Imed El FRAY[1], Tomasz HYLA[1], Witold MAĆKÓW [1], Jerzy PEJAŚ[1]

West Pomeranian University of Technology, Faculty of Computer Science and Information (1)

Threshold sharing scheme for large files

***Abstract**. The known threshold sharing schemes applied directly to large secret files are ineffective and dangerous. Ineffectiveness of standard methods results from the need to generate and store a large number of shadows. In turn, the low security level of standard methods may be caused by not taking into account the properties of large files, such as file format and multiple reduplication of the same information contained in it. The paper proposes a new threshold secret sharing scheme, intended to distribute the large secret files. Due to used obfuscation techniques the proposed scheme prevents the reduction of the privacy threshold and thereby increases its security level. This technique is realized in three consecutive stages, the message expansion, linking and permutation using Rivest permutation polynomial. Splitting file into multiple groups with the same number of subsecrets strongly reduces the number of generated and stored shadows and in effect our scheme requires less storages as well as computing time. The Vandermond matrix is used on the stage of message recovery.*

***Streszczenie.** Schematy podziału progowego zastosowane bezpośrednio do dużych plików są nieefektywne i niebezpieczne. Niska wydajność tych metod wynika z konieczności generowania i przechowywania dużej ilości cieni. Brak bezpieczeństwa związany jest z właściwościami dużych plików, takimi jak znany format pliku czy potencjalne wielokrotne powtórzenie w pliku tych samych fragmentów informacji. W artykule zaproponowano nowy schemat podziału progowego sekretu przeznaczony do dystrybucji dużych plików. Dzięki wykorzystaniu mechanizmów zaciemniania i wiązania proponowany schemat zwiększa poziom bezpieczeństwa podziału. Technika ta realizowana jest w trzech kolejnych krokach: rozszerzeniu wiadomości, jej wiązaniu i permutacji z wykorzystaniem wielomianów permutacyjnych Rivesta. Podział pliku na grupy zawierające te samą liczbę sekretów zdecydowanie zmniejsza liczbę generowanych i przechowywanych cieni, w wyniku czego zmniejszają się wymagania związane z miejscem przeznaczonym do ich przechowywania jak i czasem potrzebnym do wykonania niezbędnych obliczeń. Na etapie odtwarzania wiadomości wykorzystywana jest macierz Vandermonda.(**Progowy schemat podziału dużych plików**)*

Keywords: threshold secret sharing scheme, multi-secret scheme, Shamir's scheme, large secret files distribution.
Słowa kluczowe: schemat podziału progowego sekretu, schemat wielosekretowy, schemat Shamira, dystrybucja dużych tajnych plików

doi:10.12915/pe.2014.02.28

1 Introduction

Secrets, such as cryptographic keys, need a special protection. In these specific cases a secret m is partitioned into n shares (shadows) which are shared among n members (shareholders) belonging to the set P. The secret sharing is implemented in such a way that in order to reconstruct the secret the knowledge of a threshold number of shares is necessary. In other words, the secret can be reconstructed only by a certain group of authorized shareholders $A \subseteq P$. Any unauthorized group should not be able to reconstruct this secret. A collection of all subsets of P, which contain users only authorized to reconstruct the secret is usually described as the access structure of the secret sharing scheme.

Secret sharing schemes have been independently introduced by G. R. Blakley [1] and A. Shamir [2] to effectively solve the problem of cryptographic keys protection. In both schemes, denoted as (t, n), the secret m is distributed among n participants in such a way that to its reconstruction the knowledge of at least t shadows is required, i.e. the cardinality of a set of authorized participants $A \subseteq P$ must be greater than or equal to t ($|A| \geq t$). In Shamir's scheme the Lagrange interpolating polynomial is used to reconstruct the secret, while in Blakley's scheme - the method of linear projective geometry. These types of schemes allow the subject named dealer to distribute the secret m (in the form of shadows) between n entities and next reconstruct it when the dealer receives at least t shadows.

