RANDOM MATRICES OVER $\mathbb{F}_2$ IN CRYPTOGRAPHY

FALKUSH

Abstract. We describe a one-way function based on random matrices over $\mathbb{F}_2$. A hash function and a cryptographic pseudorandom number generator based on this one-way function are described.

1. Introduction

Let A be an $n \times n$ matrix over $\mathbb{F}_2$ and let $V := \mathbb{F}_2^n$. We say that A is *primitive* if and only if

$$\{A^i v\}_{i=0}^{2^n-2} = V - \{v_0\}$$

for any $v \in V - \{v_0\}$ where v_0 is the zero vector. In other words, the matrix A acting repeatedly on a non-zero vector will cycle through all the non-zero vectors of V.

A Java implementation is available at github.com/falkush/RP269.

2. One-way Function

Let A_0 and A_1 be two primitive $n \times n$ matrices. The one-way function $F : V \to V$ is defined as

$$F(v) := M_v v$$

where

$$M_v := A_{v(n)} A_{v(n-1)} \dots A_{v(1)}$$

and $v(i)$ is the ith bit of v.

3. Hash Function

Let m be a message divided into ℓ blocks of n bits, under the usual padding. Let

$$h_i = A_0 h_{i-1} \oplus F(m_i)$$

where m_i is the ith block and $h_0 := (1, 0, \dots, 0, 0) \in V$. Then, the hash function is defined as

$$H(m) := F(h_\ell).$$

4. Cryptographic Pseudorandom Number Generator

Let $r_0, s_0 \in V$ be two secret keys. We generate the states s_i by

$$r_i = A_0 r_{i-1}$$
$$s_i = F(s_{i-1} \oplus r_{i-1}).$$

The key r_0 protects against the *next-bit test* and the key s_0 protects against *state compromise extensions*.

Date: November 1, 2022.

5. Circuit

The action of a matrix on a vector can be computed using around $n^2/2$ XOR gates. By connecting the XOR gates in a binary tree structure, $\lfloor \log_2(n) \rfloor$ gates needs to be evaluated successively to compute the action. The computation of F can be made m times faster using 2^m such circuits. As an example, the following circuit computes the action of

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

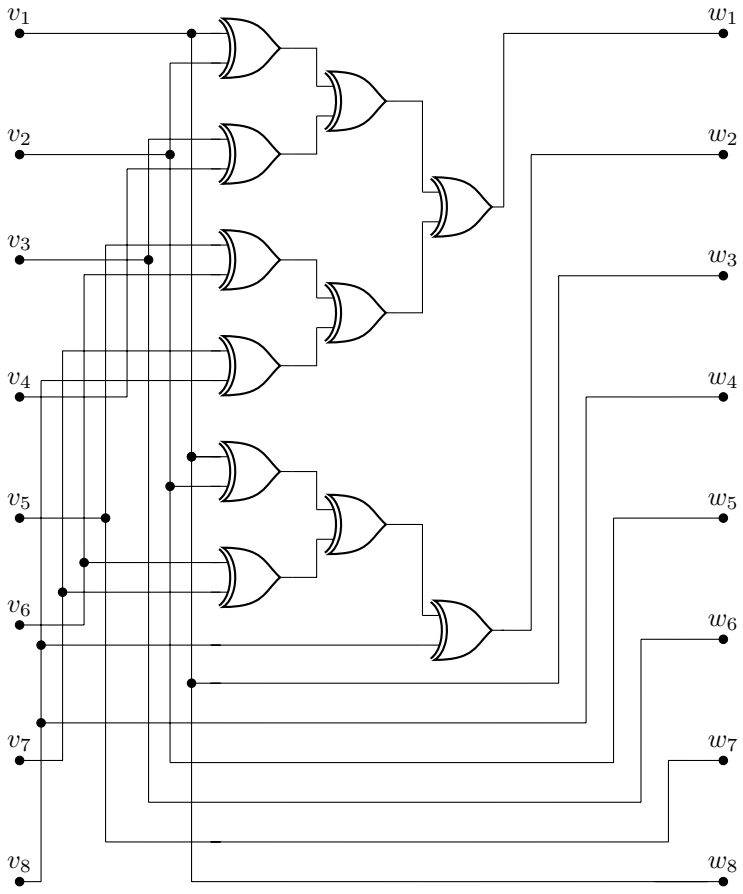

Figure 1. Matrix action circuit example ($n = 8$).

6. Statistics

We present numerical data on the size of the image of F, F_{Δ_1}, and F_{Δ_1,Δ_2} where $\Delta_1 = (0, 0, \ldots 0, 0, 1)$, $\Delta_2 = (0, 0, \ldots 0, 1, 0)$, F_{Δ_1} is the first order differential at Δ_1, and F_{Δ_1,Δ_2} is the second order differential at Δ_1, Δ_2.

