Alexander Rostovtsev
rostovtsev@ssl.stu.neva.ru
St. Petersburg State Polytechnic University

Changing probabilities of differentials and linear sums via isomorphisms of ciphers

Ciphers $y = C(x, k)$ and $\mathfrak{y} = \mathfrak{C}(\mathfrak{x}, \mathfrak{k})$ are isomorphic if there exists invertible computable in both directions map $y \leftrightarrow \mathfrak{y}$, $x \leftrightarrow \mathfrak{x}$, $k \leftrightarrow \mathfrak{k}$. Cipher is vulnerable if and only if isomorphic cipher is vulnerable. Instead of computing the key of a cipher it is sufficient to find suitable isomorphic cipher and compute its key. If φ is arbitrary substitution and T is round substitution, its conjugate $\mathfrak{T} = \varphi T \varphi^{-1}$ is cipher isomorphism. Conjugate substitutions have the same cycle type. Conjugation can be composed with affine maps.

Combining conjugation and affine equivalence, sometimes we can transform non-linear special S-box to conjugate affine substitution $\mathfrak{S}$. Usually for given S, $\mathfrak{S}$ there are many different auxiliary substitutions φ. Conjugate diffusion map and XOR operation become non-linear, but taking appropriate φ we can get large probabilities of differentials and linear sums of diffusion map and XOR.

For example AES substitution (as finite field inverting) is approximately conjugate with bit changing substitution. That conjugate substitution has differentials and linear sums of probability 1. Corresponding byte substitution φ defines non-linear conjugate diffusion map and non-linear conjugate to XOR operation with round key. Probabilities of differentials (biases of linear sums) of byte substitution of conjugate diffusion map are 8–12 times more then corresponding values of original S-box. Probabilities of differentials of conjugate XOR with the round key byte depends on the round key and can be 1 for some key bytes.

1. Introduction

Binary maps are widely used in cryptology. Block ciphers and hash functions are built as composition of binary maps.

Strength of cipher, hash function is determined as complexity of computing secret key (hash-function input) under known plaintexts and corresponding ciphertexts (hash function output). Well-known cryptanalysis methods take large number of known plaintexts/ciphertexts (differential [3] and linear [8] attacks) or few ones (algebraic attacks, based on Groebner bases [5] or agreeing/gluing methods [9]). Combination of those methods was considered in [1].

Modern ciphers resist those attacks due to special S-boxes. Let substitution S acts on set of n-bit strings and input $\mathbf{x} = (x_1, \ldots, x_n)$ to output $\mathbf{y} = (y_1, \ldots, y_n)$. If x_i, y_i are independent variables, linear over $\mathbb{F}_2$ sum $\sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n} b_i y_i$, $a_i, b_i \in \mathbb{F}_2$, takes 0 and 1 with probability 0.5. But if x_i, y_i are input and output bits, probabilities $P(0)$, $P(1)$ of 0, 1 may differ from 0.5. Difference $P(0) - 0.5$ for given linear sum is its *bias*. Addition modulo 2 does not change linear sums and their biases. Linear diffusion map change sums but keeps its biases. Bias of composition of two maps is computable is biases of the maps are known. Linear cryptanalysis is based on

search linear sums with maximal absolute result biases. Hence biases of *S*-boxes determine the complexity of linear attack. The most weak substitutions have biases ±0.5.

Let $\mathbf{x}$, $\mathbf{x}'$ — is a pair of inputs of substitution *S*, $\mathbf{y} = S(\mathbf{x})$, $\mathbf{y}' = S(\mathbf{x}')$. Let $\Delta\mathbf{x} = \mathbf{x} + \mathbf{x}'$, $\Delta\mathbf{y} = \mathbf{y} + \mathbf{y}'$ (notice that $\Delta\mathbf{y} = 0$ if and only if $\Delta\mathbf{x} = 0$). Probability of differential $(\Delta\mathbf{x}, \Delta\mathbf{y})$ may be positive. As in previous case, addition modulo 2 with a constant keeps differential and its probability, linear diffusion map changes differential but keeps its probability. Probability of composition of two maps is product of corresponding probabilities. Differential cryptanalysis is based on search result differentials that have large probabilities. So complexity of differential attack is defined by *S*-boxes. The most weak substitution has differentials that have probability 1.

Usually *S*-boxes provide both minimal probabilities of most like differentials and minimal absolute bias of most (least) like linear sums.

In this paper we generalize known attacks on block ciphers and show that special *S*-boxes have little advantage comparatively to random ones. There are no publications that defines “strong” and “weak” *S*-boxes with respect to algebraic attacks. So we illustrate proposed cryptanalysis technique only for differential and linear methods. But we believe that this approach can be used algebraic and other types of attacks.

2. Algebraic basics

Let $\mathbf{y} = T(\mathbf{x})$ is arbitrary map of set of *n*-bit strings to itself. This map can be defined using interpolating polynomials over finite field of characteristic 2. Usually normal algebraic form is used

$$\mathfrak{G}_n[\mathbf{x}] = \mathbb{F}_2[x_1, \ldots, x_n]/(x_1^2 + x_1, \ldots, x_n^2 + x_n).$$

Ring $\mathfrak{G}_n[\mathbf{x}]$ is finite and hence is Artinian, its Krull dimension is 0 [2]. Intersection of ideals coincides with their product. Each ideal can be factored as product of prime ideals. Prime ideal is maximal and consists of polynomials that take zero on given set of arguments. There exist 2^n prime ideals. Prime ideal can be given by one polynomial, for example, $1 + x_1 \ldots x_n$. Hence each ideal is principal and each polynomial is idempotent.

Set of *n*-bit binary vectors $\mathbf{x}$ form affine space $\mathbb{A}^n$. Ideal $\mathfrak{A} \subseteq \mathfrak{G}_n[\mathbf{x}]$ is determined by set of its zeroes in $\mathbb{A}^n$ and back, any subset of $\mathbb{A}^n$ is determined by corresponding ideal. Zero ideal corresponds to $\mathbb{A}^n$, ideal $\mathfrak{G}_n[\mathbf{x}] = (1)$ corresponds to empty set.

Automorphism of $\mathfrak{G}_n[\mathbf{x}]$ fixes 0 and 1 and maps prime ideal to prime ideal. Indeed, if we assume that image of prime ideal is product of different prime ideals, then it cannot have single prototype. Each permutation of prime ideals is automorphism of $\mathfrak{G}_n[\mathbf{x}]$. Since there is a bijection between $\mathbb{A}^n$ and set of prime ideals,

automorphism group of ring $\mathfrak{G}_n[\mathbf{x}]$ is isomorphic to group of all permutations of points of $\mathbb{A}^n$.

