Jacek KRYŃSKI, Witold MAĆKÓW
WEST POMERANIAN UNIVERSITY OF TECHNOLOGY, FACULTY OF COMPUTER SCIENCE,
ul. Żołnierska 49, 71-200 Szczecin

Long Message Threshold Scheme with a Low Degree Polynomial

M.Sc. Jacek KRYŃSKI

Ph.D. student of the Software Technology Department of the West Pomeranian University of Technology, Szczecin. He received M.Sc. degree in Computer Science and Information Technology with major in programming techniques in 2010. His interest include: secret sharing methods and pairing based cryptography on bilinear maps.

e-mail: jkrynski@wi.zut.edu.pl

Ph.D. eng. Witold MAĆKÓW

W. Maćków (1974) received M.Sc. degree in 1998 and Ph.D. degree in 2007 from Faculty of Computer Science and Information Technology, Technical University of Szczecin (at present West Pomeranian University of Technology, Szczecin). His scientific interests include public key infrastructure application, long term archive systems, authenticated data structures, threshold secret sharing and still image steganography.

e-mail: wmackow@wi.zut.edu.pl

Abstract

The standard threshold sharing schemes applied directly to large secret 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. For these reasons extended methods are used to share large secrets, methods belonging to class of so called multi-secret threshold schemes. Most of them are based on generalized Shamir's scheme. The paper introduces a new threshold secret sharing scheme belonging to the mentioned class. An efficiency of our solution is comparable to other analyzed solutions based on generalized Shamir's scheme while degree of interpolation polynomial is decreased. The performance of the implemented method was additionally compared with the implementation of method using 3DES encryption and classic Shamir scheme to share the encryption key.

Keywords: threshold secret sharing scheme, multi-secret scheme, Shamir's scheme.

Podział progowy długich wiadomości z wykorzystaniem wielomianów niskiego stopnia

Streszczenie

Standardowe schematy podziału sekretu stosowane bezpośrednio do długich wiadomości są nieefektywne i potencjalnie niebezpieczne. Ich niska efektywność wynika głównie z konieczności generowania i przechowywania dużej ilości cieni. Z kolei niski poziom bezpieczeństwa jest następstwem nieuwzględniania właściwości długich wiadomości, takich jak format pliku czy powtarzalność fragmentów informacji w niej zawartej. Z tego powodu w praktyce do podziału długich wiadomości stosuje się metody rozszerzone, tzw. wielosekretowe schematy podziału. Większość z nich oparta jest na uogólnionym schemacie Shamira. W artykule zaproponowano nowy schemat podziału należący do tej klasy metod. Wydajność proponowanego rozwiązania jest porównywalna z innymi przeanalizowanymi rozwiązaniami wykorzystującymi uogólniony schemat Shamira, natomiast wyraźnie niższy jest stopień wykorzystywanych w schemacie wielomianów interpolacyjnych. Efektywność zaimplementowanej metody została dodatkowo porównana z metodą hybrydową wykorzystującą szyfrowani 3DES i klasyczny schemat Shamira do podziału klucza.

Słowa kluczowe: schemat podziału progowego sekretu, schemat wielosekretowy, schemat Shamira.

1. Introduction

A group of cryptographic protocols called threshold sharing schemes are generally used for distributing a secret amongst entities (shareholders). In such a scheme each shareholder receives different shares (also called the shadow). Collecting a set of shadows allows for recovering the secret. We are talking about (k, n) threshold sharing model, where k indicates threshold number of shadows needed for secret reconstruction and n indicates number of shareholders. Usually secret is a short message, very often some kind of cryptographic primitive, e.g. key.

There are many classic solutions of this problem based on the different mathematical foundations. KGH method (names of originators: Karnin, Green and Hellman) [5] allows to create (n, n) model, where all shadows are required to recover the secret. Most popular variant of this method is one using bit vectors and addition modulo 2 operations. Shamir scheme [12] is based on Lagrange polynomial and supports model (k, n). This method is widely used in practical applications thanks to its simplicity and efficiency. Chinese Remainder Theorem is used in Mignotte's [9] and Asmuth-Bloom's [1] schemes. Yet another approach for secret sharing presents Blakley scheme [2], which make use of n-dimensional hyperplanes intersections.

2. Previous works

New applications of the threshold sharing schemes resulted in extending the functionality of the basic methods. New ideas appear successively, probably most important are following:

- verifiable secret sharing schemes – shareholders are able to verify the correctness of shadows received from dealer [11]; schemes are called publicly verifiable when information needed for verification are commonly available [13];
- proactive secret sharing schemes – shadows are periodically renewed [7] to increase the long-term security of the secret;
- multi-secret sharing schemes – the secret is a long message, usually splited into small sub-secrets; standard threshold sharing schemes applied directly to large secret are ineffective; most of multi-secret sharing schemes uses generalized Shamir's scheme.