In practice, however, it may be necessary to generate shadows on a base of long secret, which may be treated as the set of the secrets $M = \{m_1, m_2, ..., m_k\}$. The simplest, naive solution is to use (t, n) threshold scheme repeatedly for each subsecret $m_i \in M$. Such an approach is inefficient because of the need to generate shadows, whose number is a multiple of shadows for a single secret (e.g. a multiple of k in the case of k secrets) [3]. In order to avoid this problem so called multi-secret sharing schemes are used. In these schemes multiple secrets are protected using the same amount of data usually needed to protect a single secret. According to W. A. Jackson, et al. [4] multi-secret sharing schemes can be divided into two classes: onetime use schemes and multiple use schemes. In onetime use scheme after each secret reconstruction the secret holder must change the sharing scheme for new secrets (because a secret holder should usually assumed that each secret reconstruction means that it becomes publicly known), and next redistribute fresh shadows to every participant. Multiple use schemes are devoid of this defect , because every participant needs to keep only one secret shadow, even when there is a need for distribution/redistribution of new secrets (i.e., many secrets can be shared independently without refreshing the secret shadow).

There are several schemes designed especially for multi-secrets distribution (e.g., see N. Y. Lee, et al. [5], J. He and E. Dawson [6, 7], H. Y. Chien, et al. [8], C. C. Yang, et al. [9] and Y. J. Geng, et al. [10]). Generally, all these schemes are based on one-way functions (L. Gong [11]). A few schemes allow secret reconstruction only in stage-by-stage way (reconstruction in predetermined order), others provide mechanisms for simultaneous reconstruction of multi-secret. Among the secret sharing schemes of this latter type most interesting are proposals of H. Y. Chien, et al. [8], C. C. Yang, et al. [9] and L. J. Pang, et al. [12].

One of this schemes variant, where the secrets are reconstructed simultaneously, is the solution described by the H. X. Xian, et al. [13]. The sharing of k secrets requires in this scheme only one n-th degree polynomial (the size of n does not depend on the number of secrets), which is used to distribute the first secret m_1. For other secrets only the values of $\forall i \in \langle 2, k \rangle$: $m'_i = m_1 \oplus m_i$ is published. When the secret m_1 is reconstructed then it is easy to find the remaining $k - 1$ secrets. It should be noted that in this secret scheme the first secret m_1 is used k-times, so to make the cryptanalysis harder all secrets should be about the same size in bits.

All above schemes can be applied to any set of secrets M, whose cardinality is not a big order, and the secrets are not repeated. These conditions are difficult to meet for files containing, for example, electronic documents. File sizes are usually large and it is difficult to guarantee that certain fragments of the document, such as headers of business documents, will not be repeated and publicly known. Direct application of the above multi-secret sharing schemes requires the sharing of the file into g groups, each containing k elements (corresponding to k secrets of set M). With this approach, each group of k elements must be treated as an independent secret, for which another secret sparing polynomial should be generated. In such a case thresholdness property is kept in relation to single groups only, not to whole complex secret. In practice the reconstruction of k elements (even belonging to different groups) allows for the reasoning about the next groups and their components (e.g., based on a known message format or its well-known header).

Another threat is the possibility to change the proper order of the published values related to the different groups. Replacing the order of these values may prevent proper reconstruction of the file content. It is important to note that mentioned above multi-secret sharing schemes are not resistant for such a manipulation.

The rest of this paper is organized as follows. In Section 2 is briefly described a generalization of Shamir scheme, which is the basis for most of the known multi-secret sharing schemes, including our proposal. Section 3 contains the detailed explanation of our schema, i.e. basic assumption and step-by-step description of shadows generation process (including also the obfuscation mechanism). Description of the reverse process – the reconstruction of a secret on the basis of the authorized set of shadows - was described in Section 4. Final conclusions can be found in Section 5.

2 Generalization of Shamir's Threshold Secret Sharing Scheme

Let t_p is the privacy threshold describing the maximum number of shareholders that cannot determine the secret. On the opposite side we define t_f - the fault-tolerance threshold - the minimum number of shareholders that are needed to recover the secret. For a basic version of the Shamir's scheme the difference between t_f and t_p is 1. A (t, n) secret sharing scheme is a set of two functions, a sharing function and a recovering function. The probabilistic sharing function takes as an input the secret belonging to some finite set of secrets M and returns for this secret n shares as an output. In turn, the recovering function is a deterministic algorithm which recreates the message from some or all of the shares.