Similarly to ring $\mathfrak{G}_n[\mathbf{x}]$ may be defined ring

$$\mathfrak{G}_{2n}[\mathbf{x}, \mathbf{y}] = \mathbb{F}_2[x_1, \ldots, x_n, y_1, \ldots, y_n]/(x_1^2 + x_1, \ldots, x_n^2 + x_n, y_1^2 + y_1, \ldots, y_n^2 + y_n).$$

Affine functions are $\sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n} b_i y_i + c \in \mathfrak{G}_{2n}[\mathbf{x}, \mathbf{y}]$.

Set of $2n$-bit vectors $(\mathbf{x}, \mathbf{y})$ form affine space $\mathbb{A}^{2n}$. Subset of $\mathbb{A}^{2n}$, where any n-bit vector $\mathbf{x}$ corresponds to one n-bit vector $\mathbf{y}$, defines map of space $\mathbb{A}^n$ to itself, where $\mathbf{x}$ is input and $\mathbf{y}$ is output of the map.

Let $\mathbf{y} = T(\mathbf{x})$ is a map $\mathbb{A}^n \to \mathbb{A}^n$. Set of polynomials that take zero in points $(\mathbf{x}, \mathbf{y})$ forms ideal $\mathfrak{A}_T \subset \mathfrak{G}_{2n}[\mathbf{x}, \mathbf{y}]$ of map T. There is bijection between such maps and ideals. Ideal $\mathfrak{A}_T$ defines algebraic set $V(T)$ that has 2^n points.

Hence there exists quotient ring $\mathfrak{G}_n[T] = \mathfrak{G}_{2n}[\mathbf{x}, \mathbf{y}]/\mathfrak{A}_T$ — set of polynomials determined up to map T.

Theorem 1. For arbitrary map T: $\mathbb{A}^n \to \mathbb{A}^n$ there exists ring isomorphism $\mathfrak{G}_n[T] \cong \mathfrak{G}_n[\mathbf{x}]$.

Proof. Ideal $\mathfrak{A}_T$ consists of all polynomials from $\mathfrak{G}_{2n}[\mathbf{x}, \mathbf{y}]$ that take zero if $\mathbf{y} = T(\mathbf{x})$. This equality holds in 2^n points, each polynomial in $\mathfrak{A}_T$ takes at least 2^n zeroes. Quotient ring $\mathfrak{G}_n[T] = \mathfrak{G}_{2n}[\mathbf{x}, \mathbf{y}]/\mathfrak{A}_T$ consists of classes of polynomials that take arbitrary values in those 2^n points. Since affine space $\mathbb{A}^{2n}$ has 2^{2n} points, $\#\mathfrak{G}_n[T] = \#\mathfrak{G}_n[\mathbf{x}] = 2^n$. Ring $\mathfrak{G}_n[T]$ has characteristic 2, addition and multiplication in $\mathfrak{G}_n[T]$ corresponds to addition and multiplication in $\mathfrak{G}_n[\mathbf{x}]$. **n**

If φ, T are elements of symmetric group G, conjugation σ_φ: $T \to \varphi T \varphi^{-1}$ is automorphism of G.

Centralizer of substitution T is subgroup $C_T \subseteq G$ such that $\varphi T \varphi^{-1} = T$ for all $\varphi \in C_T$. Centralizer C_T partitions group G into cosets. If T, T_1 are conjugate, number of substitutions φ that maps $T \to T_1$ equals to $\#C_T$. Indeed, if $\psi \in C_T$, then $\psi T \psi^{-1} = T$, $\varphi T \varphi^{-1} = \varphi \psi T \psi^{-} \varphi^{-1} = (\varphi\psi) T (\varphi\psi)^{-1}$ and if ψ runs through C_T, all substitution $\varphi\psi$ are different. Hence number of substitutions conjugate to T is $\frac{\#G}{\#C_T}$. Substitutions T, T_1 are conjugate if and only if the have the same cycle type [7]. Indeed, if $T^r(x) = x$, $T_1 = \varphi T \varphi^{-1}$ and $z = \varphi(x)$, then $T_1^r(z) = (\varphi T \varphi^{-1})^r(z) = \varphi T^r \varphi^{-1}(z) = z$. Conjugate substitutions have a cycle of the same length. Applying this statement to other cycles shows that conjugate substitutions have the same cycle types. Back, if T, T_1 have cycle of the same length, then there exists corresponding substitution φ,

Affine substitution is given by equation $\mathbf{y} = L\mathbf{x} + \mathbf{c}$, where L is invertible matrix and $\mathbf{c}$ is arbitrary vector. Affine substitutions form subgroup of symmetric group. Substitutions T, T_1 are affine equivalent if there exist affine substitutions A, B such that $T_1 = ATB$. Affine equivalence of substitutions can be easily detected [4].

Affine equivalence preserves probabilities of differentials and absolute biases of linear sums, but differentials and linear sums with the fixed probabilities can change. Affine equivalent substitutions can have different cycle type.

Count number of affine n-bit substitutions. There are 2^n vectors $\mathbf{c}$. First row of matrix L can be arbitrary but non-zero $(2^n - 2^0)$ possibilities. Second row can be arbitrary but non-zero and different from the first one $(2^n - 2)$ possibilities. Third row must differ from zero, first row, second row and their sum $(2^n - 2^2)$ possibilities and so on. Hence number of affine substitutions is $2^n \prod_{i=0}^{n-1} (2^n - 2^i)$.

If substitution S_n acts on n-bit words, it can be extended to substitution S_{n+1} acting on $(n + 1)$-bit strings adjoining identity function $y_{n+1} = x_{n+1}$. Its cycle type is double cycle type of substitution S_n (one set of cycles corresponds to $x_{n+1} = 0$, other set of the same cycles corresponds to $x_{n+1} = 1$). We say that substitution S_{n+1} is equivalent to substitution S_n.

Theorem 2. Let substitution S acts on n-bit words. Next statements are equivalent.

1. Substitution S has linear sums with probability 1.
2. Ideal A_S contains affine polynomial.
3. Substitution S is affine equivalent to substitution that acts on $(n - 1)$-bit words.

Proof. 1 $\Leftrightarrow$ 2. If ideal A_S of substitution $\mathbf{y} = S(\mathbf{x})$ contains affine function $\sum a_i x_i + \sum b_i y_i + c$, this function takes zero with probability 1. Back, if function takes zero with probability 1, it is in ideal A_S.

2 $\Rightarrow$ 3. If ideal A_S contains linear function $\sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n} b_i y_i$, there exists such invertible affine change of variables that $x_n = \sum_{i=1}^{n} a_i x_i$, $y_n = \sum_{i=1}^{n} b_i y_i$. Under this change x_n can be expressed as linear combination of $x_1, \ldots, x_{n-1}$ and y_n can be expressed as linear combination of $y_1, \ldots, y_{n-1}$. Hence function $\sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n} b_i y_i$ will be changed by $y_n + x_n$, and other functions of ideal basis will depend only of $x_1, \ldots, x_{n-1}, y_1, \ldots, y_{n-1}$ and correspond to a map that acts on $(n - 1)$-bit words. Assume that this map is not invertible. Then map adjoined by identity bit cannot be invertible, so assumption is wrong.

