ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 2012. No. 2(118)
ELEKTRONIKA IR ELEKTROTECHNIKA

SYSTEM ENGINEERING, COMPUTER TECHNOLOGY
T 120
SISTEMŲ INŽINERIJA, KOMPIUTERINĖS TECHNOLOGIJOS

Implementation Analysis of Matrix Power Cipher in Embedded Systems

K. Luksys, E. Sakalauskas
Department of Applied Mathematics, Kaunas University of Technology,
Studentu str. 50-327a, 51368 Kaunas, Lithuania, e-mails: kestutis.luksys@ktu.lt, eligijus.sakalauskas@ktu.lt
A. Venckauskas
Department of Computer Science, Kaunas University of Technology,
Studentu str. 50, 51368 Kaunas, Lithuania, e-mail: algimantas.venckauskas@ktu.lt

crossref http://dx.doi.org/10.5755/j01.eee.118.2.1182

Introduction

In this paper we present the implementation analysis of the matrix power cipher (MPC) in embedded systems. These systems have restricted computation resources, i.e. computation speed and memory.

So far the question of fast ciphers construction is very actual since a lot of projects were announced to solve this problem. For example in 2000–2003 the NESSIE (New European Schemes for Integrity and Encryption) project was carried out [1]. Some block ciphers were proposed and accepted. Among them was AES-128 cipher. But nevertheless despite the possibility to transform the block cipher to stream cipher, the project authorities recognized that no one proposed stream cipher met the security and speed requirements. The other project was dedicated to fast stream cipher design and was named eStream [2]. Some solutions of fast stream ciphers were proposed and two closely related directions of investigation were determined. The first one is hardware encryption and the second one is software encryption. International Association for Cryptographic Research is organizing annual conferences: International Workshop on fast software encryption (FSE), and International Workshop on Cryptographic Hardware and Embedded Systems (CHES).

The main requirements for the new cipher proposal are security and speed. It is assumed that new cipher should have a speed no less than AES-128 speed.

We would like to present here a theoretical implementation analysis of new matrix power cipher in embedded systems. The components of this cipher are presented and their security is analysed in [3, 4]. This analysis is necessary to get a preliminary cipher speed data and to compare it with AES-128 speed. Since so far AES cipher is realized in a number of microprocessors using hardware co-processors, to be honest we are comparing AES implementation in ordinary AVR family microprocessors with our cipher implementation in same microprocessors. The data of AES-128 speed was taken from [5]. The speed of our cipher was estimated by counting the microprocessor operations required for cipher realization and estimating their speed in microprocessor's clock cycles. Hence the number of cycles for 1 bit can be evaluated and compared with the same figure of AES-128 realization. On the base of these data the decision can be made if it is sensible to realize the proposed matrix power cipher using other software and hardware improvements.

Matrix power S-box

The main component of MPC is the matrix power function (MPF). To define the MPF for symmetric ciphering we use two sets of matrices. One set $\mathbf{M_G}$ is defined as a matrix group with ordinary matrix multiplication over the ring, and the other set $\mathbf{M}$ is the matrices over the finite field. All matrices are square and of the same order m.

The MPF f is defined as a composition of two functions, which are called left and right MPFs. The left MPF provides a mapping from $\mathbf{M_G} \times \mathbf{M}$ to $\mathbf{M}$. Symbolically this operation can be expressed as

$$Y = {}^{L}X = \left\{ \prod_{s=1}^{m} x_{sj}^{l_{is}} \right\}, \quad (1)$$

here $L \in \mathbf{M_G}$ and $X, Y \in \mathbf{M}$. Similarly, the right MPF provides a mapping from $\mathbf{M} \times \mathbf{M_G}$ to $\mathbf{M}$ and can be expressed as

$$Z = Y^{R} = \left\{ \prod_{t=1}^{m} y_{it}^{r_{tj}} \right\}, \quad (2)$$

here $R \in \mathbf{M_G}$ and $Y, Z \in \mathbf{M}$. Then the MPF can be noted as [4]

$$f(X) = {}^{L}X^{R} = \left\{ \prod_{t=1}^{m} \prod_{s=1}^{m} x_{st}^{l_{is} \cdot r_{tj}} \right\}. \quad (3)$$

For the successful inversion of the MPF, i.e. calculation of $f^{-1}(X)$, we must be able to calculate the inverse matrices of L and R. These matrices will exist since L and R are from the group. In [4] we proved that

$$f^{-1}\left(f(X)\right) = {}^{L^{-1}}\left({}^{L}X^{R}\right)^{R^{-1}} = {}^{L^{-1}L}X^{RR^{-1}} = X. \quad (4)$$

This equation holds only then matrix X is chosen from the Galois field $GF(2^n)^{m \times m}$, and then, according to Fermat theorem, the group $\mathbf{M_G}$ is a subset of $\boldsymbol{Z}_{2^n-1}^{*\,m \times m}$.