Assume F_p is a finite field of order p, where $p > n$ is a large prime number. Assume also that the fault-tolerance threshold should be equal to $t = t_f = t_p + u$. Value $u \geq 1$ denotes the number of secrets to share at the same time (belonging to the set M). Selected secrets create the set $\{m_1, \dots, m_u\} \subseteq M$. We randomly choose $t - u$ coefficients $\{\beta_1, \dots, \beta_{t-u}\} \subseteq F_p$ and creates random polynomial of $t - u - 1$ degree $Q(x) = \beta_1 x^u + \beta_2 x^{u+1} + \dots + \beta_{t-u} x^{t-1} \pmod{p}$. Finally we define generalized Shamir's polynomial of t-1degree, which may be used to share u secrets:

$$(1)\quad h(x) = m_1 + m_2 x + ... + m_u x^{u-1} + Q(x) (\bmod p)$$

It is necessary to ensure the confidentiality of coordinates for which the shares are generated on the basis of above polynomial. Let's recall, due to this type of the requirement, the definition of a two-variable one-way function (see, J. He, E. Dawson [6] and C. C. Yang, et al. [9]).

Definition 1 The two-variable one-way function $f(r, v)$ is a function that maps any r and v onto a bit string $f(r, v)$ of a fixed length. This function has the following properties:

(a) given r and v, it is easy to compute $f(r, v)$,

(b) given v and $f(r, v)$, it is hard to compute r,

(c) having no knowledge of v, it is hard to compute $f(r, v)$ for any r,

(d) given v, it is hard to find two different values r_1 and r_2 such that $f(r_1, v) = f(r_2, v)$,

(e) given r and $f(r, v)$, it is hard to compute v,

(f) given pairs of r_i and $f(r_i, v)$, it is hard to compute $f(r', v)$ for $r' = r_i$.

Suppose then, that the secret holder randomly chooses n values $v_1, v_2, ..., v_n$ and distributes them amongst them over a secret channel. Generated values $V = \{v_1, v_2, ..., v_n\}$ are the secret shares and allow each of i-th authorized participant to calculate the value of the function $f(r, v_i)$. After the distribution of V, the secret shadows holder executes the following steps:

(a) randomly chooses an integer r and for $(t$-1$)$-th degree polynomial $h(x)$ from Equation (1) compute $y_i = h(f(r, v_i)) \bmod p$ for $i = 1, 2, \dots, n$,

(b) publishes $\{r, y_1, y_2, ..., y_n\}$ in any authenticated manner, e.g. using Merkle tree (generally, as authenticated dictionary, see for example M. T. Goodrich [5]).

In order to reconstruct the group of secrets $\{m_1, \dots, m_u\}$ at least t shareholders should deliver their pseudo shadow $f(r, v_i)$. Using a $(t-1)$-th degree Lagrange interpolation polynomial, the knowledge of t pairs $(f(r, v_i), y_i)$ is sufficient to determine all coefficients of the polynomial (1) and to reconstruct secrets.

3 Our scheme

Suppose we are given a large size file, which will be treated further as a plain message M, the content of which should remain secret. The secret holder chooses arbitrarily large prime number p, being the degree of the finite field F_p. Next, a public message is divided into k blocks $M = \{m_1, \dots, m_k\}$, each of the size less then p size in bits $\forall i \in \langle 1,k \rangle$: $m_i < p$. Each block should be treated as a separated sub-secret. Note that the number of blocks (k value) is large. Message should be prepared before the final sharing – modification covers message expansion, blocks linking and permutation.

3.1 Message obfuscation

The secret holder generates a random number r from F_p, which will be used later for the two-variable one-way functions. Next the ring Z_q is chosen for Rivest permutation polynomial (see R. L. Rivest [15]), where $q = 2^w$. The value of the variable w is the smallest value that satisfies the inequality $2^w \geq k + 1$, namely $w = \lceil \log_2(k + 1) \rceil$.