3 $\Rightarrow$ 2. Let substitution S if affine equivalent to substitution S_1 with identity $y_n = x_n$, и $S = AS_1B$. Then affine substitution B changes x_n by affine function of $\mathbf{x} = (x_1,$

…, x_n) and affine substitution A changes y_n by affine function of $\mathbf{y} = (y_1, \ldots, y_n)$, i.e. ideal of substitution S contains affine function. ∎

In practice usually substitution has no linear sums with probability 0, 1. So we need generalization of theorem 2. Polynomial $f(\mathbf{x}, \mathbf{y}) \in \mathfrak{G}_{2n}[\mathbf{x}, \mathbf{y}]$ can be represented by its values in 2^{2n} points of affine space $\mathbb{A}^{2n}$. Non-linearity NL(f) of polynomial $f(\mathbf{x}, \mathbf{y}) \in \mathfrak{G}_{2n}[\mathbf{x}, \mathbf{y}]$ is Hamming distance between the binary vector and the set of vectors that correspond to affine polynomials.

Theorem 3. Maximal absolute bias of linear sums of substitution S equals to $\min(\mathrm{NL}(f))/2^n$ for all $f \in \mathfrak{A}_S$.

Proof follows directly from the definitions of non-linearity and of bias of linear sums. ∎

Differential of substitution can be defined using formal derivations. Input and output variables $\mathbf{x}$, $\mathbf{y}$ of ideal $\mathfrak{A}_S$ of n-bit substitution S are algebraically dependent. Let $y_1 = f_1(\mathbf{x}), \ldots, y_n = f_n(\mathbf{x})$. Choose a set of algebraically independent variables, for example, $\mathbf{x}$.

Derivative of polynomial $f(x_1, \ldots, x_n)$ for variable x_1 can be defined as difference $f'_{x_1} = f(1 + x_1, x_2, \ldots, x_n) + f(x_1, \ldots, x_n)$. This difference corresponds to point (1, 0, …, 0) of affine space $\mathbb{A}^n$. Similarly we can define derivative for arbitrary point $\mathbf{u} = (u_1, \ldots, u_n)$ as difference $f(\mathbf{x} + \mathbf{u}) + f(\mathbf{x})$.

Substitution has differential $(\Delta\mathbf{x}, \Delta\mathbf{y})$ of probability 1 if derivative map corresponding to point $\Delta\mathbf{x}$ is constant and equals to $\Delta\mathbf{y}$.

3. Isomorphic and conjugate ciphers

Definition 1. Let x, y, k are plaintext, ciphertext and key correspondingly. Isomorphism of ciphers C, $\mathfrak{C}$ is invertible computable in both directions map $x \leftrightarrow \mathfrak{x}$, $y \leftrightarrow \mathfrak{y}$, $k \leftrightarrow \mathfrak{k}$ such that equality $y = C(x, k)$ hold if and only if equality $\mathfrak{y} = \mathfrak{C}(\mathfrak{x}, \mathfrak{k})$ holds. Ciphers are isomorphic, if there exists an isomorphism between them.

Theorem 4. Cipher C is vulnerable if and only if isomorphic cipher $\mathfrak{C}$ is vulnerable.

Proof. It is sufficient to map set of plaintexts, ciphertexts of original cipher to ones of vulnerable isomorphic cipher, find key of vulnerable cipher and map it to key of original cipher. ∎

Well known example of cipher isomorphism is affine equivalence [4]. Consider other simple example of cipher isomorphism. Let T is round encryption function, k_i is round key and cipher is defined by equation

$$y = T(k_r + T(k_{r-1} + \ldots T(k_1 + x)\ldots)). \qquad (1)$$

Texts x, y and key k do not change if we apply to them $\varphi^{-1}\varphi$ map for arbitrary substitutions φ, ψ. Then (1) can be written as

$$\varphi^{-1}\varphi y = T\psi^{-1}\psi(\varphi^{-1}\varphi k_r + \varphi^{-1}\varphi T\psi^{-1}\psi(\varphi^{-1}\varphi k_{r-1} + \dots \varphi^{-1}\varphi T\psi^{-1}\psi(\varphi^{-1}\varphi k_1 + \varphi^{-1}\varphi x)\dots)).$$

Left multiplying this equation by φ gives

$$\varphi y = \varphi T\psi^{-1}\psi(\varphi^{-1}\varphi k_r + \varphi^{-1}\varphi T\psi^{-1}\psi(\varphi^{-1}\varphi k_{r-1} + \dots \varphi^{-1}\varphi T\psi^{-1}\psi(\varphi^{-1}\varphi k_1 + \varphi^{-1}\varphi x)\dots)). \quad (2)$$

Let $\mathfrak{k} = \varphi(k)$, $\mathfrak{x} = \varphi(x)$, $\mathfrak{y} = \varphi(y)$. Define isomorphic cipher with round encryption $\mathfrak{T}_{\varphi\psi} = \varphi T\psi^{-1}$, and isomorphic commutative and associative addition

$$\mathfrak{x} \,{}_{\psi}{+}_{\varphi}\, \mathfrak{k} = \psi(\varphi^{-1}(\mathfrak{x}) + \varphi^{-1}(\mathfrak{k})).$$

We can simplify equation (2) since

$$\varphi^{-1}\varphi(k_1) + \varphi^{-1}\varphi(x) = \varphi^{-1}(\mathfrak{k}_1) + \varphi^{-1}(\mathfrak{x}),$$

$$\varphi^{-1}\varphi T\psi^{-1}\psi(\varphi^{-1}(\mathfrak{k}_1) + \varphi^{-1}(\mathfrak{x})) = \varphi^{-1}\mathfrak{T}_{\varphi\psi}(\mathfrak{k}_1 \,{}_{\psi}{+}_{\varphi}\, \mathfrak{x}).$$

Isomorphic cipher satisfies equation

$$\mathfrak{y} = \mathfrak{T}_{\varphi\psi}(\mathfrak{k}_r \,{}_{\psi}{+}_{\varphi}\, \mathfrak{T}_{\varphi\psi}(\mathfrak{k}_{r-1} \,{}_{\psi}{+}_{\varphi} \dots \mathfrak{T}_{\varphi\psi}(\mathfrak{k}_1 \,{}_{\psi}{+}_{\varphi}\, \mathfrak{x})\dots)). \quad (3)$$

If round encryption function is composition of two maps $T = UV$, we can write $\mathfrak{T} = \mathfrak{V}\mathfrak{U}$, где $\mathfrak{V} = \varphi V\chi^{-1}$, $\mathfrak{U} = \chi U\psi$ for arbitrary invertible map χ. This approach can be used if round encryption function contains more then two maps.

The main idea is to find such isomorphism of ciphers that isomorphic cipher will be breakable. For example, it will have more like differentials and linear sums. This approach can be used in algebraic attacks too.