However the MPF cannot be used directly due to special requirements for the input data. None element of input matrix X should be equal to zero. In such a case, the output matrix would contain only zeros. Hence matrix X cannot be an input data matrix for the symmetric ciphering, i.e. matrix representing plain text. If the input data matrix we denote by D, then this matrix must be transformed to the matrix X without zero entries.

This problem is solved by constructing the S-box function (SBF) F based on the MPF as an injective mapping F: $GF(2^{n-1})^{m \times m} \rightarrow GF(2^n)^{m \times m}$. The SBF F is a composition of some auxiliary function g_K and the MPF f with both defined by additional key matrix $K \in \boldsymbol{Z}_{2^{n-1}}^{m \times m}$ and matrices $L, R \in \mathbf{M_G}$ correspondingly.

The MPF is a mapping one-to-one, thus function g_K must perform an injective affine transformation from $GF(2^{n-1})^{m \times m}$ to $GF(2^n)^{m \times m}$. We proposed to express it in the following way

$$g_K(D) = D + K + \mathbf{1} = X, \quad (5)$$

here the addition operations are the ordinary additions of matrices. It is the additions of entries of matrices but they are defined according to the addition rules in $\boldsymbol{Z}_{2^n}$. Matrix denoted by $\mathbf{1}$ is the matrix in $\boldsymbol{Z}_{2^n}^{m \times m}$ consisting of arithmetical unity elements in all its positions. Using this transformation we obtain a matrix $X \in \mathbf{M}$ which does not contain zero elements, despite the presence of zero elements in matrix D. The smallest possible element of $\{x_{ij}\}$ is 1 and the largest is $2^n - 1$.

Then the SBF F explicitly is defined by the following relations

$$F(D) = f(g_K(D)) = {}^{L}(D + K + \mathbf{1})^{R} = C. \quad (6)$$

Single ciphertext matrix element c_{ij} can be expressed for $i, j = 1, 2, \ldots, m$ by the formula

$$\prod_{t=1}^{m} \prod_{s=1}^{m} (d_{st} + k_{st} + 1)^{l_{is} r_{tj}} = \prod_{t=1}^{m} \prod_{s=1}^{m} x_{st}^{l_{is} \cdot r_{tj}} = c_{ij}, \quad (7)$$

here 1 is a unity in $\boldsymbol{Z}_{2^n}$.

The function of inverse matrix power S-box, i.e. decryption operation, can be written in a similar formal way as in (6):

$$F^{-1}(C) = g_K'\left({}^{L^{-1}}C^{R^{-1}}\right) = {}^{L^{-1}}C^{R^{-1}} - K - \mathbf{1} = D. \quad (8)$$

Matrix power cipher

The matrix power cipher (MPC) is a t-round symmetric cipher which main round function is the MPF. The main data blocks are $m \times m$ matrices with elements of n bits length. Due to the use of the MPF the elements of plaintext data matrix D are 2^{n-1} bits length and the elements of corresponding ciphertext matrix C are 2^n bits length.

The MPC uses $2t + 1$ key matrices. The key matrices L_i and R_i are randomly chosen from the group $\mathbf{M_G}$ and are used in the MPF. In addition there is one key matrix K randomly chosen from $\boldsymbol{Z}_{2^{n-1}}^{m \times m}$ and used in function g_K in the first round. In this paper we will not specify the key generation phase and will focus only on encryption and decryption operations.

