White-Box Cryptography and SPN ciphers. LRC method.

Schelkunov D., Ph.D.
Bauman Moscow State Technical University
d.schelkunov@gmail.com

Abstract. *The method of concealing a linear relationship between elements of a finite field (LRC method) is described. An LRC method based approach to the secure white-box implementations creating problem is considered. SPN cipher characteristics to create its secure White-Box implementation are revealed.*

Keywords. *public-key cryptography, secret-key cryptography, White-Box cryptography, SPN cipher, block cipher, finite field, LRC method.*

1. Introduction

The current cryptography state of art is characterized by active researches in the field of the *White-Box* cryptography [1] - [8]. *White-Box* cryptography methods are intended for modifying a symmetric cipher for the asymmetric cryptography. Indeed, if key recovery from the *White-Box* implementation of the encryption algorithm is a difficult task, as well as the creation of the decryption algorithm, having an encryption algorithm, then the pair of algorithms (encryption and decryption) is actually a key pair, where the *White-Box* implementation of the encryption algorithm is public, and the implementation of the decryption algorithm is not available for the analyst. Such scheme may have several advantages compared with classical asymmetric cryptosystems. The main advantage is an equation speed.

There were several *White-Box* implementations of the known block ciphers in recent years. The secret key was hidden in the lookup tables *(S-boxes*) in all of these implementations. Thus, if one uses a white box implementation of a block cipher, neither the encryption key, nor the round keys are presented explicitly in the program. However, all these implementations are not resistant to the chosen plaintext attacks *(CPA)* [6] - [8]. Moreover, even if it is difficult to restore the key itself, it is possible to construct reverse lookup tables, and therefore, it is possible to reconstruct a decryption algorithm having a white box implementation of the encryption algorithm.

This paper discusses an approach which can help to hide a linear relationship between elements of a finite field (*LRC* method). A special symmetric scheme with a structure similar to *Rijndael* [9] was created to show how *LRC* method could be used. The impossibility of the *LRC* method applying for *Rijndael White-Box* implementation creating is also concluded.

2. LRC-method

Consider the following system:

$$\begin{cases} x_1 = ((a \cdot b)(mod\ p_1) \cdot c)(mod\ p_2) \\ x_2 = (a \cdot (b \cdot c)(mod\ p_2))(mod\ p_1) \end{cases} \quad (1)$$

Here p_1, p_2 are irreducible polynomials of *n*-th degree and a,b,c,x_1,x_2 are polynomials of a degree less than *n*. We can assume that in the general case $x_1 \neq x_2$.

Proposition 1.

There are polynomials a,b,c *in system (1) such that* $x_1 \neq x_2$.

Proof:

If $x_1 = x_2$, then we have the following:

$$a \cdot b \cdot c - q \cdot c \cdot p_1 - v \cdot p_2 = a \cdot b \cdot c - a \cdot u \cdot p_2 - r \cdot p_1 \qquad (2)$$

Here q – quotient of $a \cdot b$ and p_1, v - quotient of $(a \cdot b)(mod\ p_1) \cdot c$ and p_2, u – quotient of $b \cdot c$ and p_2, r - quotient of $(b \cdot c)(mod\ p_2) \cdot a$ and p_1.

From (2) it follows that:

$$\frac{q \cdot c - r}{a \cdot u - v} = \frac{p_2}{p_1} \qquad (3)$$

From (3) it is obvious that $q \cdot c - r$ must be divisible by p_2, but the degree of the polynomial r does not exceed the degree of the polynomial p_2. The same can be said about the degree of polynomials q,c,a,u and v. Thus, the equality (3) is preserved either if one of the polynomials a,b or c is zero, or if the degree of $a \cdot b \cdot c$ does not exceed the degree of the polynomial p_1 (p_2). It means that $q = r = u = v = 0$. That is not always feasible. It follows that in (1) $x_1 \neq x_2$ in general case. So, proposition 1 is true.