For example, invertible linear map, defined as matrix L, acts on XOR operation ${}_{L}{+}_{L}$ trivially: $L(L^{-1}\mathbf{x} + L^{-1}\mathbf{k}) = \mathbf{x} + \mathbf{k}$ because multiplying by matrix is distributive under addition.

In this paper we consider isomorphism of ciphers given by conjugation. Let φ is some substitution and $\mathfrak{T}_{\varphi} = \varphi T\varphi^{-1}$, $\mathfrak{x} \,{}_{\varphi}{+}_{\varphi}\, \mathfrak{k} = \varphi(\varphi^{-1}(\mathfrak{x}) + \varphi^{-1}(\mathfrak{k}))$.

Definition 2. Ciphers C, $\mathfrak{C}$ are conjugate if isomorphism between them is given by conjugation.

Usually S-boxes are chosen using special requirements to protect known attacks. For example, probability of most likely differentials of substitution must be small, absolute bias of linear sums must be small.

If substitution acts on binary words of even length n, there exists non-zero differential with probability $\geq 2^{-n+2}$ and non-zero linear sum with absolute bias $\geq 2^{-n+2}$ [6]. The best substitutions (with respect to differential and linear attacks) have probability of most likely differentials 2^{-n+2} and maximal absolute bias of linear sums 2^{-n+2}.

Example of such substitution is finite field inverting (input 0 corresponds to output 0). This substitution has two fixed points (0 and 1), other points form cycles of length 2. If substitution S satisfies $S(0) = 1$, $S(1) = 0$ and $S(x) = x^{-1}$ for $x \neq 0, 1$, then all cycles have length 2. Hence substitution S has conjugate affine substitution $\mathfrak{S}$: $\mathbf{y} = \mathbf{x} + \mathbf{a}$ for arbitrary non-zero $\mathbf{a}$.

Substitution S has 2^{n-1} cycles of length 2. Each cycle defines two input/output pairs of centralizer. Element φ of centralizer satisfies equation $S\varphi = \varphi S$. For cycle (0, 1) we have $S\varphi(0) = \varphi(1)$, $S\varphi(1) = \varphi(0)$. We can choose arbitrary $\varphi(1)$ (there are 2^n possibilities) and find unique $\varphi(0)$. Next cycle has $2^n - 2$ possibilities, etc. Cardinality of centralizer of substitution S is $\#C_S = 2^n(2^n - 2)(2^n - 4)\cdot\ldots\cdot 4\cdot 2 \approx \sqrt{2^n!}$.

Assume that each round has XOR operation with round key, n-bit substitution S as above and linear diffusion map L. Then there exist a large number of substitutions φ such that equation $\mathfrak{S} = \varphi S \varphi^{-1}$. Isomorphic conjugate cipher has operations ${}_{\varphi}+_{\varphi}$, affine substitution $\mathfrak{S}$ and conjugate diffusion map $\mathfrak{L} = \varphi L \varphi^{-1}$. Possibly there exists such substitution φ that both operations ${}_{\varphi}+_{\varphi}$, $\mathfrak{L}$ has high probability of differentials or large biases of linear sums. Our goal is to find such substitution φ.

Algorithm 1. Computing substitution φ.

Input: conjugate n-bit substitutions S, $\mathfrak{S}$.

Output: substitution φ.

Method.

1. Make list of marked inputs Tb_1 of substitution φ and list of marked outputs Tb_2 of substitution φ with length 2^n. Let $\mathrm{Tb}_1 = \varnothing$, $\mathrm{Tb}_2 = \varnothing$.
2. While Tb_1, Tb_2 do not contain whole set of n-bit words, do the next.
   2.1. Choose arbitrary input x that is not in Tb_1 and arbitrary output $\varphi(x)$ that is not in Tb_2.
   2.2. Compute $\mathfrak{S}\varphi(x)$.
   2.3. Compute $S(x)$ and $\varphi S(x)$ using equation $\varphi S(x) = \mathfrak{S}\varphi(x)$.
   2.4. Join x, $S(x)$ to $\mathrm{Tb}_1$and sort.
   2.5. Join $\varphi(x)$ и $\varphi S(x)$ to Tb_2 and sort.
3. Return: Tb_2. **n**

Complexity of algorithm 1 is $O(2^n)$.

If substitution φ is found, we can compute conjugate substitutions for diffusion map $\mathfrak{L} = \varphi L \varphi^{-1}$ and key XOR and find probabilities of differentials and biases of linear sums. If those probabilities are small, find next substitution φ, etc. Since affine equivalent substitutions φ have the same maximal probabilities of differentials (biases of linear sums), it is sufficient to search substitutions φ that are not equivalent. Experiment shows that differential probabilities increase if substitution φ has many fixed points.

Assume that substitution S has such cycles that LCM of their lengths does not divide number of affine substitutions. Then there is no conjugate affine substitution $\mathfrak{S}$. In this case we can apply affine equivalence to original cipher, that changes cycle type of substitution. Affine equivalent cipher can be considered as modified original cipher, we can find isomorphic cipher, apply affine equivalence again, etc. So we obtain random walk in the set of conjugate and affine equivalent ciphers.

Search of substitution φ can be sensible by special choosing x, $\varphi(x)$ on step 2.1 of algorithm 1. We need to obtain small non-linearity of a polynomial in ideal of some substitution.

4. Incrementing probabilities of differentials and linear sums: a toy example

Show that conjugate cipher can have more likely differentials, linear sums then the original cipher. Consider a toy cipher with two operations: XOR and 4 bit substitution T in each round.

$$T = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 1 & 0 & 9 & 6 & 13 & 7 & 3 & 5 & 15 & 2 & 12 & 14 & 10 & 4 & 11 & 8 \end{pmatrix}.$$

Substitution T corresponds to map $x \to x^{15}$ (nod 17) for group $\mathbb{F}_{17}^{*}$ if $x \neq 0, 1$. All cycles have length 2. This substitution has the best possible differential probabilities and biases of linear sums equal to 0.25. Tables of differentials, linear sums are the next.

