Rostovtsev A.G., Shemyakina O.V.,
St. Petersburg State Polytechnic University

AES side channel attacks protection using random isomorphisms

General method of side-channel attacks protection, based on random cipher isomorphisms is presented. Isomorphic ciphers produce common outputs for common inputs. Cipher isomorphisms can be changed independently on transmitting and receiving sides. Two methods of RIJNDAEL protection are considered. The first one is based on random commutative isomorphisms of underlying structure. The set of field $\mathbb{F}_{256}$ isomorphisms consists of 30 subsets; each of them has 8 commutative elements presented as Galois group elements. This allows increasing the strength with respect to side channel attacks about 32 times, the encryption ratio decreases slightly. This method has comparatively small efficiency.

The second method is based on cipher byte affine isomorphisms $\sigma(\mathbf{x}) = L\mathbf{x} + \mathbf{a}$, and allows in practice eliminate side-channel attacks. The rate of this method is approximately the same as in previous case. The most convenient affine isomorphisms are involutions. Method of such affine isomorphisms generation is presented.

Key words: AES, block ciphers, finite field, random isomorphism, side-channel attack.

1. Side channel attacks and random isomorphisms

Information system is secure if it can resist adversary attacks. The set of possible attacks is determined by adversary model — the set of its possibilities. Adversary models can be ordered as sets. Model *A* is larger then model *B*, if set of possibilities of *B* is subset of possibilities of *A*. The larger is adversary model, the larger is the set of possible attacks. Adversary model ordering induces dual ordering on the set of secure systems. Each information system is characterized by maximal adversary model, under which this system is secure. Each adversary model induces minimal set of protection mechanisms that provides security of information system. For example, if adversary model is empty, all information systems are secure. If adversary possesses extrasensory possibilities and can guess all secrets, there are no secure systems.

Traditional cryptanalytic attacks use mathematical, computational and cryptanalytic possibilities. But adversary can also use laboratory possibilities. Such attack was implemented by Intelligence service against French encryption apparatus about 50 years ago [1]. Adversary can detect and recognize signals of different nature, which appear while cryptographic device processes secret information (intermediate texts during encryption).

Side channel attacks include set of attacks based on timing analysis, analysis of instant supplying power, electromagnetic and acoustic signals that carry information on a secret key [2, 3]. Side channel attacks are often more effective then well-known differential [4] or linear [5] attacks. For instance, such attack allows easy computing the secret key of mobile telephone [6].

To prevent side channel attacks users are to change secret keys periodically. This stipulates inconvenience in large communication systems. The goal of this paper is key lifetime magnification due to protecting side channel attacks.

Universal approach to protection the side channel attacks is based on random isomorphisms of cryptographic algorithms [7]. Side channel attack can be successful only if adversary knows the encryption algorithm. But if he knows input and output of encryption algorithm and does not know the method, he cannot compute secret information (the key).

Definition 1. Algorithms A and A' are isomorphic if they produce common outputs for common inputs. The computable invertible map between isomorphic algorithms inputs, outputs and methods is algorithm isomorphism.

There exist physical and mathematical type isomorphisms of algorithms. Physical isomorphisms use random permutation of processed words, empty commands, varying the clock generator frequency and so on. Mathematical isomorphisms use isomorphisms of underlying algebraic structure that determines the algorithm.

For example, elliptic curve digital signature generation according to ECDSA is described in terms of group $\mathbb{Z}/r\mathbb{Z}$, where r is prime group order. Multiplication by $m \neq 0$ is automorphism of group $\mathbb{Z}/r\mathbb{Z}$. Side channel attack protection can be based on random isomorphisms of this group [7].

Symmetric ciphers are usually described in terms of Boolean functions normal algebraic form defined as quotient ring

$$\boldsymbol{G}_n \cong \mathbb{F}_2[x_1, \ldots, x_n]/(x_1^2 + x_1, \ldots, x_n^2 + x_n).$$

Each element $f \in \boldsymbol{G}_n$ satisfies equation $f^2 + f = 0$ and hence divides zero. The only invertible element in $\boldsymbol{G}_n$ is 1. Indeed if we assume that $fg = 1$, then $f = f^2g = fg$ and $g = fg^2 = fg$, so $f = g = 1$.