For a given dimension n, a simulation consists of randomly choosing two distinct primitive matrices and evaluating the one-way function on the whole domain. One million simulations were performed for each line of the tables below.

The quantity μ_o is the average of the simulations and μ_e is the expected value for random functions, given by

$$b\left(1 - \left(1 - \frac{1}{b}\right)^a\right)$$

where a is the size of the domain and b is the size of the codomain.

The quantity σ_o is the corrected standard deviation of the simulations and σ_c is the corrected standard deviation when using Java's default random function.

n	μ_o	μ_e	μ_o/μ_e	σ_o	σ_c
5	20.803003	20.414230743797866	1.0190441785968471	1.748598936174113	1.767387878619437
6	41.01376	40.640862448389925	1.0091754340125931	2.478938241976756	2.4976720225220785
7	81.470055	81.09597236197925	1.004612838678979	3.5183927688736905	3.5249066645569487
8	162.374742	162.00710274776674	1.0022692786056773	4.9793075047579896	4.98731424857502
9	324.192228	323.8298156776779	1.0011191443924448	7.049477150196609	7.067203815712816
10	647.831977	647.4754668428584	1.0005506156995878	9.958697427230403	9.985773325869355
11	1295.13735	1294.7668816332418	1.0002861274658885	14.113804454397691	14.130517160352843
12	2589.713745	2589.3497673959537	1.0001405671835568	19.971201213844218	19.961849228467376

Figure 2. Size of the image of F for random matrices, performing one million simulations for each n.

n	μ_o	μ_e	μ_o/μ_e	σ_o	σ_c
5	12.710451	12.745270290173682	0.9972680618471838	1.3451729205158665	1.3172060294941428
6	25.330462	25.33496665845165	0.9998221959984248	1.870978451804394	1.8668058097196212
7	50.508779	50.51635309520687	0.9998500664687217	2.6454137535313693	2.6458132537514385
8	100.87437	100.88010541335538	0.9999431462394703	3.7416027451736644	3.7420118513650777
9	201.609509	201.60809550983953	1.000007011078384	5.290650977895515	5.289325860069415
10	403.072833	403.06431737827006	1.0000211272031851	7.474891720160047	7.487285593432481
11	805.978415	805.9768816905535	1.000001902423607	10.584839211182947	10.590202128534287
12	1611.799379	1611.802070537417	0.9999983301067382	14.972957436786272	14.96842361265906

Figure 3. Size of the image of F_{Δ_1} for random matrices, performing one million simulations for each n.

n	μ_o	μ_e	μ_o/μ_e	σ_o	σ_c
5	7.116513	7.177603848249419	0.9914886848673993	0.8433258160577274	0.7852940026529368
6	14.224129	14.25502905962157	0.9978323397663849	1.1581435716937398	1.1307648775184027
7	28.373867	28.41131186819699	0.99868204367434	1.646431042299123	1.611234564487996
8	56.70057	56.72458000500663	0.9995767266147318	2.3254287954523747	2.2893831130108033
9	113.321474	113.35146419563728	0.9997354229532888	3.2999665690887476	3.2460684964984465
10	226.574204	226.60540570891055	0.9998623081880463	4.662570697152967	4.595848731118722
11	453.092682	453.1133750959782	0.9999543313062127	6.587047398954209	6.49386673439979
12	906.103319	906.1293569991394	0.9999712645893898	9.318824873595167	9.192685281541062

FIGURE 4. Size of the image of F_{Δ_1,Δ_2} for random matrices, performing one million simulations for each n.

order	μ_o	μ_e	μ_o/μ_e	σ_o	σ_c
0	2589.713745	2589.3497673959537	1.0001405671835568	19.971201213844218	19.961849228467376
1	1611.799379	1611.802070537417	0.9999983301067382	14.972957436786272	14.96842361265906
2	906.103319	906.1293569991394	0.9999712645893898	9.318824873595167	9.192685281541062
3	481.323555	481.3478516278383	0.9999495237638308	5.148911309516087	5.097486564474034
4	248.182777	248.1934560411746	0.9999569729140124	2.697242822474433	2.67592323669592
5	126.030914	126.03581829063069	0.9999610881200502	1.3750382595892434	1.372989915854195
6	63.509718	63.51028664900605	0.9999910463479846	0.6937925063187421	0.6927643245097205
7	31.879289	31.879201366466987	1.000002748924981	0.3455661642796845	0.3472877793412859
8	15.970843	15.97073647713296	1.0000066698781984	0.17037282171138207	0.17094480138958634
9	7.993083	7.993167399341473	0.999989441064192	0.08304915416885357	0.08243594579845105
10	3.998596	3.998535394654027	1.0000151568862075	0.03744369354150557	0.03788055524852382
11	1.999743	1.999755859375	0.9999935695275303	0.016029167412383215	0.015457785027070275