{16, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
{0, 2, 2, 2, 0, 2, 2, 2, 0, 0, 2, 0, 0, 0, 2, 0},
{0, 2, 2, 2, 4, 0, 2, 0, 2, 0, 0, 0, 0, 0, 2, 0},
{0, 2, 2, 0, 0, 0, 2, 2, 2, 2, 0, 0, 2, 0, 0, 2},
{0, 0, 4, 0, 0, 0, 2, 2, 0, 0, 0, 4, 2, 2, 0, 0},
{0, 2, 0, 0, 0, 4, 0, 2, 2, 2, 2, 0, 0, 2, 0, 0},
{0, 2, 2, 2, 2, 0, 0, 0, 2, 2, 2, 0, 0, 0, 0, 2},
{0, 2, 0, 2, 2, 2, 0, 0, 0, 2, 2, 4, 0, 0, 0, 0},
{0, 0, 0, 0, 4, 0, 0, 4, 0, 2, 0, 2, 0, 2, 0, 2},
{0, 2, 0, 0, 0, 2, 2, 2, 0, 2, 0, 0, 2, 0, 4, 0},
{0, 0, 0, 2, 0, 2, 2, 2, 0, 2, 2, 2, 0, 0, 0, 2},
{0, 0, 2, 2, 0, 2, 2, 0, 4, 0, 0, 0, 0, 2, 0, 2},
{0, 0, 0, 2, 0, 0, 2, 0, 2, 2, 0, 2, 4, 0, 0, 2},
{0, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 4, 2, 2, 0},
{0, 0, 0, 0, 2, 2, 0, 0, 2, 0, 0, 2, 2, 4, 2, 0},
{0, 0, 0, 2, 2, 0, 0, 0, 0, 0, 2, 0, 0, 2, 4, 4}

{8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
{0, -2, -2, -4, 2, 0, 0, -2, -2, 4, 0, -2, 0, -2, 2, 0},
{0, 2, 2, 0, 0, -2, -2, 0, 2, 0, 0, 2, 2, -4, 4, 2},
{0, 0, 0, -4, -2, 2, 2, 2, 0, -4, 0, 0, -2, -2, 2, -2},
{0, 2, 0, -2, 0, -2, 4, -2, 4, 2, 0, 2, 0, 2, 0, -2},
{0, 0, 2, -2, 2, -2, 0, 0, 2, -2, -4, -4, 0, 0, -2, 2},
{0, 0, 2, 2, 0, 0, 2, 2, 2, 2, 4, -4, -2, -2, 0, 0},
{0, -2, -4, 2, -2, -4, 2, 0, 0, -2, 0, -2, 2, 0, 2, 0},
{0, 2, 2, 0, 4, -2, 2, 0, -4, -2, 2, 0, 0, 2, 2, 0},
{0, -4, 0, 0, 2, -2, -2, -2, 2, -2, 2, 2, -4, 0, 0, 0},

{0, 0, 0, 0, 0, 4, 0, -4, 2, -2, 2, -2, 2, 2, 2, 2},
{0, 2, -2, 0, 2, 0, 0, -2, 0, -2, 2, 0, 2, -4, -4, -2},
{0, -4, 2, -2, 0, 0, 2, 2, 0, 0, 2, 2, 4, 0, -2, 2},
{0, -2, 4, 2, -2, 0, 2, -4, -2, 0, -2, 0, 0, -2, 0, -2},
{0, -2, 0, 2, 4, 2, 0, 2, 2, 0, -2, 0, 2, 0, 2, -4},
{0, 0, 2, -2, -2, -2, -4, 0, 0, 0, 2, -2, 2, 2, 0, -4}

Probability of differential (bias of linear sum) is corresponding number in the table divided by 16.

Choose conjugate substitution $\mathfrak{T}$ as inversion of the least bit. Compute substitution φ using algorithm 1:

$$\varphi = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 0 & 1 & 2 & 5 & 7 & 9 & 4 & 8 & 11 & 3 & 13 & 15 & 12 & 6 & 14 & 10 \end{pmatrix}.$$

Substitution $\mathfrak{T}$ does not change input differences and liner sums. Probability of differential is 1, bias of linear sum is ±0.5. Conjugate XOR operation $\mathfrak{x} +_\varphi \mathfrak{k}$ becomes non-linear, for fixed round key probability of differential (bias of linear sum) depends on the round key. If $k = 0$, $\varphi^{-1}(\mathfrak{k}) = 0$, then $\mathfrak{x} +_\varphi \mathfrak{k} = \varphi(\varphi^{-1}(\mathfrak{x}) + \varphi^{-1}(\mathfrak{k})) = \mathfrak{x}$, conjugate differentials and linear sums have probability 1. Some other keys have differentials with probability 1 and linear sums with bias ±0.5. For example key 11 gives such differentials and linear sums:

{16, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
{0, 4, 0, 0, 0, 4, 0, 0, 4, 0, 0, 0, 4, 0, 0, 0},
{0, 0, 4, 4, 0, 0, 4, 4, 0, 0, 0, 0, 0, 0, 0, 0},
{0, 0, 4, 0, 0, 0, 4, 0, 0, 0, 4, 0, 0, 0, 4, 0},
{0, 0, 0, 0, 4, 4, 0, 0, 0, 0, 0, 0, 4, 4, 0, 0},
{0, 4, 0, 0, 4, 0, 0, 0, 4, 0, 0, 0, 0, 4, 0, 0},
{0, 0, 4, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 4},
{0, 0, 4, 0, 0, 0, 0, 4, 0, 0, 4, 0, 0, 0, 0, 4},
{0, 4, 0, 0, 0, 4, 0, 0, 0, 4, 0, 0, 0, 4, 0, 0},
{0, 0, 0, 0, 0, 0, 0, 0, 4, 4, 0, 0, 4, 4, 0, 0},
{0, 0, 0, 4, 0, 0, 0, 4, 0, 0, 4, 0, 0, 0, 4, 0},
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 16, 0, 0, 0, 0, 0},
{0, 4, 0, 0, 4, 0, 0, 0, 0, 4, 0, 0, 4, 0, 0, 0},
{0, 0, 0, 0, 4, 4, 0, 0, 4, 4, 0, 0, 0, 0, 0, 0},
{0, 0, 0, 4, 0, 0, 4, 0, 0, 0, 4, 0, 0, 0, 0, 4},
{0, 0, 0, 0, 0, 0, 4, 4, 0, 0, 0, 0, 0, 0, 4, 4}

{8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
{0, 0, 0, 0, 0, 0, 0, 0, -4, 0, 0, 4, -4, 0, 0, -4},
{0, 0, -8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
{0, 0, 0, 0, 0, 0, 0, 0, -4, 4, 0, 0, 4, 4, 0},
{0, 0, 0, 0, 4, 0, 0, 4, 0, 0, 0, 0, 0, -4, 4, 0},
{0, 0, 0, 0, 0, 4, -4, 0, -4, 0, 0, -4, 0, 0, 0, 0},
{0, 0, 0, 0, 0, -4, -4, 0, 0, 0, 0, 0, -4, 0, 0, 4},
{0, 0, 0, 0, 4, 0, 0, -4, 0, 4, 4, 0, 0, 0, 0, 0},
{0, -4, 0, 0, 0, -4, 0, 0, -4, 0, 0, 0, 4, 0, 0, 0},
{0, 0, 0, -4, 0, 0, 0, 4, 0, 4, 0, 0, 0, 4, 0, 0},
{0, 0, 0, 4, 0, 0, 0, 4, 0, 0, 4, 0, 0, 0, -4, 0},
{0, 4, 0, 0, 0, -4, 0, 0, 0, 0, 0, -4, 0, 0, 0, -4},
{0, -4, 0, 0, 0, 0, -4, 0, 4, 0, 0, 0, 0, 0, 0, -4},
{0, 0, 0, 4, -4, 0, 0, 0, 0, 4, 0, 0, 0, 0, 4, 0},
{0, 0, 0, 4, 4, 0, 0, 0, 0, 0, -4, 0, 0, 4, 0, 0},
{0, -4, 0, 0, 0, 0, 4, 0, 0, 0, 0, -4, -4, 0, 0, 0}

Since probability of differential, linear sum strongly depends on the round key, it is useful to modify technique of differential and linear cryptanalysis. Compute table of differences and linear sums as functions of round key. Obtained frequency of differential can help to find the key if the table of corresponding probabilities of differentials is known.