Now consider the following system:

$$\begin{cases} y_1(x) = (s(x) \cdot a(mod\ p_1)) \cdot b(mod\ p_2) \\ y_2(x) = (s(x) \cdot c(mod\ p_1)) \cdot d(mod\ p_3) \end{cases} \qquad (4)$$

Here p_1, p_2, p_3 - pairwise unequal irreducible polynomials over $GF(2)$ of equal degree, x, a, b, c, d – arbitrary chosen polynomials over $GF(2)$, $s(x)$ - non-linear function of x. Let p_1, p_2, p_3, a, b, c, d, $s(x)$ be unknown, and let $y_1(x)$, $y_2(x)$ be functions which are set via lookup tables. So, it is a hard problem to find a linear relationship between elements $s(x) \cdot a(mod\ p_1)$ and $s(x) \cdot c(mod\ p_1)$ in the field of order of the p_1 degree when $y_1(x)$ and $y_2(x)$ values are known. In accordance to proposition 1, the law of associativity for each of the expressions of (4) is not satisfied in general. That is why there is no linear dependence between $y_1(x)$ and $y_2(x)$ in the field of order of the p_1 degree. It is necessary to know b, d, p_2 and p_3 to find a linear relationship between $s(x) \cdot a(mod\ p_1)$ and $s(x) \cdot c(mod\ p_1)$. It is obvious that the complexity of this task is approximately 2^{2n}, where n – degree of the p_1. Let's modify system (4) as follows:

$$\begin{cases} y_1(x) = (...(s(x) \cdot a(mod\ p_1)) \cdot b^{(0)}(mod\ p_2^{(0)})...) \cdot b^{(k)}(mod\ p_u^{(k)}) \\ y_2(x) = (...(s(x) \cdot c(mod\ p_1)) \cdot d^{(0)}(mod\ p_3^{(0)})...) \cdot d^{(k)}(mod\ p_v^{(k)}) \end{cases} \qquad (5)$$

Here $p_i^{(\alpha)} \neq p_j^{(\alpha)}$. The complexity of restoration of a linear relationship between $s(x) \cdot a(mod\ p_1)$ and $s(x) \cdot c(mod\ p_1)$ is $2^{2n(k+1)}$ in this case. System (5) describes an approach named *LRC* method.

3. White-Box implementation

So, we can assume that if we multiply many times each element of the *T-box* [9] for the randomly chosen polynomials by modulo of the randomly chosen irreducible polynomials of the degree 8, the restoration of the linear relationship between elements of the *T-boxes* will be a hard problem. Formally, this *LRC* method implementation is presented as follows:

$$T'_i[a] = \begin{bmatrix} ((...(t_i^{(0)}(a)\cdot b_i^{(0,0)})(mod\ p_i^{(0,0)})\cdot b_i^{(0,1)}(mod\ p_i^{(0,1)})...)\cdot b_i^{(0,k_0)}(mod\ p_i^{(0,k_0)}) \\ ((...(t_i^{(1)}(a)\cdot b_i^{(1,0)})(mod\ p_i^{(1,0)})\cdot b_i^{(1,1)}(mod\ p_i^{(1,1)})...)\cdot b_i^{(1,k_1)}(mod\ p_i^{(1,k_1)}) \\ ((...(t_i^{(2)}(a)\cdot b_i^{(2,0)})(mod\ p_i^{(2,0)})\cdot b_i^{(2,1)}(mod\ p_i^{(2,1)})...)\cdot b_i^{(2,k_2)}(mod\ p_i^{(2,k_2)}) \\ ((...(t_i^{(3)}(a)\cdot b_i^{(3,0)})(mod\ p_i^{(3,0)})\cdot b_i^{(3,1)}(mod\ p_i^{(3,1)})...)\cdot b_i^{(3,k_3)}(mod\ p_i^{(3,k_3)}) \\ ..... \\ ((...(t_i^{(n)}(a)\cdot b_i^{(n,0)})(mod\ p_i^{(n,0)})\cdot b_i^{(n,1)}(mod\ p_i^{(n,1)})...)\cdot b_i^{(n,k_n)}(mod\ p_i^{(n,k_n)}) \end{bmatrix} \quad (6)$$