Encryption. The first round of the MPC is the matrix power S-box function

$$F_1(D) = {}^{L_1}(D + K + \mathbf{1})^{R_1} = X_1. \quad (9)$$

After the first round, the size of each data matrix element is increased by one bit. This does not take place for the next rounds, since the output matrix X will have no zero elements. But the direct repeated use of the MPF will not increase the security because it can be substituted with one MPF with the adequate key. For this reason additional function H is used instead of g_K. Function H consists of component functions h which are not equivalent to power mappings

$$H(X) = \{h(x_{ij})\}. \quad (10)$$

All functions h are chosen to be a permutations of $GF(2^n)$ to ensure valid decryption. To increase the security of the cipher these permutations should be cryptographically strong S-boxes. Some new cryptographically strong functions can be found in [6, 7, 8].

The next rounds ($1 < i \le t$) are the composition of the function H and the MPF

$$F_i(X_{i-1}) = {}^{L_i}(H(X_{i-1}))^{R_i} = X_i. \quad (11)$$

The output X_t of the last round is the ciphertext C.

Decryption. For the decryption of the cipher text C, all key matrices L_i , R_i must be inverted and inverse function of H must be calculated as well. Inverse matrices can be found using ordinary matrix arithmetic over Z_{2^n-1}

$$L_i' = L_i^{-1},\ R_i' = R_i^{-1},\ 1 \le i \le t. \quad (12)$$

All these matrices must exist since L_i and R_i are chosen from the group $\mathbf{M_G}$. For the valid decryption, these matrices must be used in reverse order. H^{-1} can be easily found by inversion of component function h.

The first $t - 1$ rounds are the compositions of the MPF with inversed keys and H^{-1} ($1 \le i < t$) in the following order

$$F_i^{'}(X_{i-1}) = H^{-1}\left({}^{L_{t+1-i}^{'}} X_{i-1}^{\ R_{t+1-i}^{'}} \right) = X_i , \qquad (13)$$

here $X_0 = C$.

The last round of MPC decryption is a composition of the MPF and the modified function $g_K^{'}$

$$F_t^{'}(X_{t-1}) = g_K^{'}\left({}^{L_1^{'}} X_{t-1}^{\ R_1^{'}} \right) = {}^{L_1^{'}} X_{t-1}^{\ R_1^{'}} - K - \mathbf{1} = D . \quad (14)$$

If all key matrices are true, then all elements of D are in GF(2^{n-1}).

Security assumptions of MPC

Although the MPF is based on exponentiation, the security of the matrix power S-box and whole cipher does not rely on classical DLP problem. The orders of the finite fields are considerable small and hence DLP can be efficiently solved by using look-up tables.

The security parameters of the MPC were analysed in [3, 4]: the order of matrices m, the size in bits of their elements n and the number of rounds t.

The matrix power S-box is resistant to algebraic cryptanalysis when $n \geq 3$ and $m \geq 4$. The MPF used in symmetric ciphering has different characteristics than the one used in asymmetric protocols [9]. The necessity of the MPF inversion requires the use of finite field with cyclic multiplicative group. However algebraic equations relating S-box input, output and key data form an underdefined multivariate quadratic system of equations over the ring. The solution of such system becomes intractable when parameters comply with given limits [4].

The guess and determine attack, when some key data is guessed and other is computed, is also infeasible for the matrix power S-box with $n \geq 3$ and $m \geq 4$ [4]. With these parameters the MPC becomes resistant to algebraic cryptanalysis and guess and determine attacks after the first round. All additional rounds only increase the complexity of those attacks and thus increase the security of the cipher.

The unique feature of the MPF is that its nonlinear part depends on the secret keys. And using it in S-box construction for the block cipher we do not get the ordinary S-box which could be used as nonlinear part of substitution-permutation (SP) or Feistel networks. Instead of that the matrix power S-box is like the whole complex SP network where diffusion and confusion of data bits are made at the same time. Secret key data is used as powers, thus nonlinear part of the matrix power S-box is unknown to the attacker and classical cryptanalysis attacks such as linear or differential are almost impossible to implement.

However we estimated the possible complexity of differential cryptanalysis of the matrix power S-box as of ordinary key dependent S-box. The expected differential probability of the whole S-box when $n = 8$ and $m = 4$ is less than 2^{-52}, i.e. it would be needed more than 2^{52} pairs of plaintext–ciphertext to at least try to mount the differential attack. In embedded systems we can safely state that it is impossible to gather more than 2^{60} bits of information encrypted with the same key. Of course, expected differential probability shows only the average case complexity. For assurance and greater security the MPC should be used in three or more rounds. In this case the actual differential probabilities spread close to expected probability which decreases with increasing number of rounds.