Definition 2. Element $p \in \boldsymbol{G}_n$ is irreducible if its possible factorizations are only $p = 1 \cdot p$ or $p = p \cdot p$.

Theorem 3. Ring $\boldsymbol{G}_n$ is isomorphic to ring V of 2^n-dimentional binary vectors with item addition and multiplication modulo 2.

Proof. Each Boolean function f of n variables can be unequally determined both as binary vector $\mathbf{f} \in V$ and as polynomial $f \in \boldsymbol{G}_n$. Hence there is bijection between V and $\boldsymbol{G}_n$. Define vector addition and multiplication in V as bit-wise XOR and & operations. Then V becomes a commutative ring with zero $\mathbf{0} = (0..., 0)$ and unit $\mathbf{1} = (1, ..., 1)$. Let two Boolean functions are represented both as vectors $\mathbf{f}, \mathbf{g} \in V$ and as polynomials $f, g \in \boldsymbol{G}_n$. Then vector $\mathbf{f} + \mathbf{g}$ corresponds to polynomial $f + g$, and vector $\mathbf{fg}$ corresponds to polynomial fg. Hence rings V and $\boldsymbol{G}_n$ are isomorphic. **n**

This isomorphism is representation of polynomial (as the Boolean function) by vector of its binary values. There are 2^n irreducible elements in V, that contain only one zero coordinate. Hence there are 2^n irreducible polynomials in $\boldsymbol{G}_n$.

Product of two different irreducible vectors in V has two zero coordinates, product of r different irreducible vectors has r zero coordinates. It is easy to see that ring V and hence ring $\boldsymbol{G}_n$ has unique factorization.

Theorem 4. Any automorphism of ring $\boldsymbol{G}_n$ maps irreducible polynomial to an irreducible polynomial.

Proof. Let p_i, p_j be irreducible polynomials and T — permutation defined over set of irreducible polynomials such that $T(p_i) = p_k$. Then $T(p_ip_j) = T(p_i)T(p_j)$, $T(p_i + p_j) = T(p_i) + T(p_j)$. Due to unique factorization property we have $T(fg) = T(f)T(g)$ for any $f, g \in \boldsymbol{G}_n$. Permutation T can be represented as $2^n \times 2^n$ matrix over $\mathbb{F}_2$, so $T(f + g) = T(f) + T(g)$. Assume that φ is isomorphism, p is irreducible polynomial and $\varphi(p) = gh$, where g, h are two different non-constant polynomials. Then $\varphi^{-1}(gh) = = \varphi^{-1}(g)\varphi^{-1}(h) \neq p$ because binary vector in V, corresponding to $\varphi^{-1}(g)\varphi^{-1}(h)$, has at least two zero coordinates. **n**

Theorem 5. Set of automorphisms of ring $\boldsymbol{G}_n$ coincides with the set of permutation of irreducible polynomials.

Proof. According to theorem 1 any such permutation is isomorphism. Assume that there exists an automorphism φ that isn't a permutation of irreducible polynomials. Since $\varphi(f + g) = \varphi(f) + \varphi(g)$, φ can be represented as binary invertible matrix. Then there exists a row that has at least two units. Each row and column corresponds to irreducible polynomial. So φ maps an irreducible polynomial to product of at least two different irreducible polynomials, which is impossible. **n**

Product of permutations corresponds to product of automorphisms, so group of ring $\boldsymbol{G}_n$ automorphisms is isomorphic to group of permutations of 2^n elements. There exists $2^n!$ different isomorphisms of ring $\boldsymbol{G}_n$ and the cardinality of $\boldsymbol{G}_n$ automorphisms is larger then the key set cardinality.