5. Application to AES

Apply proposed technique to encryption standard AES.

5.1. AES and its conjugate image

AES cipher has 10, 12 or 14 rounds. Block length is 128 bits, key length is 128, 192 or 256 bits. Each round has next operations.

1. XOR with round key.
2. Block is partitioned in 16 bytes. Each byte is changed according to substitution code. Substitution is defined as composition of inversion in field $\mathbb{F}_{256} = \mathbb{F}_2[t]/(t^8 + t^4 + t^3 + t + 1)$ and affine byte substitution $\mathbf{y} = M\mathbf{x} + \mathbf{c}$, where
$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$
3. Linear diffusion map (shift rows and mix columns) can be considered as matrix W with size 16*16 over $\mathbb{F}_{256}$, that acts on the set of 16 element vectors.

$$W = \begin{pmatrix} t & 0 & 0 & 0 & 0 & 1+t & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & t & 0 & 0 & 0 & 0 & 1+t & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & t & 0 & 0 & 0 & 0 & 1+t \\ 1+t & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & t \\ 0 & 0 & 0 & 1 & t & 0 & 0 & 0 & 0 & 1+t & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & t & 0 & 0 & 0 & 0 & 1+t & 0 \\ 0 & 0 & 0 & 1+t & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & t & 0 \\ 0 & 0 & 0 & t & 1+t & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & t & 0 & 0 & 0 & 0 & 1+t & 0 & 0 \\ 0 & 0 & 1+t & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & t & 0 & 0 \\ 0 & 0 & t & 0 & 0 & 0 & 0 & 1+t & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & t & 1+t & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1+t & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & t & 0 & 0 & 0 \\ 0 & t & 0 & 0 & 0 & 0 & 1+t & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & t & 0 & 0 & 0 & 0 & 1+t & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & t & 1+t & 0 & 0 & 0 \end{pmatrix}$$

Each byte can be represented as binary vector in basis $[1, t, t^2, \ldots, t^7]$. Then elements 0, 1, t, $1 + t$ of matrix W correspond to block matrices: zero 0, identity E,

$L_t = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$ and $E + L_t$. Notice that $L_t^8 + L_t^4 + L_t^3 + L_t + E$ is zero matrix, matrix polynomials of L_t form finite field $\mathbb{F}_{256}$. So we can consider diffusion map W as block matrix over $\mathbb{F}_2$ with four types of blocks.

AES substitution has next cycle type: {87, 81, 59, 27, 2}. LCM of cycle lengths does not divide the group order of byte affine substitutions. Hence there is no affine substitution conjugate to the AES substitution.

For convenience we consider affine byte substitution $\mathbf{y} = M\mathbf{x} + \mathbf{c}$ as a part of diffusion map W (there are 16 equal affine substitutions in the 16 byte diffusion map). This changes diffusion map W. Zero block 0 of matrix W will stay zero due to equality $0(M\mathbf{x} + \mathbf{c}) = 0$. Identity block E of matrix W becomes affine byte substitution: $E(M\mathbf{x} + \mathbf{c}) = M\mathbf{x} + \mathbf{c}$. Block L_t of matrix W becomes affine byte substitution $L_t(M\mathbf{x} + \mathbf{c}) = L_tM\mathbf{x} + L_t\mathbf{c}$. Block $E + L_t$ of matrix W becomes affine byte substitution $(E + L_t)(M\mathbf{x} + \mathbf{c}) = (M + L_tM)\mathbf{x} + (E + L_t)\mathbf{c}$. Denote those three non-zero affine substitutions as M_1, M_t, M_{1+t}.

Then AES substitution becomes finite field inverting U. Substitution U has two fixed points (0 and 1), other cycles have length 2.

Define substitution $V(x) = U(x)$ if $x \neq 0, 1$, $V(0) = 1$, $V(1) = 0$. All cycles of substitution V have length 2 and $U(x) = V(x)$ with probability $1 - 2^{-7}$.

V= {1, 0, 137, 246, 199, 82, 119, 209, 228, 75, 37, 192, 176, 225, 229, 199, 116, 180, 166, 71, 149, 39, 96, 91, 84, 59, 249, 200, 251, 64, 234, 178, 54, 106, 86, 241, 85, 73, 164, 197, 193, 6, 148, 21, 48, 68, 162, 194, 40, 69, 146, 104, 243, 53, 102, 66, 242, 53, 32, 107, 119, 183, 85, 21, 25, 250, 55, 103, 41, 49, 245, 101, 167, 100, 167, 19, 84, 37, 229, 5, 233, 88, 5, 198, 72, 36, 135, 187, 20, 58, 34, 240, 81, 232, 97, 23, 22, 90, 171, 211, 69, 166, 54, 67, 244, 71, 145, 219, 51, 147, 33, 55, 117, 183, 151, 133, 16, 181, 182, 56, 182, 112, 208, 6, 161, 246, 129, 130, 131, 122, 123, 128, 150, 115, 186, 86, 151, 154, 149, 213, 247, 2, 181, 164, 218, 102, 50, 105, 212, 134, 132, 114, 38, 20, 155, 132, 245, 216, 133, 150, 247, 120, 42, 195, 139, 180, 101, 68, 38, 196, 18, 70, 202, 231, 210, 98, 8, 224, 27, 235, 17, 117, 116, 113, 165, 138, 118, 57, 185, 184, 134, 87, 7, 36, 43, 163, 214, 212, 228, 11, 165, 39, 83, 4, 23, 248, 168, 230, 118, 7, 170, 99, 197, 215, 226, 230, 148, 135, 196, 213, 153, 244, 144, 103, 177, 9, 214, 231, 198, 10, 203, 169, 4, 74, 215, 227, 89, 80, 26, 179, 87, 35, 52, 52, 100, 70, 3, 136, 217, 152, 121, 160, 201, 22, 65, 24}.

Hence substitution V has conjugate image $\mathfrak{V}$ defined as inverting of the least bit. Its differentials have probability 1 and linear sums have biases ±0.5. Centralizer of substitution V has cardinality $\prod_{i=0}^{127}(256-2i) = 1.3 \cdot 10^{254}$.