FIGURE 5. Size of the image for higher order differentials of F for $n = 12$, performing one million simulations for each order.

image size	F	Java Random
$[2490, 2500)$	0	6
$[2500, 2510)$	35	39
$[2510, 2520)$	211	208
$[2520, 2530)$	1063	1148
$[2530, 2540)$	4790	4972
$[2540, 2550)$	15966	16659
$[2550, 2560)$	43079	44340
$[2560, 2570)$	90440	92329
$[2570, 2580)$	148553	150722
$[2580, 2590)$	191390	192554
$[2590, 2600)$	192353	191461
$[2600, 2610)$	151437	149397
$[2610, 2620)$	92879	90928
$[2620, 2630)$	44756	42947
$[2630, 2640)$	16683	16325
$[2640, 2650)$	4973	4701
$[2650, 2660)$	1162	1049
$[2660, 2670)$	199	191
$[2670, 2680)$	28	22
$[2680, 2690)$	2	2
$[2690, 2700)$	1	0

FIGURE 6. Distribution of the simulations for order 0

image size	F	Java Random
$[1530, 1540)$	0	1
$[1540, 1550)$	20	20
$[1550, 1560)$	254	215
$[1560, 1570)$	2240	2264
$[1570, 1580)$	13149	13184
$[1580, 1590)$	52765	52829
$[1590, 1600)$	136839	137163
$[1600, 1610)$	233176	232815
$[1610, 1620)$	257681	258144
$[1620, 1630)$	185530	185399
$[1630, 1640)$	86426	86171
$[1640, 1650)$	26201	26091
$[1650, 1660)$	5037	5010
$[1660, 1670)$	637	637
$[1670, 1680)$	44	53
$[1680, 1690)$	1	4

FIGURE 7. Distribution of the simulations for order 1

image size	F	Java Random
$[780, 790)$	9	0
$[790, 800)$	59	0
$[800, 810)$	91	0
$[810, 820)$	71	0
$[820, 830)$	15	0
$[830, 840)$	6	0
$[840, 850)$	0	0
$[850, 860)$	0	0
$[860, 870)$	51	37
$[870, 880)$	2154	2196
$[880, 890)$	34207	34581
$[890, 900)$	197676	197329
$[900, 910)$	406484	406345
$[910, 920)$	288046	287690
$[920, 930)$	66450	67144
$[930, 940)$	4569	4602
$[940, 950)$	110	76
$[950, 960)$	2	0

FIGURE 8. Distribution of the simulations for order 2

image size	F	Java Random
$[430, 440)$	23	0
$[440, 450)$	177	0
$[450, 460)$	409	34
$[460, 470)$	12737	12588
$[470, 480)$	337858	339095
$[480, 490)$	598699	597924
$[490, 500)$	50070	50316
$[500, 510)$	27	43

FIGURE 9. Distribution of the simulations for order 3

image size	F	Java Random
$[210, 220)$	1	0
$[220, 230)$	2	0
$[230, 240)$	2249	1726
$[240, 250)$	670392	669710
$[250, 260)$	327356	328564

FIGURE 10. Distribution of the simulations for order 4

image size	F	Java Random
115	3	0
116	1	0
117	5	2
118	19	20
119	141	95
120	599	538
121	2597	2548
122	10037	9955
123	33017	32804
124	86677	86778
125	181383	180749
126	276720	275904
127	274946	275858
128	133855	134749

FIGURE 11. Distribution of the simulations for order 5

image size	F	Java Random
57	1	2
58	9	7
59	100	109
60	1259	1193
61	11293	11131
62	73132	73224
63	304542	304683
64	609664	609651

FIGURE 12. Distribution of the simulations for order 6

image size	F	Java Random
28	8	4
29	185	225
30	6035	6171
31	108054	108640
32	885718	884960

