A Short Note on a Weight Probability Distribution Related to SPNs

Sondre Rønjom

Department of Informatics
University of Bergen, Norway

Abstract. We report on a simple technique that supports some recent developments on AES by Grassi and Rechberger and Bao, Guo and List. We construct a weight transition probability matrix related to AES that characterises fixed configurations of active bytes in differences of ciphertexts when plaintext differences are fixed to some (possibly other) configuration of active bytes. The construction is very simple and requires only a little bit of linear algebra. The derived probabilities are essentially identical to recent results on 5- and 6-rounds AES derived through more sophisticated means, indicating that it might be worth a further investigation.

1 Introduction

We consider transition probability distributions related to active bytes in differences of ciphertexts and plaintexts in AES. The objective is to determine whether the probability distribution for configurations of active bytes in ciphertext differences vary depending on the distribution of configurations of active bytes in the plaintexts. We show how to utilize symmetries in the AES MixColumn matrix to construct efficient (small) transition matrices for r rounds of AES where r is any number of rounds. The results and techniques are very simple and require only a little bit of linear algebra [1]. We are unable to prove how exact these transition probability matrices emulate the true AES distributions, however, we can confirm that our method obtains results that seem to be identical to recent results reported with more sophisticated analysis, presented in several recent papers on 5- and 6-rounds distinguishers for AES (e.g. [1], [3], [5] and [4]). In particular, probabilities derived from the weight transition probability matrix matches the results recently presented by Bao, Guo and List in [1] on 5- and 6-rounds AES and Grassi and Rechberger in [4] on 5-rounds AES. This may support the view that, as long as the s-box is sufficiently generic (e.g. not linear), the s-box layer has little effect on this kind of structural analysis. It is thus tempting to conjecture that the presented weight transition probability matrix emulates the true AES-distributions.

[1] Simple C/C++ code for experimenting with and verifying our results can be found at https://github.com/sondrer/SPNTransitionProbability

2 Weight Probability Distributions in AES

One round of AES [2] consists of four steps. First, a non-linear permutation SubBytes is applied to each individual byte of the state. Then each row is shifted cyclically via ShiftRows, followed by applying a fixed linear transformation $\mathrm{M} \in \mathbb{F}_{2^8}^{4\times 4}$ to each column (MixColumns). In the end a fixed round-key is added to the whole state (AddKey).

We view states in AES in terms of the SuperBox representation, i.e. as a collection of vectors $S = (s_0, s_1, s_2, s_3) \in (\mathbb{F}_{2^8}^4)^4$ corresponding to the columns of the AES state. To each state we associate a vector $\nu(S) \in (\mathbb{F}_2^4)^4$ which indicates the active bytes in each column s_i of the state. Let $\rho(s_i) \in \mathbb{F}_2^4$ denote this vector which is 1 in position i if the i'th byte of the vector s_i is non-zero such that the vector defined as

$$\nu(S) = (\rho(s_0), \rho(s_1), \rho(s_2), \rho(s_3)) \tag{1}$$

is the configuration of active bytes for a state S. We also use a vector

$$\wp(S) = (\mathrm{wt}(s_0), \mathrm{wt}(s_1), \mathrm{wt}(s_2), \mathrm{wt}(s_3)) \tag{2}$$

to identify the weights of the columns of a state. To simplify the notation we threat vectors also as integers, i.e. $c \in \mathbb{F}_2^4$ is also treated as an integer $\sum_{i=0}^{3} c_i \cdot 2^i$, $u \in (\mathbb{F}_2^4)^4$ as the integer $\sum_{i=0}^{3}(\sum_{j=0}^{3} c_j \cdot 2^j)16^j$, and $a \in \mathbb{Z}_5^4$ as an integer $\sum_{i=0}^{3} a_i \cdot 5^i$. The indices of the matrices will then correspond naturally to configurations of active bytes and weights and are then easily derived from each other.