Side channel attack protection of a cipher can be based on random periodically changed automorphisms. Before encryption beginning plaintext x and key k are transformed by automorphisms $x \leftarrow \sigma(x)$, $k \leftarrow \sigma(k)$. Let S — an encryption operation for input x in original algorithm and σ is automorphism of algebraic structure. Then isomorphic operation S' is defined by equation $S'(x) = \sigma(S(\sigma^{-1}(x)))$ or

$$S' = \sigma S \sigma^{-1}. \qquad (1)$$

After encryption is finished inverse automorphism is applied to ciphertext. Automorphism σ must agree with operator S structure so that modified operator S' is effectively computable. Most of automorphisms of $\boldsymbol{G}_n$ follows that modified encryption operation (such as DES substitution) depends on larger number of variables.

Assume that operator S' uses secret information and is cryptographically weak. If adversary knows input and output of operator S', he can compute the secret. If input, output and method of operator S' change according to (1) at random and ad-

versary knows input and output of operator S', but does not know its method, he cannot obtain the secret.

Let S_i, S_{i+1} are two consecutive operators and σ_i, σ_{i+1} — corresponding commutative isomorphisms. Then due to commutative property equality holds

$$S_{i+1}'S_i' = \sigma_{i+1}S_{i+1}\sigma_{i+1}^{-1}\sigma_i S_i \sigma_i^{-1} = \sigma_{i+1}S_{i+1}\sigma_i\sigma_{i+1}^{-1}S_i\sigma_i^{-1}.$$

This equation shows that between two consecutive operators the text is always protected: previous isomorphism is taken off only after the next isomorphism is applied. Last equation by induction describes protection of encryption algorithm. Used random automorphisms must commute.

So the protection method takes operations:

1. Take at random set of commutative automorphisms. Apply initial automorphism to key and plaintext (to ciphertext if decryption holds).
2. Modify set of encryption operators according to (1).
3. Apply the inverse last automorphism to ciphertext (to plaintext if decryption holds).
4. Periodically change random automorphisms.

Note that random isomorphisms can be used independently on transmitting and receiving sides.

Note that almost all automorphisms of ring $\boldsymbol{G}_n$ are not computable. Usually it is hard to find set of commutative random isomorphisms that doesn't decrease the rate of the cipher significantly. For example Russian cipher GOST 28147–89 is not convenient for such protection method.

2. Isomorphisms of RIJNDAEL based on field $\mathbb{F}_{256}$ isomorphisms

RIJNDAEL has block size 16 bytes, key size 16, 24 or 32 bytes for 10, 12 or 14 rounds correspondingly [8]. Encrypted text is presented as set of elements of field $\mathbb{F}_{256} = \mathbb{F}_2[x]/(p(x))$, where $p(x) = x^8 + x^4 + x^3 + x + 1$ is irreducible polynomial. Let t be the root of $p(x)$. Each round has next operations.

1. XOR (text, round key).
2. Byte substitution represented as a table (method of its computation is inessential).
3. Shift rows. Block is represented as 4×4 matrix $\begin{pmatrix} b_0 & b_4 & b_8 & b_{12} \\ b_1 & b_5 & b_9 & b_{13} \\ b_2 & b_6 & b_{10} & b_{14} \\ b_3 & b_7 & b_{11} & b_{15} \end{pmatrix}$. It is transformed to matrix $\begin{pmatrix} b_0 & b_4 & b_8 & b_{12} \\ b_5 & b_9 & b_{13} & b_1 \\ b_{10} & b_{14} & b_2 & b_6 \\ b_{15} & b_3 & b_7 & b_{11} \end{pmatrix}$.
4. Mix columns. Columns of 4×4 matrix are multiplied by matrix $C = \begin{pmatrix} t & t+1 & 1 & 1 \\ 1 & t & t+1 & 1 \\ 1 & 1 & t & t+1 \\ t+1 & 1 & 1 & t \end{pmatrix}$ over field $\mathbb{F}_{256}$.

Decryption procedure takes inverse operators in inverse order.