The secret holder expands the message M to the message L complementing it with random values to q blocks: $L = \{l_0, \dots, l_{q-1}\}$, where $l_0 = k$, $\forall i \in \langle 1, k \rangle$: $l_i = m_i$ and $\forall i \in (k,q)$: $l_i \in F_p$ (random values from F_p).

All blocks of expanded message L are linked in a form of chain. XOR operation and the two-variable one-way function f are used for this purpose. The process starts from the last block with a randomly chosen initialization vector IV. All linked blocks create together the new message $O = \{o_0, \dots, o_{q-1}\}$, where $o_{q-1} = l_{q-1} \oplus f(r, IV)$ and $\forall i \in \langle 0, q–2 \rangle$: $o_i = l_i \oplus f(r, o_{i+1})$.

Then the d-th degree permutation polynomial is arbitrarily chosen: $P(x) = a_0 + a_1 x + \dots + a_d x^d \pmod{q}$. The secret holder generates random coefficients $\{a_0, \dots, a_d\}$ which satisfy the Rivest's conditions: $a_1 (\bmod 2) = 1 \wedge (a_2 + a_4 + a_6 + \dots)(\bmod 2) = 0 \wedge (a_3 + a_5 + a_7 + \dots)(\bmod 2) = 0$. This particular polynomial $P(x)$ is used to change the order of message O blocks. Permuted blocks creates a new message $C = \{c_0, \dots , c_{q-1}\}$, where $\forall i\in\langle 0, q)$: $c_i = o_{p(i)}$.

First of all the obfuscation technique presented above is used to protect the correct order of message blocks. Next purpose is to eliminate possibility of partial message recovery, which could be considered as a violation of thresholdness property on the level of the entire message M (separately recovered blocks potentially could allow inference about content of other blocks). Such a solution ensures that the reconstruction of any part of the message M without reconstruction of all others parts is impossible. This technique is presented on figure 1 and is realized in three consecutive stages: an expansion, a linking and a permutation. The resulting encoded message C is then partitioned into shadows and distributed as specified below.

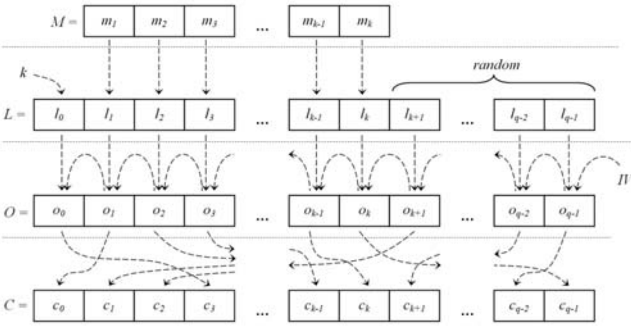

Fig.1. Obfuscation technique used for large message M.

3.2 Gathering block into groups

The secret holder divides prepared message C into g disjoint groups $C = C_1 \cup C_2 \cup \dots \cup C_g$. Each group consists of the same number of following blocks: $\forall i\in\langle 1, g\rangle$: $z = |C_i| = \lceil q / g \rceil$, maybe except the last group $|C_g| = q - |C_1|(g-1)$. The content of single group is denoted as follows: $\forall i\in\langle 1, g\rangle$: $C_i = \{b_{i,0}, b_{i,1},\dots , b_{i,z-1}\}$, where $\forall i\in\langle 1, g\rangle$, $\forall j\in\langle 0, z)$: $b_{i,j} = c_{(i-1)z+j}$.