Auxiliary substitution φ acts on bytes. There is a large number of substitutions φ and hence a large number of conjugate ciphers with given image $\mathfrak{V}$ of substitution. Denote conjugate AES as $\mathfrak{AES}_\varphi$. $\mathfrak{AES}_\varphi$ has non-linear diffusion map and non-linear addition with round key. Block M_1 becomes to $\mathfrak{M}_1 = \varphi M_1 \varphi^{-1}$, block M_t becomes to $\mathfrak{M}_t = \varphi M_t \varphi^{-1}$, block M_{1+t} becomes to $\mathfrak{M}_{1+t} = \varphi M_{1+t} \varphi^{-1}$.

5.2. Differentials and linear sums of conjugate AES

Our goal is to choose such auxiliary substitution φ that conjugate diffusion map and conjugate XOR will have large probabilities of differentials and linear sums. There are different ways to computing suitable substitution φ. It can have many fixed points; it can have many points in which it commutes with substitutions M_1, M_t, M_{1+t}; it can be chosen so that ideals of substitutions $\mathfrak{M}_1$, $\mathfrak{M}_t$, $\mathfrak{M}_{1+t}$ contain polynomials with small non-linearity, etc.

If substitution φ has many fixed points, we choose $\varphi(x) = x$ for all possible x in the step 2.1 of algorithm 1. In this case probabilities of conjugate XOR operation tends to increase. If substitution φ commutes with substitution M_1 in many points, we choose arbitrary x, compute $M_1(x)$ and if it is possible set $\varphi(M_1(x)) = M_1(\varphi(x))$ in the step 2.1 of algorithm 1. In this case maximal bias of linear sums and probabilities of differential of substitution $\mathfrak{M}_1$ tends to increase.

Choose substitution φ using algorithm 1 so that φ has many fixed points (we did no significant optimization of φ).

φ = {0, 1, 2, 247, 4, 83, 6, 208, 8, 85, 10, 193, 12, 224, 14, 89, 16, 181, 18, 97, 20, 151, 22, 153, 24, 171, 26, 205, 28, 65, 30, 179, 32, 111, 34, 240, 36, 229, 38, 200, 40, 11, 42, 150, 44, 69, 46, 195, 45, 49, 50, 109, 52, 233, 54, 67, 56, 232, 33, 59, 60, 186, 62, 170, 64, 255, 66, 102, 68, 48, 70, 243, 72, 101, 74, 96, 76, 228, 78, 84, 80, 93, 82, 203, 77, 37, 86, 190, 25, 63, 35, 91, 92, 237, 94, 152, 23, 95, 98, 210, 100, 167, 55, 103, 104, 242, 106, 222, 108, 146, 110, 58, 112, 182, 114, 245, 17, 117, 118, 61, 120, 113, 122, 7, 124, 251, 126, 131, 128, 127, 130, 129, 132, 244, 134, 87, 136, 159, 138, 216, 140, 3, 142, 165, 144, 107, 51, 147, 148, 139, 133, 115, 43, 21, 154, 137, 156, 221, 158, 155, 160, 125, 47, 163, 164, 185, 166, 73, 39, 169, 19, 75, 172, 230, 174, 99, 13, 177, 178, 238, 180, 116, 121, 183, 184, 143, 119, 187, 188, 189, 135, 191, 192, 41, 194, 162, 196, 213, 198, 88, 168, 201, 202, 5, 204, 253, 173, 207, 123, 209, 175, 211, 212, 218, 214, 235, 149, 217, 197, 219, 220, 249, 145, 223, 176, 225, 215, 227, 199, 15, 206, 231, 9, 79, 234, 226, 236, 81, 31, 239, 90, 241, 57, 53, 105, 71, 246, 141, 248, 157, 250, 161, 252, 27, 254, 29}.

Probabilities of differentials and biases of linear sums of conjugate diffusion map become large. For example, substitution $\mathfrak{M}_1$ has most likely differential (149, 2) with probability 32/256 = 0.125 — it is 8 times more then differential probabilities of original AES. Linear sum (47, 107) has bias –46/256 = –0.18, (it is 11.5 times more then most likely biases in original AES).

Block $\mathfrak{M}_t$ has the most likely differential (13, 67) with probability 34/256 = 0.133. Linear sum (202, 126) has bias –52/256 = –0,2.

Block $\mathfrak{M}_{1+t}$ has the most likely differential (59, 63) with probability 34/256 = 0.133. Linear sum (40, 90) has bias 44/256 = 0.17.

Conjugate image of XOR ($_\varphi{+}_\varphi$) becomes non-linear substitution. Hence probabilities of differentials and linear sums depend on the key byte $\mathfrak{k} = \varphi(k)$. If $\mathfrak{k} = k = 0$, differentials and linear sums have probability 1. If key byte is non-zero, its differentials and linear sums have significant probabilities. For example, $\mathfrak{k}$ = (0, 0, 0, 0, 1, 0, 1, 0) gives the most likely differential with probability 90/256 = 0.35, absolute bias of linear sum is 68/256 = 0.26.

For key bytes $\mathfrak{k}$ in range (0, …, 255) probabilities of most likely differentials, positive and negative biases of liner sums are in the next tables (probability, bias equals to the number in corresponding position divided by 256).

{256, 10, 82, 48, 78, 42, 86, 38, 80, 74, 90, 30, 82, 72, 80, 38, 82, 80, 84, 74, 86, 42, 80, 82, 74, 82, 70, 50, 78, 50, 78, 70, 76, 90, 76, 80, 66, 38, 76, 78, 82, 48, 78, 68, 78, 82, 84, 68, 36, 30, 74, 82, 74, 40, 76, 80, 82, 72, 36, 26, 84, 38, 82, 36, 76, 40, 80, 40, 76, 34, 74, 42, 80, 46, 80, 46, 82, 70, 78, 46, 78, 46, 74, 34, 38, 36, 82, 34, 50, 40, 68, 32, 84, 38, 82, 32, 32, 28, 80, 40, 80, 34, 38, 32, 76, 72, 78, 44, 76, 40, 80, 36, 82, 30, 82, 44, 46, 38, 78, 76, 82, 70, 74, 76, 74, 56, 72, 30, 82, 42, 70, 42, 84, 70, 78, 76, 72, 46, 76, 42, 68, 50, 70, 40, 68, 72, 38, 32, 74, 72, 36, 32, 36, 32, 70, 52, 64, 44, 60, 42, 62, 50, 46, 44, 66, 42, 68, 34, 72, 34, 30, 38, 72, 74, 72, 68, 74, 52, 66, 42, 70, 36, 32, 42, 72, 46, 30, 36, 82, 40, 34, 40, 66, 24, 72, 30, 70, 70, 62, 70, 40, 40, 76, 42, 70, 44, 70, 40, 28, 40, 34, 44, 66, 42, 74, 80, 36, 40, 46, 42, 72, 48, 36, 40, 38, 38, 38, 36, 38, 48, 46, 42, 36, 40, 74, 42, 72, 36, 46, 42, 40, 48, 36, 38, 34, 32, 64, 30, 70, 38, 66, 48, 62, 48, 68}.