2.1 A $2^{16} \times 2^{16}$ Transition Probability Matrix for Active Bytes

The MixColumns matrix $\mathrm{M} \in \mathbb{F}_{2^8}^{4\times 4}$ is applied to each column of the state. We are interested in the transition probabilities for active bytes through M, i.e. for two binary vectors $u, v \in \mathbb{F}_2^4$ let

$$\mathrm{T}_{\mathrm{M}}(u, v) = \Pr(\rho(x \cdot \mathrm{M}) = v \,|\, \rho(x) = u) \tag{3}$$

denote the probability that a vector $x \cdot \mathrm{M}$ is non-zero in byte positions indicated by the 1's in v when x is non-zero in positions indicated by the 1's in u. The matrix M in AES is derived from a linear $[8, 4, 5]$ MDS code over $\mathbb{F}_{2^8}$. We have computed the transition probabilities exhaustively[2], but there exist explicit formulas for the weight distribution of MDS codes that simplifies this for the general case. There are in total 16 different configurations of active bytes at the input and 16 configurations of active output bytes, so $\mathrm{T}_{\mathrm{M}} \in \mathbb{R}^{16\times 16}$. Let $u, v \in \mathbb{F}_2^4$ and let Z denote the 16×16 matrix where $\mathrm{Z}(u, v)$ counts the number of elements $x \in \mathbb{F}_{2^8}^4$ with active bytes in positions indicated by u and where $y = x \cdot M$ have active byes indicated by v. The state transition for a column through the MixColumns layer is then as follows.

[2] Table for Z can be found at https://github.com/sondrer/SPNTransitionProbability

Definition 1. *Let* $\mathrm{T_M} \in \mathbb{R}^{16\times 16}$ *denote the transition probability matrix for active bytes over the MixColumn* M *with entries*

$$\mathrm{T_M}(u,v) = \frac{\mathrm{Z}(u,v)}{(2^8-1)^{\mathrm{wt}(u)}} \tag{4}$$

for indicators $u, v \in \mathbb{F}_2^4$ *for configurations of active bytes.*

Since the MC-layer applies the matrix M to each column individually, it is now straight-forward to construct a transition probability matrix for the whole MC-layer. The MC-layer maps an input state $x = (x_0, x_1, x_2, x_3) \in (\mathbb{F}_{2^8}^4)^4$ to an output state $y = (y_0, y_1, y_2, y_3)$ where

$$y_i = \mathrm{MC}(x)_i \tag{5}$$

$$= x_i \cdot \mathrm{M} \tag{6}$$

thus we have the following trivial extension.

Definition 2. *For vectors* $u, v \in (\mathbb{F}_2^4)^4$ *indicating the active bytes in each column, let a matrix* $\mathrm{T_{MC}} \in \mathbb{R}^{2^{16}\times 2^{16}}$ *with entries*

$$\mathrm{T_{MC}}(u,v) = \Pr(\nu(\mathrm{MC}(x)) = v \,|\, \nu(x) = u) \tag{7}$$

$$= \prod_{i=0}^{3} \Pr(\rho(\mathrm{MC}(x)_i) = v_i \,|\, \rho(x_i) = u_i) \tag{8}$$

$$= \prod_{i=0}^{3} \mathrm{T_M}(u_i, v_i) \tag{9}$$

denote the transition probability matrix for the full MixColumns layer in AES and where u, v *are also treated as indices* $0 \le u, v < 2^{16}$ *and* u_i, v_i *as indices* $0 \le u_i, v_i < 16$.

Similarly, let $\mathrm{T_{SR}}$ denote the $2^{16} \times 2^{16}$ matrix with indices

$$\mathrm{T_{SR}}(u,v) = \Pr(\nu(\mathrm{SR}(y)) = v \,|\, \nu(x) = u) \tag{10}$$

where the vectors $u, v \in (\mathbb{F}_2^4)^4$ are treated as indices in the range $0 \le u, v, < 2^{16}$. The $\mathrm{T_{SR}}$ matrix is a permutation matrix and has a single 1 in each row and column, hence there is no uncertainty associated with it. Notice also that the SubBytes layer corresponds to the identity map with regards to active bytes transition probabilities, thus this layer is disregarded (determining the real effect, if any, of the s-box layer is the main remaining open problem). We use the SuperBox representation, which means that we remove the first SR-layer in order to work with columns and may or may not remove the final linear layer. Then we define the following r-round transition matrices for active bytes.

Definition 3. *Let* $\mathrm{T_{MC\circ SR}} = \mathrm{T_{SR}} \cdot \mathrm{T_{MC}}$. *Then define the r-round transition probability matrix for active bytes as*

$$\mathrm{T}_{f^r} = \mathrm{T_{MC}} \cdot \mathrm{T}_{\mathrm{MC}\circ\mathrm{SR}}^{r-1} \cdot \tag{11}$$

If the aim is solely to distinguish AES reduced to r rounds, then since the adversary may remove the first SR layer and last MC $\circ$ SR-layer, the adversary can work with $\mathrm{T}_{f^{r-1}}$. If the aim is to construct an r-round distinguisher in the hope to extend it to a $(r+t)$-round key-recovery, the adversary may consider only removing the first SR-layer and thus work with the T_{f^r}-matrix. The $2^{16} \times 2^{16}$ matrices are quite large, but in the next section we show how to compress them down to 625×625 *weight transition probability* matrices.