Here $t_i^{(j)}$ - element of the *T-box* before applying multiplications, $b_i^{(j,u)}$ – randomly chosen polynomial in $GF(2^8)$, $p_i^{(j,u)}$ – randomly chosen irreducible polynomial of the degree 8 under $GF(2)$. The following condition must be observed:

$$p_i^{(0,v)} \neq p_i^{(1,v)} \neq ... \neq p_i^{(n,v)} \quad (7)$$

It is possible to create a White-Box implementation of the *SPN* cipher which is based on the *T-box*es modified by *LRC*-method. For the *Rijndael*, for example, such modification would be as follows:

$$Y_j = \begin{bmatrix} y_j^{(0)} \\ y_j^{(1)} \\ y_j^{(2)} \\ y_j^{(3)} \end{bmatrix} = \begin{bmatrix} mix_j^{(0)}(t_j^{(0,0)}) \\ mix_j^{(1)}(t_j^{(1,0)}) \\ mix_j^{(2)}(t_j^{(2,0)}) \\ mix_j^{(3)}(t_j^{(3,0)}) \end{bmatrix} \oplus \begin{bmatrix} mix_j^{(0)}(t_j^{(0,1)}) \\ mix_j^{(1)}(t_j^{(1,1)}) \\ mix_j^{(2)}(t_j^{(2,1)}) \\ mix_j^{(3)}(t_j^{(3,1)}) \end{bmatrix} \oplus \begin{bmatrix} mix_j^{(0)}(t_j^{(0,2)}) \\ mix_j^{(1)}(t_j^{(1,2)}) \\ mix_j^{(2)}(t_j^{(2,2)}) \\ mix_j^{(3)}(t_j^{(3,2)}) \end{bmatrix} \oplus \begin{bmatrix} mix_j^{(0)}(t_j^{(0,3)}) \\ mix_j^{(1)}(t_j^{(1,3)}) \\ mix_j^{(2)}(t_j^{(2,3)}) \\ mix_j^{(3)}(t_j^{(3,3)}) \end{bmatrix} \quad (8)$$

Here $t_j^{(i,k)}$ - element of the T-box before applying any transformations, $mix_j^{(i)}$ - applied to $t_j^{(i,k)}$ transformation.

$$mix_j^{(i)}(t_j^{(i,k)}) = ((...(t_j^{(i,k)}\cdot b_j^{(i,0)})(mod\ p_j^{(i,0)})\cdot b_j^{(i,1)}(mod\ p_j^{(i,1)})...)\cdot b_j^{(i,n)}(mod\ p_j^{(i,n)}) \quad (9)$$

In formula (11) i - index of the *T-box*'s element, k - index of the *T-box*, j - number of 4-byte sequence, $b_j^{(u,v)}$ - arbitrary polynomial in $GF(2^8)$, $p_i^{(u,v)}$ – randomly chosen irreducible polynomial of the degree 8 under $GF(2)$.

4. Known attacks

The approach described above is not suitable for the *Rijndael*. It is possible to apply the chosen plaintext attack to such a White-Box implementation if a principle of creating lookup tables and a polynomial used in the *MixColumns* transformation are known. It is a simple task to find $mix_j^{(i)}$ transformations presented in (8) in this case. Indeed, we know that:

$$t_j^{(i,k)} = s[a] \cdot n \oplus key_j^{(i,k)} \qquad (10)$$

Here a - byte of the plaintext, $s[a]$ - known non-linear transformation in the *Rjndael* field, $key_j^{(i,k)}$ - part of the secret round key. The operation of multiplication in the formula (10) is made in the *Rijndael* field. After applying the *mix* transformation we obtain the following:

$$mix_j^{(i)}(t_j^{(i,k)}) = mix_j^{(i)}(s[a] \cdot n) \oplus mix_j^{(i)}(key_j^{(i,k)}) \qquad (11)$$

Let a and a' be two plaintext bytes. From (11) it follows that:

$$mix_j^{(i)}(t_j^{(i,k)}) \oplus mix_j^{(i)}(t'^{(i,k)}_j) = mix_j^{(i)}(n \cdot (s[a] \oplus s[a'])) \qquad (12)$$

We know n, $s[a]$, $s[a']$ in formula (14). Thus, we can easily find $mix_j^{(i)}$ by building a 2^8 bytes lookup table. After finding all of the *mix*-transformations (which are actually lookup tables), an adversary can easily reverse them and find $t_j^{(i,k)}$. It is a very simple task to find $key_j^{(i)}$ if we know $s[a]$. Thus, all of the round keys restorations are possible.