FIGURE 13. Distribution of the simulations for order 7

image size	F	Java Random
13	5	1
14	345	392
15	28452	28503
16	971198	971104

FIGURE 14. Distribution of the simulations for order 8

image size	F	Java Random
6	14	18
7	6889	6770
8	993097	993212

FIGURE 15. Distribution of the simulations for order 9

image size	F	Java Random
3	1404	1437
4	998596	998563

FIGURE 16. Distribution of the simulations for order 10

image size	F	Java Random
1	257	239
2	999743	999761

FIGURE 17. Distribution of the simulations for order 11

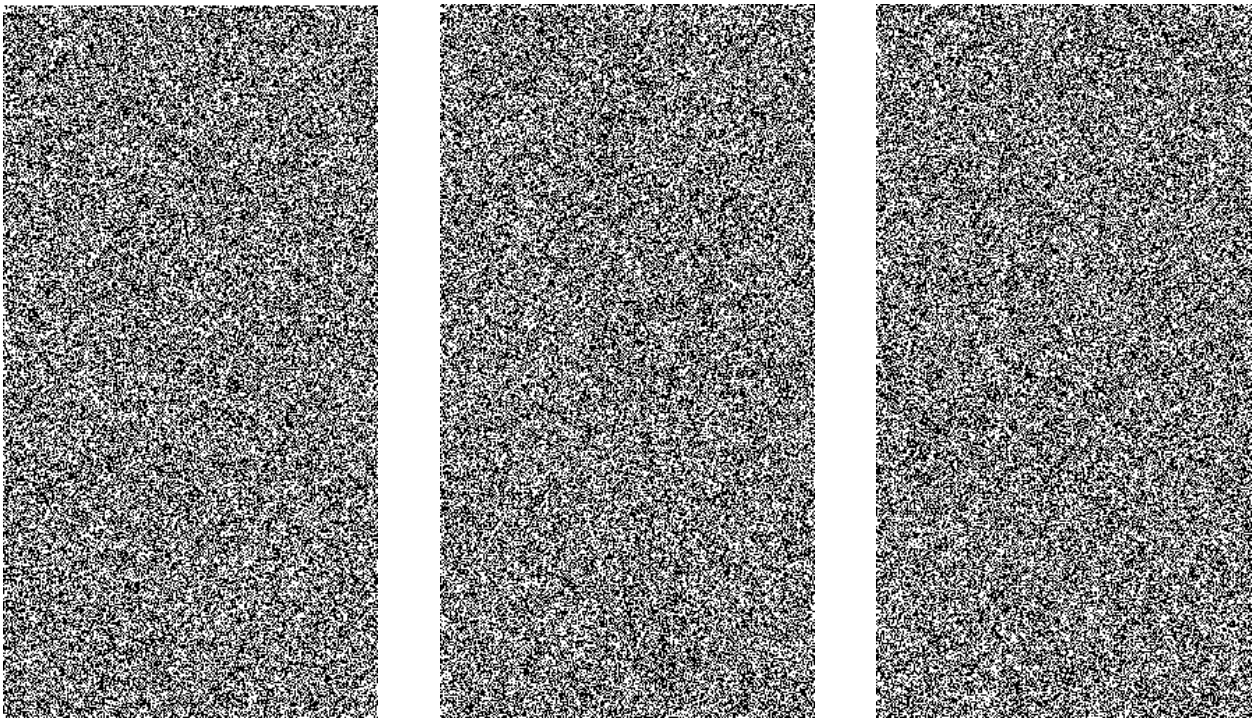

FIGURE 18. Values of $F(i)$, $F_{\Delta_1}(2i)$, and $F_{\Delta_1,\Delta_2}(4i)$ respectively for $0 \leq i \leq 512$ with $n = 269$. Matrices available on the github repository.

7. Skew

The skew in the distribution of the second order gets further away as we increase n. For $n = 12$, there were 251 simulations off the main distribution and for $n = 13$ there were 117. More numerical computations are needed to understand this skew in the distributions. It is crucial to understand this skew in order to justify if A_0 and A_1 can be chosen randomly.

image size	F	Java Random
$[1570, 1580)$	3	0
$[1580, 1590)$	6	0
$[1590, 1600)$	22	0
$[1600, 1610)$	30	0
$[1610, 1620)$	23	0
$[1620, 1630)$	22	0
$[1630, 1640)$	9	0
$[1640, 1650)$	2	0
$[1650, 1660)$	0	0
$[1660, 1670)$	0	0
$[1670, 1680)$	0	0
$[1680, 1690)$	0	0
$[1690, 1700)$	0	0
$[1700, 1710)$	0	0
$[1710, 1720)$	0	0
$[1720, 1730)$	0	0
$[1730, 1740)$	0	0
$[1740, 1750)$	2	1
$[1750, 1760)$	35	43
$[1760, 1770)$	603	613
$[1770, 1780)$	5821	6017
$[1780, 1790)$	35270	35181
$[1790, 1800)$	123315	122609
$[1800, 1810)$	251198	251250
$[1810, 1820)$	295403	295506
$[1820, 1830)$	197856	197689
$[1830, 1840)$	73401	74224
$[1840, 1850)$	15210	15116
$[1850, 1860)$	1666	1667
$[1860, 1870)$	100	79
$[1870, 1880)$	3	5

Figure 19. Distribution of the simulations for order 2 with $n = 13$