2.2 The Weight Transition Probability

In this section we will construct a compressed weight transition probability that only depends on the Hamming weight of columns. The difference will be that we now only consider the number of active bytes in each column and do no longer have control over the exact active bytes. Thus, while the transition probability distributions derived from the previous matrices can be thought of as distributions for the exact configurations of active bytes in the state columns, we now only consider distributions on the number of active bytes (Hamming weight) of the state columns.

The MixColumns matrix M is symmetric with respect to weights in the sense that

$$\mathrm{T}_{\mathrm{M}}(u, v) = \mathrm{T}_{\mathrm{M}}(u', v') \tag{12}$$

for any choice of $u', v' \in \mathbb{F}_2^4$ with $\mathrm{wt}(u) = \mathrm{wt}(u')$ and $\mathrm{wt}(v) = \mathrm{wt}(v')$. The weight transition probabilities through the M-matrix therefore depend only on the number, and not on the particular configuration, of active bytes. So we can construct a compressed weight transition probability matrix $\mathrm{C}_{\mathrm{M}} \in \mathbb{R}^{5\times 5}$ for the matrix M that satisfy

$$\begin{aligned} \mathrm{C}_{\mathrm{M}}(a, b) &= \Pr(\mathrm{wt}(\mathrm{MC}(x)) = b \mid \mathrm{wt}(x) = a) && (13)\\ &= \sum_{\substack{v \in \mathbb{F}_2^4 \\ \mathrm{wt}(v)=b}} \mathrm{T}_{\mathrm{M}}(u, v) && (14)\\ &= \binom{4}{b} \mathrm{T}_{\mathrm{M}}(u', v') && (15) \end{aligned}$$

for weights a, b and where v, u', v' are any fixed vectors with $\mathrm{wt}(v) = \mathrm{wt}(v') = b$ and $\mathrm{wt}(u') = a$. This probability follows since there are $\binom{4}{\mathrm{wt}(b)}$ possible byte configurations for a vector $b \in \mathbb{F}_q^4$ of weight $\mathrm{wt}(b)$ at the output and for each of those the probability is $\mathrm{T}_{\mathrm{M}}(a, b)$. We can now construct a 625×625 weight transition probability matrix C_{MC} with entries

$$\mathrm{C}_{\mathrm{MC}}(u, v) = \prod_{k=0}^{3} \mathrm{C}_{\mathrm{M}}(u_k, v_k) \tag{16}$$

which is the probability for a state S with column weights $\wp(S) = (u_0, u_1, u_2, u_3)$ to map to a state with column weights $\wp(\mathrm{MC}(S)) = (v_0, v_1, v_2, v_3)$ through the MixColumn layer.

We may construct a weight transition matrix $\mathrm{C_{SR}}$ for the ShiftRows layer in a similar fashion. For column weight indicators $\wp(S) = u$ and $\wp(\mathrm{SR}(S)) = v$ the entries of this matrix are given by

$$\mathrm{C_{SR}}(u, v) = \frac{1}{\prod_{j=0}^{3} \binom{4}{u_j}} \sum_{\substack{a,b \in (\mathbb{F}_2^4)^4 \\ \mathrm{wt}(a_i)=u_i \\ \mathrm{wt}(b_i)=v_i}} \mathrm{T_{SR}}(a, b). \tag{17}$$

The probability follows as the sum sums over all possible active byte configurations in the output while the first fraction averages over the number of possible byte configurations in the input. We can now construct weight transition probability matrices for r rounds of AES.

Definition 4. *Let* $\mathrm{C_{MC \circ SR}} = \mathrm{C_{SR}} \cdot \mathrm{C_{MC}}$. *Then let*

$$C_r = \mathrm{C_{MC}} \cdot \mathrm{C_{MC \circ SR}}^{r-1}$$

denote the weight transition probability matrix for r *rounds of AES.*

3 Some Results

If $a \in \mathbb{R}^{625}$ denotes a weight probability distribution for the plaintext difference, then $b = a \cdot C_{r-1}$ is the weight distribution on the ciphertext differences after r rounds, when the last linear layer is omitted (thus we focus on reduced round distinguishers). The uniform distribution is given by a vector $q \in \mathbb{R}^{625}$ of values