There are 8 commutative automorphisms of field $\mathbb{F}_{256}$, given by degrees of map $\sigma(x) = x^2$. Since there are 30 irreducible polynomials of degree 8, RIJNDAEL can be described in 240 different views. Those automorphisms save the structure of the cipher and are RIJNDAEL isomorphisms [9]. Note that only 8 automorphisms commute, defined for given irreducible polynomial. Galois group of irreducible polynomial $p(x)$ over field $\mathbb{F}_2$ is generated by automorphism $\sigma(t) = t^2$. If t is a root of irreducible polynomial $p(x)$, then other its roots are t^2, t^4, ..., t^{128}.

All 240 isomorphisms of RIJNDAEL, based on finite field isomorphisms, act on set of text and key bytes as 8×8 matrices over field $\mathbb{F}_2$ [10]. Each isomorphism changes substitution and constants of mix column operator according to (1), but elements 0 and 1 stay fixed. Hence if substitution output byte is zero for original RIJNDAEL, then it is zero for any isomorphic RIJNDAEL.

Note that these 240 isomorphisms remain the RIJNDAEL structure and represent a small part of all possible isomorphisms of this cipher. Each round of isomorphic cipher includes XOR text and key operation, byte look-up substitution, shift rows (as in original cipher) and mix columns (with changed constants) operations.

Consider next side channel attack using chosen plaintext (ciphertexts). Assume that adversary knows clock frequency of the encryption device, can precisely determine the moment of substitution output reading and can choose plaintexts. Assume that substitution is performed for bytes and adversary can find hamming weight of substitution output byte. This assumption can be explained because electronic cell switches $0 \to 1$ and $1 \to 0$ in different time, hence electromagnetic signals are different. The smaller is hamming weight of substitution output, the larger is corresponding signal level. In practice this signal is smaller than noise, but adversary can detect, recognize and receive the signal by repeating it sufficiently many times.

Cryptanalytic side-channel attack has two stages. First stage is search for plaintext bytes that give likely zero output substitution bytes in the first round. If substitution output byte and corresponding plaintext byte are known, the key byte can be easily computed. This allows ordering possible keys according their probabilities to be true. Second stage is key enumeration beginning most likely ones.

In the first stage of side channel attack adversary can adaptively change bits in the plaintext first byte to minimize hamming weight of the corresponding substitution output byte in the first round. Then he can find first key byte. Other key bytes can be obtained in similar way.

Definition 6. Let N_0 is number of chosen plaintext encryptions needed to find a key using side channel attack without protection, N_1 is number of chosen plaintext encryptions needed to find a key using side channel attack with given protection. Efficiency of side channel attack protection method is ratio N_1/N_0.

If cipher is protected by finite field random isomorphisms, adversary can use the next attack method. He adaptively chose the plaintext first byte so that substitution output byte becomes zero. Each random isomorphism fixes zero, hence this substitution output byte does not depend on random isomorphisms.

If cipher has no protection, adversary must change ≈8 plaintext bits to obtain zero byte. If cipher has protection, based on finite field isomorphisms, adversary must make ≈256 attempts to obtain zero output byte. So efficiency of such protection method is ≈32.

If RIJNDAEL random isomorphisms based on of field $\mathbb{F}_{256}$ isomorphisms are used, then each byte, different from 0 and 1, becomes random and its hamming weight is random too. Before encryption irreducible polynomial and set of commutative field automorphisms are to be chosen, plaintext and the round keys must be transformed by these automorphisms. After encryption is finished ciphertext must be transformed by inverse automorphism. If one uses common automorphism for all bytes during encryption, the encryption rate is maximal. Assume that processor can multiply elements over $\mathbb{F}_{256}$ fast. Then encryption process in original and transformed RIJNDAEL differ only by plaintext, key and ciphertext transformations. So the rate decreases less then 10%.