For each i-th group the separate polynomial of degree $t-1 > z_i$ should be generated to distribute secrets. The structure of this polynomial corresponds to the polynomial of Equation (1) for u=z.

$$(2)\ B_i(x) = b_{i,0} + b_{i,1}x + \dots + b_{i,z_i}x^{z_i} + \dots + b_{i,t-1}x^{t-1} (\mathrm{mod}\, p)$$

The coefficients $\{b_{i,z},\ b_{i,z+1},\dots\ ,b_{i,t-2}\}$ of the distribution polynomial (2) are randomly chosen from the F_p. Optionally the highest coefficients $b_{i,t-1}$ may be chosen in other way. We should store and keep in secret coefficients of permutation polynomial and coefficients $b_{i,t-1}$ of distribution polynomial may be treated as good place for hiding of this information. In such a situation the coefficients of permutation polynomial are treated as secrets and split into shadows which is hidden in $b_{i,t-1}$ (see Equations (3) and (4)). This requirement means in practice that before we are able to recover the message we must first obtain information about the coefficients of the permutation polynomial. Given the above, for the j-th coefficient of permutation polynomial (where $j\in\langle 0, d-1\rangle$) the coefficients at the highest power of distribution polynomial are calculated according to the following scheme:

$$(3)\ \begin{cases} b_{je+1,t-1} \overset{R}{\leftarrow} F_p, \\ b_{je+2,t-1} \overset{R}{\leftarrow} F_p, \\ \dots, \\ b_{(j+1)e,t-1} = b_{je+1,t-1} \oplus b_{je+2,t-1} \oplus \dots \oplus a_j \end{cases}$$

and

$$(4)\ \begin{cases} b_{de+1,t-1} \overset{R}{\leftarrow} F_p, \\ b_{de+2,t-1} \overset{R}{\leftarrow} F_p, \\ \dots, \\ b_{g,t-1} = b_{de+1,t-1} \oplus b_{de+2,t-1} \oplus \dots \oplus a_d \end{cases}$$

where the value of e is determined from the following equation:

$$(5)\ e = \left\lfloor \frac{g}{d+1} \right\rfloor$$

Note that the last coefficient a_d of the permutation polynomial will be used to generate the coefficients at the highest power of the distribution polynomials starting from $B_{de+1}(x)$ up to $B_g\,(x)$. All coefficients calculated according to formulas (3) and (4) form a set $\{b_{1,t-1}, b_{2,t-1},\dots ,b_{g,t-1}\}$.

3.3 Shadows generation and distribution

Shadows are generated for $n > t$ shareholders. For each i-th participant the secretholder generates a random value v_i and distributes over a secret channel. These values can be generated for a single session (for one message distribution), as well as for several successive sessions.

Next, the secret holder calculates the shadows for each group. The shadow $y_{i,j}$ calculated for the i-th group and j-th participant has the following form:

$$(14)\ \forall i \in \langle 1,g\rangle, \forall j \in \langle 1,n\rangle, y_{i,j} = B_i(\,f(r,v_j\,))$$

where f(...) is the two-variable one-way function. Every j-th participant receives the values $\{t, g, d, q, m, r, Y_j\}$ over an unsecured channel, where $Y_j = \{y_{1,j}, \dots, y_{g,j}\}$.

On the basis of information received from the secret holder any participant is able to prepare the final form of the shadows $S_j = \{s_{1,j}, \dots, s_{g,j}\}$, where $s_{i,j} = \{x_{i,j} = f(r, v_j), y_{i,j}\}$. As the result the following array of shadows is potentially available (distributed amongst all participants):

$$(6)\, S = \begin{bmatrix} S_1 \\ \vdots \\ S_n \end{bmatrix} = \begin{bmatrix} s_{1,1} & \cdots & s_{g,1} \\ \vdots & \ddots & \vdots \\ s_{1,n} & \cdots & s_{g,n} \end{bmatrix}$$

In summary, each j-participant keeps in secret S_j vector and some additional information $\{t, g, d, q, m, r, Y_j\}$.

4 Message reconstruction

It is obvious that at least t shadows for each group is required to reconstruct complete message. We may assume that shadows for each reconstructed group are collected from the same participants for simplicity, but it's not necessary. As an analogy to matrix S (see Equation (6)) we can present collected shadows in the form of the matrix S':

$$(7)\ S' = \begin{bmatrix} s'_{1,1} & \cdots & s'_{g,1} \\ \vdots & \ddots & \vdots \\ s'_{1,t} & \cdots & s'_{g,t} \end{bmatrix}$$

where $s'_{i,j} = \{x'_{i,j} = f(r, v_i), y'_{\mathrm{i},j}\}$. The shadows can be always delivered by the authorized group of t ($t \le |A|$) participant belonging to the set $A \subseteq P(\{1, 2, \dots, n\})$.