$$q_v = 2^{-128} \prod_{i=0}^{3} \binom{4}{v_i} (2^8 - 1)^{v_i}$$

which is the probability that the output difference has weight pattern (v_0, v_1, v_2, v_3) where $v = \sum_{i=0}^{3} v_i \cdot 5^i$ regardless of the input. Now the goal is to determine vectors $a, e \in \mathbb{R}^{624}$ and investigate the sum

$$\sum_{k=0}^{624} (q_k - b_k) e_k. \tag{18}$$

The scaling vector e_k is just an enforced weighting on the ciphertext distribution which the adversary can impose as he would like. If e_v is zero, then ciphertext differences with a weight arrangement according to v is ignored completely. For instance, if we only consider events in which the three last columns are zero, then we have to ignore roughly 2^{-96} ciphertexts until we receive a pair of ciphertexts with our preferred property. Thus, there is a penalty in terms of data-complexity

if we fixate on events that seldom happen, unless the cipher itself has a very unlikely probability distribution.

The matrices $\mathrm{C_{MC}}, \mathrm{C_{SR}}$ and C_{r-1} are easy to work with and thus we will now use C_{r-1} to compute the probabilities corresponding to the same event as recently investigated in [1] and [4].

3.1 Aligning With Recent Results on AES

To begin we have to define an input distribution $a = (a_0, a_1, \ldots, a_{624})$. For instance, if we assume that the input differences are non-zero and random in only the first column (remember that we are omitting the first SR-layer), we can let the 5 first indices of a correspond to

$$a_i = \binom{4}{i} \cdot \frac{(2^8-1)^i}{(2^{32}-1)}$$

where a_i for $0 \leq i < 5$ is the probability that we hit a difference with weight i in the first column given that the plaintext difference is known to be non-zero only in the first column. The corresponding output distribution then becomes $b = a \cdot C_{r-1}$. If we want to compute the probability that the output is non-zero in at least one column, we may sum over the probabilities that contributes to this case in the event of AES

$$p_{AES} = \sum_{\substack{v=(v_0,v_1,v_2,v_3) \\ \text{at least one } v_j \text{ zero}}} b_v$$

and compare this against the random case given by

$$p_{rand} = 2^{-128} \cdot \sum_{\substack{v=(v_0,v_1,v_2,v_3) \\ \text{at least one } v_j \text{ zero}}} \prod_{k=0}^{3} \binom{4}{v_k} (2^8-1)^{v_k}.$$

For instance, if we assume the above input probability distribution a (i.e. input differences are non-zero in the first column only) and the corresponding output distribution p_{AES} for $r = 5$ and $r = 6$ rounds, we get the results of Table 1. In particular, [4] arrives at $2^{-30} + 2^{-50.980}$ for the event that a ciphertext difference is zero in at least one column when the plaintext difference is active in the first column, which is identical to the result obtained via the weight transition probability matrix.

In the case that the plaintext difference has exactly one active byte in the first column, i.e. the input distribution a has probability 1 in a_1, the method based on weight transition probabilities returns a probability $2^{-30} + 2^{-51.983}$ while [1] arrives at $2^{-30} + 2^{-51.985}$.

For 6 rounds the probability in [1] for the same type of event is estimated to be $2^{-30} + 2^{73.995}$, which is also identical to the probability derived using the weight transition probability matrix. We have added the probability for 7 rounds into the table for completeness.

Table 1: *Comparison between results obtained in literature vs our results obtained using the weight transition probability (WTP) method of AES probabilities, with collision in any ciphertext column as the plaintext event (PE) and with plaintext differences restricted to either one active column or one active byte as the ciphertext event (CE).*

Rounds	PE	CE	**Probability in literature**	**WTP**	**Ref.**
5	Active Byte	Zero-column	$2^{-30} + 2^{-51.985}$	$2^{-30} + 2^{-51.983}$	[1]
5	Active Column	Zero-column	$2^{-30} + 2^{-50.980}$	$2^{-30} + 2^{-50.980}$	[4]
6	Active Column	Zero-column	$2^{-30} + 2^{-73.995}$	$2^{-30} + 2^{-73.995}$	[1]
7	Active Column	Zero-column		$2^{-30} + 2^{-126.891}$	

However, there might be better choices of input and output distributions that can be used to optimize this further. For instance, in the same setting as above and for 7 rounds, we get

$$p_{AES-7R} = 2^{-30} + 2^{-126.891}.$$

If we instead ask what the probability of getting at least one zero-byte, we get

$$p_{AES-7R} = 2^{-4} + 2^{-126.036}.$$

Note that the weight transition probability matrix verifies the well-known impossible difference probability too (i.e. you get probability zero for the event of less than 5 active columns in total when evaluated for C_3).