3. Affine isomorphisms of the RIJNDAEL

The cause of weakness of the finite field isomorphism protection is that each isomorphism fixes zero byte. Consider random affine isomorphism of the cipher given by equation

$$\sigma(\mathbf{x}) = L\mathbf{x} + \mathbf{a}, \quad \sigma^{-1}(\mathbf{x}) = L^{-1}(\mathbf{x} + \mathbf{a}). \tag{2}$$

for random invertible 8×8 matrix L and random vector $\mathbf{a}$. Such isomorphism does not fix zero byte and hence its efficiency can be more than in previous case. Number of plaintexts encrypted with common affine isomorphism before changing is determined by adversary laboratory possibilities. Note that affine map in field $\mathbb{F}_{256}$: $\sigma(x(t)) = g(t)x(t) + a(t)$ for constants $g(t)$, $a(t)$ is also cipher isomorphism and can be described as map (2).

Consider RIJNDAEL protection by common random isomorphisms for all bytes. Cipher affine isomorphisms differentially acts on text bytes and round key bytes. Let $\mathbf{x}$, $\mathbf{k}$ are plaintext (intermediate text) and round key bytes. Then $\sigma(\mathbf{x}) = L\mathbf{x} + \mathbf{a}$, $\sigma(\mathbf{k}) = L\mathbf{x}$, $\sigma(\mathbf{x} + \mathbf{k}) = L(\mathbf{x} + \mathbf{k}) + \mathbf{a}$.

Protected substitution operation can be computed according to (1).

Shift rows operation is the same as in original RIJNDAEL.

Mix column operation takes finite field multiplication and addition. This operation can be written as

$$(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4) = \psi C \psi^{-1}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4).$$

Here ψ means the action of σ operator for each byte. First output byte is described by equation

$$\mathbf{y}_1 = L(t\cdot(L^{-1}(\mathbf{x}_1 + \mathbf{a})) + (t + 1)\cdot(L^{-1}(\mathbf{x}_2 + \mathbf{a})) + L^{-1}(\mathbf{x}_3 + \mathbf{a}) + L^{-1}(\mathbf{x}_4 + \mathbf{a})) + \mathbf{a} =$$
$$L(t\cdot L^{-1}\mathbf{x}_1 + (t + 1)\cdot L^{-1}\mathbf{x}_2 + L^{-1}\mathbf{x}_3 + L^{-1}\mathbf{x}_4),$$

since $LL^{-1}\mathbf{a} = \mathbf{a}$ and $LtL^{-1}\mathbf{a} + LtL^{-1}\mathbf{a} = \mathbf{0}$.

This equation shows that cipher common affine isomorphism (2) acts on mix column input, output and method as cipher common linear isomorphism.

It is not hard to choose involution $\sigma(\mathbf{x}) = \sigma^{-1}(\mathbf{x})$ that allows simplification of protected encryption according to (1). Then

$$L\mathbf{x} + \mathbf{a} = L^{-1}\mathbf{x} + L^{-1}\mathbf{a}.$$

This equality holds if $L^2 = E$ — identity matrix and $(L + E)\mathbf{a} = \mathbf{0}$. If rank of 8×8 matrix $L + E$ is r, then number of solutions of equation $(L + E)\mathbf{a} = \mathbf{0}$ is 2^{8-r}.

Matrix L over field $\mathbb{F}_2$ is invertible if its rows $\mathbf{l}_i$ satisfy inequalities: $\mathbf{l}_1 \neq \mathbf{0}$, $\mathbf{l}_2 \notin \{\mathbf{0}, \mathbf{l}_1\}$, $\mathbf{l}_3 \notin \{\mathbf{0}, \mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_1 + \mathbf{l}_2\}$, $\mathbf{l}_4 \notin \{\mathbf{0}, \mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3, \mathbf{l}_1 + \mathbf{l}_2, \mathbf{l}_1 + \mathbf{l}_3, \mathbf{l}_2 + \mathbf{l}_3, \mathbf{l}_1 + \mathbf{l}_2 + \mathbf{l}_3\}$ and so on. There are $N_n = \prod_{i=0}^{n-1}\left(2^n - 2^i\right)$ invertible $n\times n$ matrices over field $\mathbb{F}_2$ and $N_8 = 2^{28}\cdot 3^5\cdot 5^2\cdot 7^2\cdot 17\cdot 31\cdot 127$.