4.2 Reconstruction of distribution polynomials

Using the matrix S' we create the Vandermonde matrix for each i-th group:

$$(8)\ \forall i \in \langle 1,g \rangle : V_i = \begin{bmatrix} 1 & x'_{1,1} & (x'_{1,1})^2 & \cdots & (x'_{1,1})^{t-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x'_{1,t} & (x'_{1,t})^2 & \cdots & (x'_{1,t})^{t-1} \end{bmatrix}$$

which allows us to reconstruct a set of coefficients of the distribution polynomial for each group of messages $\forall i \in \langle 1, g \rangle$: $\{b'_{i,0}, b'_{i,1}, \ldots, b'_{i,t-1}\} = V_i^{-1} \cdot \{y'_{i,1}, y'_{i,2}, \ldots, b'_{i,t}\}$.

4.3 Reconstruction of the permutation polynomial

From the set of reconstructed coefficients of $(t$-$1)$-th degree distribution polynomial (see Equation (2)) we choose those that form the set $\{b'_{1,t-1}, b'_{2,t-1}, \ldots, b'_{g,t-1}\}$ containing coefficients standing at the highest power of this polynomial. These coefficients can be used to reconstruct a permutation polynomial. On the basis of the values of g and d the value of e' is calculated (see Equation (5)), and then the permutation polynomial $P'(x) = a'_0 + a'_1 x + \ldots + a'_d x^d \pmod q$ reconstructed:

$$(9)\ \forall j \in \langle 0, d-1 \rangle, a'_j = b'_{je'+1,t-1} \oplus \ldots \oplus b'_{(j+1)e',t-1}$$

$$(10)\ a'_d = b'_{jd+1,t-1} \oplus b'_{de'+2,t-1} \oplus \ldots \oplus b'_{g,t-1}$$

4.4 Reconstruction of the original message

For the given values of g and q we compute the cardinality of the first $(g$-$1)$ groups of messages (this cardinality is equal to $z' = \lceil q\ /\ g \rceil$, and then rebuild an obfuscated message by making it to be equal to the appropriate coefficients of the distribution polynomials $C' = \{c'_0, \ldots, c'_{q-1}\}$, where

$$\forall i \in \langle 0, q-1), c'_i = b'_{\left\lceil \frac{i+1}{g} \right\rceil, i \pmod{z'}}$$

Now, using reconstructed permutation polynomial $P'(x)$ it is possible to restore the original order of message blocks $O' = \{o'_0, \ldots, o'_{q-1}\}$ where $\forall i \in \langle 0, m), o'_i = c'_{p'^{-1}(i)}$ and to suppress the linking introduced in the first stage of the message obfuscation $L' = \{l'_0, \ldots, l'_{q-1}\}$ while $l'_{q-1} = o'_{q-1} \oplus f(r, IV)$ and $\forall i \in \langle 0, q-2 \rangle$: $l'_i = o'_i \oplus f(r, o'_{i+1})$.

Block l'_0 is nothing more than the number of blocks in a plain message before its enlargement, so finally we get the reconstructed message $M' = \{l'_1, l'_1, \ldots, l'_{k'}\}$ where $k' = l'_0$.

5. Conclusions

In the paper, we propose a new (t, n)-threshold multi-secret and multi-use sharing scheme based on Shamir's generalized scheme. Unlike other known solutions the proposed scheme can be used primarily to spreading very large files (messages) between the n shareholders. The files do not usually contain the well formed secrets, and this property means that the information in the file may be repeated. In this case, the privacy threshold can be equal to $t_p = t - z$, where **z** is the number of well-known parts of some or all information groups created for a large secret file (see Section 3).