{128, 28, 64, 44, 72, 40, 64, 44, 60, 56, 64, 36, 64, 60, 56, 44, 64, 64, 72, 60, 64, 44, 60, 64, 56, 64, 64, 48, 56, 48, 56, 52, 60, 60, 56, 56, 52, 40, 56, 60, 64, 48, 60, 56, 60, 60, 72, 52, 40, 40, 56, 64, 60, 44, 56, 72, 64, 60, 36, 32, 64, 40, 56, 32, 64, 44, 60, 40, 64, 48, 68, 40, 60, 40, 60, 40, 68, 64, 60, 40, 60, 48, 60, 44, 40, 36, 60, 44, 48, 44, 52, 40, 60, 36, 72, 36, 40, 30, 60, 44, 60, 36, 40, 40, 52, 56, 68, 48, 60, 40, 60, 38, 60, 36, 68, 36, 40, 48, 64, 64, 64, 56, 64, 60, 56, 48, 56, 40, 72, 44, 60, 44, 60, 60, 60, 56, 60, 44, 56, 44, 56, 44, 56, 40, 60, 52, 40, 44, 60, 60, 40, 40, 44, 48, 56, 60, 56, 44, 52, 44, 64, 56, 40, 48, 52, 40, 56, 40, 56, 44, 32, 36, 60, 60, 56, 72, 60, 48, 56, 40, 56, 36, 40, 44, 60, 44, 36, 40, 60, 44, 48, 38, 56, 36, 60, 44, 56, 56, 56, 56, 40, 44, 56, 44, 64, 44, 56, 48, 40, 40, 40, 44, 60, 44, 56, 60, 44, 40, 44, 40, 60, 44, 44, 44, 40, 44, 36, 40, 44, 40, 44, 40, 36, 44, 56, 48, 52, 40, 44, 40, 48, 44, 40, 40, 48, 36, 56, 36, 64, 40, 48, 40, 48, 44, 60};

{0, -28, -68, -52, -60, -44, -72, -44, -64, -60, -68, -36, -60, -56, -68, -48, -60, -60, -60, -56, -64, -44, -60, -60, -56, -60, -60, -48, -60, -44, -60, -52, -60, -64, -60, -64, -60, -44, -60, -60, -64, -52, -68, -56, -64, -60, -72, -56, -40, -40, -64, -68, -60, -48, -60, -68, -72, -60, -48, -40, -64, -40, -60, -36, -68, -44, -64, -44, -60, -36, -60, -56, -60, -40, -72, -48, -56, -56, -60, -48, -56, -44, -64, -44, -44, -36, -60, -44, -48, -48, -56, -40, -60, -40, -64, -40, -44, -40, -64, -44, -64, -40, -44, -40, -64, -52, -60, -44, -64, -40, -68, -48, -60, -36, -64, -48, -44, -40, -68, -56, -60, -60, -60, -60, -64, -52, -60, -44, -64, -52, -60, -44, -64, -60, -64, -68, -56, -44, -60, -52, -52, -44, -56, -40, -56, -56, -40, -40, -60, -56, -40, -36, -40, -48, -60, -52, -56, -48, -60, -56, -52, -44, -44, -44, -60, -44, -56, -36, -60, -44, -40, -44, -56, -64, -60, -52, -60, -56, -56, -44, -56, -40, -44, -44, -64, -48, -40, -44, -60, -44, -36, -48, -56, -40, -60, -36, -60, -60, -52, -60, -44, -40, -56, -44, -60, -48, -64, -48, -40, -48, -40, -56, -52, -44, -64, -64, -44, -48, -52, -44, -52, -56, -40, -48, -48, -44, -40, -48, -36, -48, -48, -44, -44, -52, -56, -52, -52, -36, -52, -44, -48, -48, -40, -48, -44, -36, -56, -40, -64, -52, -52, -48, -56, -56, -56}.

Probability of round differential of original AES is defined by the number m of active S-boxes ($m \geq 1$). Usually probability of round differential cannot exceed 2^{-6m}. If we go from AES to $\mathfrak{AES}_\varphi$, then $\mathfrak{S}$-boxes becomes affine and probability of round differential depends on 16 non-linear conjugate XOR and at least 4 non-linear substitutions $\mathfrak{M}_1$, $\mathfrak{M}_t$, $\mathfrak{M}_{1+t}$. Truncated differentials can reduce number of active conjugate XOR.

Proposed approach increments probabilities of differentials, linear sums for individual substitutions comparatively to original AES, but it also increments the number of active non-linear substitutions. We cannot state that $\mathfrak{AES}_\varphi$ has more likely round differentials (linear sums) than AES. But this approach seems to be a useful tool for cryptanalysis.

If original cipher has special S-boxes or random ones, then corresponding conjugate ciphers have approximately same probabilities of most likely differentials,

because number of possible auxiliary substitutions φ is very large. So we can assume that use of special *S*-boxes gives little advantage comparatively to random ones with respect to given attack.

Using this approach can try to reduce complexity of Groebner basis algebraic attacks. For example, we can choose auxiliary substitution φ to obtain more suitable equations that define round encryption of conjugate cipher.

References

1. M. Albrecht and C. Cid. Algebraic techniques in differential cryptanalysis. Cryptology e-print archive, report 2008/177, 2008 // Available at http: // e-print.iacr.org/2008/177.
2. M. Atiyah and L. Macdonald. Introduction to commutative algebra, Addison-Wesley, 1969.
3. E. Biham and A. Shamir. Differential cryptanalysis of DES-like cryptosystems // Advances in Cryptology — CRYPTO ′90. LNCS, v. 537, Springer-Verlag, 1991, pp. 2–21.
4. A. Biryukov, C. De Canniere, A. Braeken, and B. Preneel. A toolbox for cryptanalysis: linear and affine equivalence algorithms // Advances in Cryptology — EUROCRYPT 2003. LNCS, v. 2556, Springer–Verlag, 2003, pp. 33–50.
5. J.-C. Faugere. Groebner bases. Applications in cryptology. Invited talk at FSE-07 in Luxemburg. Available at http://fse2007.uni.lu/slides/faugere.
6. H. Heyes and S. Tavares. Substitution-permutation networks resistant to differential and linear cryptanalysis // Journal of cryptology, 1996, v. 9, pp. 1–19.
7. M. Kargapolov and Y. Merzliakov. Fundamentals of the theory of groups, Springer-Verlag, N.Y., 1979.
8. M. Matsui. Linear cryptanalysis method for DES cipher. // Advances in Cryptology — EUROCRYPT ′93, LNCS, v. 765, 1994, pp. 386–397.
9. H. Raddum and I. Semaev. New technique for solving sparse equation systems. Cryptology e-print archive, report 2006/475, 2006 // Available at http: // e-print.iacr.org/2006/475.