Random square root of identity matrix can be found by next algorithm.

1. Choose at random upper triangle matrix L_u and lower triangle matrix L_l, with elements $a_{ii} = 1$ and permutation matrix P. These matrices are invertible.
2. Compute $L = PL_uL_l$.
3. Let $m = 3^5\cdot 5^2\cdot 7^2\cdot 17\cdot 31\cdot 127$. If $L^m = E$, then go to step 1, else set $L_{i+1} \leftarrow L_i^2$ until $L_k = E$.
4. If rank r of matrix $L_{k-1} + E$ satisfies $r < 8$, L_{k-1} is the required square root of identity matrix that allows getting involution σ.

Non-zero vector $\mathbf{a}$, satisfying equation $(L + E)\mathbf{a} = \mathbf{0}$ can be obtained by little enumeration or by using linear algebra methods.

For example, square roots of identity matrix

$$L_1 = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad L_2 = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

give ranks $r(L_1 + E) = 2$, $r(L_2 + E) = 1$. So there are many vectors $\mathbf{a}_1$, $\mathbf{a}_2$ which define involutions. For instance, vectors $\mathbf{a}_1 = (1, 0, 1, 0, 0, 0, 1, 1)$, $\mathbf{a}_2 = (1, 0, 1, 1, 0, 1, 1, 0)$ with the matrices give involutions $L_1\mathbf{x} + \mathbf{a}_1$, $L_2\mathbf{x} + \mathbf{a}_2$ correspondingly.

If all bytes are protected with common random isomorphism, then adversary can use differential side channel attack. He fixes the first plaintext byte and changes the second plaintext byte to obtain common substitution outputs for first and second bytes. He can detect this fact by search maximum of product of signals

corresponding to first and second byte. Then second key byte can be expressed as function of first key byte. Adversary repeats this procedure for first and third bytes, for first and fourth bytes and so on. Then round key ban be easily computed.

To protect such attack, different isomorphisms can be used for different bytes.

Theorem 7. If cipher isomorphisms $\varphi_1(\mathbf{x}) = L\mathbf{x} + \mathbf{a}$, $\varphi_2(\mathbf{x}) = L\mathbf{x} + \mathbf{b}$ are involutions, then $\varphi_3(\mathbf{x}) = L\mathbf{x} + \mathbf{a} + \mathbf{b}$ is involution and $\varphi_1\varphi_2 = \varphi_2\varphi_1$.

Proof. For first statement it is sufficient to prove that $L(\mathbf{a} + \mathbf{b}) = \mathbf{a} + \mathbf{b}$. This is true because $L(\mathbf{a} + \mathbf{b}) = L\mathbf{a} + L\mathbf{b}$, $L\mathbf{a} = \mathbf{a}$, $L\mathbf{b} = \mathbf{b}$.

Consider products of isomorphisms

$$\varphi_1\varphi_2(\mathbf{x}) = L(L\mathbf{x} + \mathbf{b}) + \mathbf{a} = L^2\mathbf{x} + L\mathbf{b} + \mathbf{a},$$

$$\varphi_2\varphi_1(\mathbf{x}) = L(L\mathbf{x} + \mathbf{a}) + \mathbf{b} = L^2\mathbf{x} + L\mathbf{a} + \mathbf{b}.$$

Sum of right parts of these equations gives $L\mathbf{b} + \mathbf{a} + L\mathbf{a} + \mathbf{b} = L\mathbf{a} + \mathbf{a} + L\mathbf{b} + \mathbf{b} = \mathbf{0}$, hence $\varphi_1\varphi_2(\mathbf{x}) + \varphi_2\varphi_1(\mathbf{x}) = \mathbf{0}$ and $\varphi_1\varphi_2(\mathbf{x}) = \varphi_2\varphi_1(\mathbf{x})$. **n**

Theorem 8. Let L — arbitrary invertible matrix over $\mathbb{F}_2$ and $L\mathbf{a} = \mathbf{a}$. Then equality $L^i\mathbf{a} = \mathbf{a}$ holds for all i.