Implementation of MPC

Before specifying specific parameters of the MPC implementation in embedded systems, we briefly review the general operation count.

The direct implementation of the MPF according equation (3) would lead to m^4 multiplications over the ring, m^4 exponentiations and $m^4 - m^2$ multiplications in finite field for one data block encryption. This operations count could be reduced by separating the MPF to left and right functions. Then for one data block it would be needed $2m^3$ exponentiations and $2m^3 - 2m^2$ multiplications in the finite field. In the matrix power S-box there would be extra $2m^2$ additions modulus 2^n. This could be reduced to m^2 if matrices K and $\mathbf{1}$ would be added in key generation phase.

The next rounds of the MPC consist of the MPF and function H. The later requires m^2 table look-up operations, if component function h is used as a look-up table.

For the efficient implementation operations in finite field should be performed applying look-up tables. It would be needed two tables: one for multiplication, and one for exponentiation. Both would consists of $(2^n - 1)^2$ elements of n bits long.

It is clearly seen that total operation count depends on parameter m in cubic manner. Encrypted data size depends on the same parameter, too, but this dependence is only quadratic. Thus the increase of m will also raise the total operations count for one data bit.

Implementing the MPC in restricted environments parameters must be as low as possible but still in permissible security level. Therefore we recommend to choose $m = 4$, $n = 8$ and $t = 3$. The MPC with these parameters will take 704 table look-up operations and 16 additions. Two look-up tables would be 127 KB size. One look-up table for function h would be 256 B size. This implies that 8-bit microcontrollers with at least 128 KB of flash memory could be used.

Table look-up operation takes three microprocessor's clock cycles when table is stored in flash memory. If the table is stored in SRAM memory, then look-up operation takes two cycles. The later case is used for function H evaluation. Addition operation modulus 2^8 takes one cycle. Thus the MPC theoretically would take 2096 cycles to encrypt one block of 112 bits. Decryption operation in the MPC consists of the same operations as encryption and it would also take 2096 cycles for one data block.

The comparison of the MPC with the fastest AES-128 implementations on 8-bit AVR microcontrollers without hardware extensions is presented in Table 1.

The fastest known AES-128 realization on AVR microcontrollers without hardware extensions [5] encrypts faster than our MPC. But the decryption speed is slower. The MPC encrypts and decrypts at the same speed, thus on average theoretically MPC is faster than fastest implementation of AES with average speed of 19,1 cycles/b. Even greater speed of the MPC could be achieved

if it would be used in cipher feedback or output feedback mode. Then the cipher would process 128 bits of data at a time.

Table 1. Comparison of the MPC with AES implementations on AVR microcontrollers

Source	Encryption		Decryption	
	cycles	cycles/b	cycles	cycles/b
AES-128 [10]	2555	20.0	3193	24.9
AES-128 [11]	2474	19.3	3411	26.6
AES-128 [5]	1993	15.6	2901	22.7
MPC (112 b)	2096	18.7	2128	18.7
MPC (128 b)	2112	16.5	2112	16.5

Conclusions

This paper presents the theoretical implementation analysis of the matrix power cipher in embedded systems. We choose to analyse the cipher with three rounds and 128 bits data block. These parameters ensure that the MPC is sufficiently immune against algebraic cryptanalysis, guess and determine attacks and differential cryptanalysis.

Implementation of this cipher in 8-bit AVR family microcontrollers would require at least 127 KB of flash memory, 256 B of SRAM memory and 2096 microprocessor's clock cycles for encryption/decryption, i.e. 18.7 cycles for one plaintext data bit.

We compared the MPC implementation with the fastest AES-128 implementations on 8-bit AVR microcontrollers without hardware extensions. The fastest AES-128 implementations encrypts in 15.6 cycles/b, but the average encryption/decryption speed is 19.1 cycles/b. Thus theoretically, the MPC can operate faster than AES-128. The actual speed of the MPC could be increased as in the case of AES by code optimization and some special software and hardware enhancements.