These results motivates a conjecture.

Conjecture 1. The weight transition probability matrix defined in Definition 4 emulates the true weight probabilities distributions in AES.

3.2 Weight Distributions Biased in Opposite Directions

Assume that we fix an output event, e.g. that at least one column in the difference is zero which happens with probability roughly 2^{-30} at random. The second type of events we could look for is the case when there exist two different input distributions, e.g. $a^1, a^2 \in R^{625}$ with $a^1_i = 1$ and $a^2_j = 1$ for $i \neq j$, such that the two output probabilities for the same event,

$$p_1 = \sum_{v=(v_0,v_1,v_2,v_3) \mid \text{ at least one } v_j \text{ zero}} b^1_v$$

and

$$p_2 = \sum_{v=(v_0,v_1,v_2,v_3) \mid \text{ at least one } v_j \text{ zero}} b^2_v,$$

move in opposite direction from random. For instance, assume

$$p_1 = p_{rand} - \epsilon_1 \tag{19}$$

and

$$p_2 = p_{rand} + \epsilon_j \tag{20}$$

thus maximizing the distance between the two same-event probabilities p_1 and p_2 instead of comparing single AES-probabilities with random. For instance, for 5-rounds we can pick two different input conditions for the weight in the first column, $u_1 = (3, 0, 0, 0)$ and $u_2 = (2, 0, 0, 0)$ such that

$$p_1 = 2^{-30} - 2^{-50.358}$$

becomes the probability for collision in any column after 5 rounds when there are exactly 3 active bytes in the input difference and

$$p_2 = 2^{-30} + 2^{-50.390}$$

for a collision when there are exactly 2 active bytes, such that the difference

$$p_2 - p_1 = 2^{-49.373}$$

is larger than if we compared a single event against random.

4 Possible Further Research

There might be several interesting directions for further research, but we mention just a few.

4.1 Linear optimization

To formally find the optimal distribution (choice of input and output events) that minimizes distinguishing complexity, one can employ a data- and computational complexity weighted linear optimization (linear programming).

4.2 Markov chains and stochastic matrices

Markov chains, stochastic matrices etc. is a very well-studied area. What can be said about these state transition matrices by employing known theory to them?

4.3 Rate of convergence

For $I, J \subset \{0, 1, 2, 3\}$, let $p^r_{I,J}$ denote the probability that the ciphertext difference is zero in columns indicated by J given that the plaintext difference is active in exactly the columns indicated by I after r rounds. The motivation in this paper has been to study how probabilities $p^r_{I,J}$ for ciphertext events J vary depending on the choice of input events I. Let $s_{I,J} = s_J$ denote the uniform probability that ciphertext differences are zero in the columns indicated by J. Then if we look at the rounded values of $\log_2(\frac{\max(p^r_{I,J}, s_J)}{(p^r_{I,J} - s_J)^2})$ for increasing r, we observe that the values become independent of the choice of input distributions (determined by I) not until $r = 10$ rounds of AES.

4.4 The effect of the s-box

The s-box layer acts as a probability 1 identity map in the transition probability matrix. However, the s-box maps differences in certain ways that might at least in theory have some effect on the probability distribution. It is not clear how large, if any, this effect is for a generic s-box.

4.5 Testing ciphers

This tool can be employed on a range of other similar ciphers which, may be an interesting study in itself, and as a simple and efficient tool to search for optimal new designs.

5 Conclusion

We have presented a weight transition probability related to SPNs that can be used to derive probabilities for collision events in AES that matches new results recently published in [1] and [4].

References

1. Bao, Z., Guo, J., List, E.: Extended expectation cryptanalysis on round-reduced aes. Cryptology ePrint Archive, Report 2019/622 (2019), https://eprint.iacr.org/2019/622
2. Daemen, J., Rijmen, V.: The design of rijndael: Aes - the advanced encryption standard. In: Springer (2002)
3. Grassi, L.: Mixture differential cryptanalysis: a new approach to distinguishers and attacks on round-reduced aes. IACR Transactions on Symmetric Cryptology **2018**(2), 133–160 (Jun 2018)
4. Grassi, L., Rechberger, C.: Rigorous analysis of truncated differentials for 5-round aes. Cryptology ePrint Archive, Report 2018/182 (2018), https://eprint.iacr.org/2018/182
5. Grassi, L., Rechberger, C., Rønjom, S.: A new structural-differential property of 5-round AES. In: Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II. pp. 289–317 (2017)