T. Peyrin, A. Y. Poschmann, and H. Yap, "FOAM: searching for hardware-optimal SPN structures and components with a fair comparison," in International Workshop on Cryptographic Hardware and Embedded Systems, pp. 433–450, Springer, 2014.