References

1. **Preneel B.** Project "New European Schemes for Signatures, Integrity, and Encryption – NESSIE". – 2004. Online: https://www.cosic.esat.kuleuven.be/nessie/.
2. **Preneel B.** eSTREAM – the ECRYPT Stream Cipher Project. – 2009. Online: http://www.ecrypt.eu.org/ stream/.
3. **Luksys K., Nefas P.** Matrix Power S–Box Analysis // Information Science And Computing, book 4 "Advanced Studies in Software and Knowledge Engineering", 2008. – P. 97–102.
4. **Sakalauskas E., Luksys K.** The Matrix Power Function and its Application to Block Cipher S–box Construction // International Journal of Innovative Computing, Information and Control (ICIC International), 2012. – Vol. 8. – No. 4. – Accepted.
5. **Osvik D. A., Bos J.W., Stefan D., Canright D.** Fast Software AES Encryption // Proc. of FSE'10. – Springer–Verlag, 2010. – P. 75–93.
6. **Bierbrauer J.** New semifields, PN and APN functions // Designs, Codes and Cryptography. – Kluwer Academic Publishers, 2010. – Vol. 54. – No. 3. – P. 189–200.
7. **Carlet C., Feng K.** An Infinite Class of Balanced Vectorial Boolean Functions with Optimum Algebraic Immunity and Good Nonlinearity // Proc. Of IWCC'09, LNCS. – Springer–Verlag, 2009. – Vol. 5557. – P. 1–11.
8. **Pasalic E., Charpin P.** Some results concerning cryptographically significant mappings over GF(2^n) // Designs, Codes and Cryptography. – Kluwer Academic Publishers, 2010. – Vol. 57. – No. 3. – P. 257–269.
9. **Vitkus P., Sakalauskas E., Listopadskis N., Vitkiene R.** Microprocessor realization of key agreement protocol based on matrix power function // Electronics and Electrical Engineering. – Kaunas: Technologija, 2012. – No. 1(117). – P. 33–36.
10. **Otte D.** AVR–Crypto–Lib. – 2009. Online: http://www.das–labor.org/wiki/AVR–Crypto–Lib/en.
11. **Poettering B.** AVRAES: The AES block cipher on AVR controllers. – 2007. Online: http://point–at–infinity.org/avraes/.

Received 2011 09 05
Accepted after revision 2011 12 12

K. Luksys, E. Sakalauskas, A. Venckauskas. Implementation Analysis of Matrix Power Cipher in Embedded Systems // Electronics and Electrical Engineering. – Kaunas: Technologija, 2012. – No. 2(118). – P. 95–98.

In this paper we present a theoretical implementation analysis of new matrix power cipher in embedded systems. This cipher is based on the matrix power function. This allows achieving required security and efficiency while minimizing the number of rounds. In this paper we briefly overview the matrix power and the whole cipher, discuss the security assumptions and specify the limits of security parameters. The speed of the cipher was estimated by counting operations considering the usage of look-up tables and realization in 8-bits AVR microcontrollers. Theoretical speed of the matrix power cipher was compared with the fastest known AES-128 implementations. Our cipher performs faster than AES-128 when encryption and decryption are considered together. Bibl. 11, tabl. 1 (in English; abstracts in English and Lithuanian).

K. Lukšys, E. Sakalauskas, A. Venčkauskas. Matricinio laipsnio šifro realizacijos įterptinėse sistemose analizė // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2012. – Nr. 2(118). – P. 95–98.

Šiame straipsnyje pateikiama naujo matricinio laipsnio šifro realizavimo įterptinėse sistemose teorinė analizė. Šio šifro pagrindą sudaro matricinio laipsnio funkcija. Tai leidžia minimizuojant šifro iteracijų skaičių pasiekti norimą saugumą ir efektyvumą. Straipsnyje trumpai apžvelgiama matricinio laipsnio funkcija ir visas šifras, aptariamos saugumo prielaidos ir pateikiamos saugumo parametrų ribos. Šifro greitis įvertinamas nustatant operacijų skaičius, atsižvelgiant į peržvalgos lentelių naudojimą ir realizaciją 8 bitų AVR tipo mikroprocesoriuose. Teorinis matricinio laipsnio šifro greitis palyginamas su greičiausiomis žinomomis AES-128 šifro realizacijomis. Kartu vertinant užšifravimą ir iššifravimą, mūsų siūlomu šifru šifruojama greičiau negu AES-128 šifru. Bibl. 11, lent. 1 (anglų kalba; santraukos anglų ir lietuvių k.).