In our scheme, the obfuscation and linking mechanisms prevent the reduction of the privacy threshold, because these mechanisms allow to recover the file content only after its entire reconstruction. Since this is possible only if at least t authorized holders will share their pseudo-shares, therefore the privacy level for each recovered group of secrets has value $t_p = t$ - 1 and the fault-tolerance threshold is equal to the threshold value, i.e. $t_f = t$. This means that the overall privacy level for the large file with *g* groups of secret is equal $t_{ptotal} = gt$-1, while $t_{ftotal} = gt$. Note, that these values are optimal like for the (gt, gn) multi-secret sharing scheme.

Acknowledgment

This scientific research work is supported by NCBiR of Poland (grant No PBS1/B3/11/2012) in 2012-2015.

REFERENCES

[1] Blakley, G. R.: Safeguarding cryptographic keys. w: Proc. AFIPS 1979 National Computer Conference, AFIPS, pp. 313--317 (1979)

[2] Shamir, A.: How to share a secret. Communication of the ACM 22, pp. 612--613 (1979)

[3] Crescenzo, G.D.: Sharing one secret vs. sharing many secrets. Theoretical Computer Science, No. 295, pp. 123--140 (2003)

[4] Jackson, W. A., Martin, K.M., O'Keefe, C.M., *On sharing many secrets (extended abstract)*, ASIACRYPT 1994, LNCS, vol. 917, pp. 42--54. Springer, Heidelberg (1995)

[5] Lee, N.-Y., Hwang, T., *New Multistage Secret Sharing Scheme Based on the Factorization Problem*, Journal of Information Science and Engineering, no 17, pp. 525—529 (2001)

[6] He, J., Dawson, E., *Multistage secret sharing based on one-way function*, Electronics Letters, Vol. 30, No 19, pp. 1591--1592 (1994)

[7] He, J., Dawson, E., *Multisecret-sharing scheme based on one-way function*, Electronics Letters, Vol. 31, No 2, pp. 93--95 (1994)

[8] Chien, H.-Y., Jan, J.-K., Tseng, Y.-M.: A practical (t, n) multi-secret sharing scheme. IEICE Transactions on Fundamentals E83-A (12), pp. 2762--2765(2000)

[9] Yang, C.-C., Chang, T.-Y., Hwang, M.-S., *A (t,n) multi-secret sharing scheme*, Applied Mathematics and Computation, Volume 151, Issue 2, pp. 483--490 (2004)

[10] Geng, Y.-J., Fan, X.-H., Hong, F.: A new multi-secret sharing scheme with multi-policy. The 9th International Conference on Advanced Communication Technology, Vol. 3, pp. 1515--1517 (2007)

[11] Gong, L., *New protocols for third-party-based authentication and secure broadcast*, in 2nd ACM Conference on Computer and Communications Security, pp. 176–183, ACM Press, New York (1994)

[12] Pang, L.-J., Wang, Y. M., *A new (t,n) multi-secret sharing scheme based on Shamir's secret sharing*, Applied Mathematics and Computation, Volume 167, Issue 2, pp. 840--848, (2005)

[13] Li, H.-X., Cheng, C.-T., Pang, L.-J., *A New (t, n)-threshold Multi-secret Sharing Scheme*, CIS2005, Berlin, Heidelberg, New York: Springer-Verlag, pp.421--426 (2005)

[14] Goodrich, M. T., Tamassia, R., Triandopoulos, N., Cohen, R., *Authenticated* data *structures for graph and geometric searching*, LNCS, Vol, 2612, Springer-Verlag, pp. 295--313(2003)

[15] Rivest, R. L., *Permutation polynomials modulo 2^w*, Finite Fields and Their Applications, Vol. 7, pp. 287--292 (2001)

***Authors**:*
Imed El Fray, PhD eng., e-mail: ielfray@wi.zut.edu.pl; Tomasz Hyla, PhD eng., e-mail: thyla@wi.zut.edu.pl, *Witold Maćków, PhD eng., e-mail: wmackow@wi.zut.edu.pl*; *Jerzy Pejaś, PhD eng., e-mail: jpejas@wi.zut.edu.pl; West Pomeranian University of Technology in Szczecin, Faculty of Computer Science and Information Technology, 52 Żołnierska St., 71-210 Szczecin*