If n in formula (12) is unknown, it means that a *MixColumns* transformation under $GF(2^8)$ is unknown as well. In this case breaking the scheme presented above is still possible. Consider the neighboring *T-box*es in formula (8).

$$\begin{cases} t_j^{(0,0)} = s[a] \cdot n^{(0)} \oplus key_j^{(0,0)} \\ t_j^{(0,1)} = s[a] \cdot n^{(1)} \oplus key_j^{(0,1)} \\ t_j^{(0,2)} = s[a] \cdot n^{(2)} \oplus key_j^{(0,2)} \\ t_j^{(0,3)} = s[a] \cdot n^{(3)} \oplus key_j^{(0,3)} \end{cases} \qquad (13)$$

Let us assume that $n^{(0)} = \alpha$. An adversary can find a $mix_j^{(0)}$ transformation using an algorithm presented above. Consequently, $n^{(1)}, n^{(2)}, n^{(3)}$ can be found too. If the assumption $n^{(0)} = \alpha$ is true, then $n^{(0)}, n^{(1)}, n^{(2)}, n^{(3)}$ are coefficients of the *MixColumns* polynomial. It is easy to verify this assumption by applying *InvMixColumns* (which can be easily found) and watching how a round input byte impacts to the output round sequence of bytes (only one byte of the output round sequence has to be changed). So, it is obvious that the complexity of breaking of such White-Box implementation is 2^8.

In the case of the unknown non-linear *S-box* transformation the scheme presented above is still breakable. We consider a typical case when for every input byte of the round fully equal *S-box* lookup tables are used. Let $s[a]$ be unknown in system (13). In this case it suffices to assume that value. Let $s[a] = \beta$ and $s[a'] = \beta'$. Now we can apply the algorithm described above assuming that $n^{(0)} = \alpha$. We can check the correctness of the assumptions like in the case when $n^{(k)}$ is unknown. Now, the complexity of breaking is 2^{24}.

5. Positive results

So, we can conclude that the complexity of breaking in excess of 2^{24} can be achieved only when the lookup table for each input byte in each round is created at random, and the polynomial used in the linear transformation (*MixColumns*) is also unique and unknown in each round. Thus, it is impossible to create a *CPA* resistant *White-Box* implementation of the *Rijndael* using *LRC* method. Moreover, if *MixColumns* polynomial by $x^4 \oplus 1$ modulo and *S-boxes* are random, then it is possible to break such scheme by building 154 GB sized lookup tables which can be used to decrypt an encrypted text. And only if a *MixColumns* polynomial in every round will be by $x^{16} \oplus 1$ modulo, then *White-Box* implementation can be *CPA* resistant. Of course, *S-boxes* must be random for every input byte of each round, and *MixColumns* polynomial by $x^{16} \oplus 1$ modulo must be random too for every round.

Of course, this paper contains a lot of open questions. But if the assumptions presented above are correct, we can construct a very fast asymmetric scheme using randomly generated *SPN* cipher. Since attack is made on each round separately, we can reduce their number to the minimum without security reducing.

References:

1. S. Chow, P. Eisen, H. Johnson, P.C. van Oorschot, A White-Box DES Implementation for DRM Applications, 2002.
2. S. Chow, P. Eisen, H. Johnson, P.C. van Oorschot, White Box Cryptography and an AES Implementation, 2002.
3. Julien Bringer, Herve Chabanne, Emmanuelle Dottax, White Box Cryptography: Another Attempt, 2006.
4. Hamilton E. Link, William D. Neumann, Clarifying Obfuscation: Improving the Security of White-Box Encoding.
5. http://www.cosic.esat.kuleuven.be/publications/thesis-152.pdf - B. Wyseur, "White-Box Cryptography," PhD thesis, Katholieke Universiteit Leuven, B. Preneel (promotor), 169+32 pages, 2009.
6. Dmitry Schelkunov On practical implementations of the White-Box cryptography, // RusCrypto-2009 conference.
7. Dmitry Schelkunov. The development of methods of software protection from analysis and modification based on the obfuscation, PhD thesis, 2009.
8. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf - AES specification.