Proof. From equality $L\mathbf{a} = \mathbf{a}$ obtain $L^2\mathbf{a} = L\mathbf{a} = \mathbf{a}$. Further by induction. **n**

Theorem 9. Let L — arbitrary invertible matrix over $\mathbb{F}_2$ and $L\mathbf{a} = \mathbf{a}$, $L\mathbf{b} = \mathbf{b}$. Then maps $\varphi_1(\mathbf{x}) = L^i\mathbf{x} + \mathbf{a}$, $\varphi_2(\mathbf{x}) = L^j\mathbf{x} + \mathbf{b}$ commute.

Proof. Consider products of maps.

$$\varphi_1\varphi_2(\mathbf{x}) = L^i(L^j\mathbf{x} + \mathbf{b}) + \mathbf{a} = L^{i+j}\mathbf{x} + L^i\mathbf{b} + \mathbf{a},$$

$$\varphi_2\varphi_1(\mathbf{x}) = L^j(L^i\mathbf{x} + \mathbf{a}) + \mathbf{b} = L^{i+j}\mathbf{x} + L^j\mathbf{a} + \mathbf{b}.$$

Sum of right parts of these equations gives $L^i\mathbf{b} + \mathbf{a} + L^j\mathbf{a} + \mathbf{b} = L^j\mathbf{a} + \mathbf{a} + L^i\mathbf{b} + \mathbf{b} = \mathbf{0}$ (according to theorem 8). Hence $\varphi_1\varphi_2(\mathbf{x}) + \varphi_2\varphi_1(\mathbf{x}) = \mathbf{0}$ and $\varphi_1\varphi_2(\mathbf{x}) = \varphi_2\varphi_1(\mathbf{x})$. **n**

According to theorem 9 different affine isomorphisms for different bytes can be used. In such isomorphisms matrices are degrees of given matrix L and vectors $\mathbf{a}_i$ satisfy condition $L\mathbf{a}_i = 0$.

If such random isomorphisms change sufficiently often, then there are no known methods of side channel attacks. So efficiency of affine isomorphisms can be extremely large and we can suppose that side channel attacks can be eliminated.

The rate of such protection method is near to the rate of protection method based on finite field isomorphisms.

References

1. Kahn D. The codebreakers. — Sphere books ltd, London, 1973.
2. Kelsey J., Schneier B., Wagner D. and Hall S. Side channel cryptanalysis of product ciphers, Proceedings of ESORICS'98, Springer-Verlag, 1998, 97-110.

3. http://en.wikipedia.org/wiki/Side_channel_attack.
4. Biham E., Shamir A. Differential cryptanalysis of DES-like cryptosystems // Advances in Cryptology — CRYPTO ’90. Lecture Notes in Computer Science. Springer–Verlag. 1991. Vol. 537. P. 2–21.
5. Matsui M. Linear cryptanalysis method for DES cipher // Advances in Cryptology — EUROCRYPT ’93. Lecture Notes in Computer Science. Springer–Verlag. 1994. Vol. 765. P. 386–397.
6. http://eprint.iacr.org/2004/049.
7. Rostovtsev A.G. Side channel attack protection based on random isomorphisms // Proceedings of conference “Mathematics and security of information technologies 2004, Moscow State University, Moscow, 2004 (in Russian), available in: http://www.ssl.stu.neva.ru/ssl/archieve/sidech1.pdf.
8. Announcing the advanced encryption standard (AES). Federal Information Processing Standards Publication, 2001.
9. E. Barkan, E. Biham. In how many ways can you write RIJNDAEL? // Proceedings of ASIACRYPT2002, LNCS 2501, pp. 160-175, 2002.