In this paper we present the new method of multi-secret threshold sharing based on generalized Shamir's scheme. Our work was prepared under strong influence of papers mentioned below.

Yang et al. [14] proposed a (k, n) multi-secret sharing scheme based on Shamir's secret sharing. It used ($n + p - k + 1$) public values, $2(k - 1)$ or $2(p - 1)$ storages and employed the Lagrange interpolation polynomial to share p secrets. This scheme was inspired by Chien et al.'s scheme [3], but it improved secret recovery efficiency (originally it was necessary to solve simultaneous complex equations). Unfortunately there were more public values required in Yang et al.'s scheme than in Chien et al.'s scheme when $p < k$.

Pang et al. [10] proposed new (k, n) multi-secret sharing scheme, which is as easy as Yang et al.'s scheme in the secret reconstruction and requires the same number of public values as Chien et al.'s scheme. In this scheme the degree of the Lagrange polynomial is dynamic and equals ($n + p - 1$), what results in gradual increase of interpolation complexity. Li et al. [8] noticed this problem and proposed an alternative (k, n)-threshold multi-secret sharing scheme based on Shamir's secret sharing scheme, which uses a fixed n-th degree Lagrange interpolation polynomial and has the same power as Pang et al.'s scheme.

3. Proposal of new scheme

The proposed solution uses some ideas of the previously mentioned methods: Yang et al.'s [14], Pang et al.'s [10] and Li et al.'s [8]. It saves their simplicity and efficiency, uses the same number of public parameters but the polynomial degree is decreased (it depends on threshold value, not on p like in a Pang et al.'s scheme). Our proposal belongs to multi-secret threshold sharing scheme class, so we decided to focus on cases when $p > k$. Scheme requires the publication of certain parameters. Some of them stay unchanged through whole scheme lifecycle (e.g. user id's), while others are generated independently for each single session (session means complete process of entire multi-secret sharing and recovering). Shadows should be regenerated for each session too.

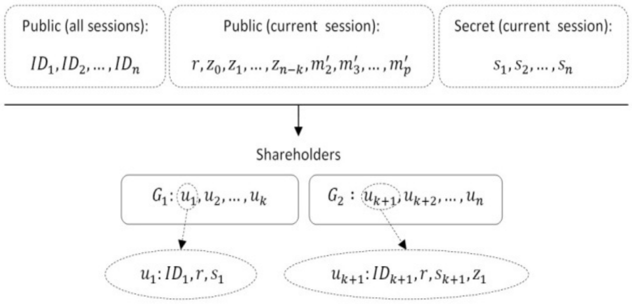
Public (all sessions):
$ID_1, ID_2, \dots, ID_n$
Public (current session):
$r, z_0, z_1, \dots, z_{n-k}, m'_2, m'_3, \dots, m'_p$
Secret (current session):
$s_1, s_2, \dots, s_n$
Shareholders
$G_1: u_1, u_2, \dots, u_k$
$G_2: u_{k+1}, u_{k+2}, \dots, u_n$
$u_1: ID_1, r, s_1$
$u_{k+1}: ID_{k+1}, r, s_{k+1}, z_1$

Fig. 1. Parameters used during secret sharing (single session)

Rys. 1. Parametry używane podczas podziału sekretu (pojedyncza sesja)

Fig. 1 shows main parameters of our scheme, dividing them into public, private, persistent (for all session) and current sessions related. All denotations are explained in section 3.1, in the description of the sharing algorithm.

3.1. Main concept

The proposed method consists of two algorithms: sharing algorithm and reconstructing algorithm. A detailed description of both algorithms is presented below. Following denotations are introduced: m_i – i-th sub-secret of complex multi-secret m, u_i – i-th shareholder, G_i – i-th group of shareholders, s_i – i-th shadow, ID_i – public identifier of i-th shareholder, q – order of finite field (all parameters are elements of this field), p – number of sub-secrets in complex multi-secret, n – number of shareholders, k - threshold, r – random parameter used for generating pseudo-shadows (session related), $f(\dots)$ – two-variable hash function, $W(\dots)$ – Lagrange polynomial, z_i – auxiliary public parameter (value of polynomial in specific point), m_i' – auxiliary public parameter (specific for i-th sub-secret).

Sharing algorithm:

- Choose a big prime $q > m_1, m_2, \dots m_p$. All following parameters are elements in the finite field $GF(q)$.
- Divide shareholders into two groups $G_1=\{u_1,u_2,\dots u_k\}$ and $G_2=\{u_{k+1},u_{k+2},\dots u_n\}$. This division may be random.
- Choose random shadows $s_1, s_2, \dots s_n$ distribute them among all shareholders in a secure manner.
- Choose n different integers $ID_1, ID_2, \dots ID_n$ satisfying the condition: $1 \le ID_1, ID_2, \dots ID_n < q$. Each of these values will be treated as a public identifier of individual shareholder.
- Choose the random parameter r and calculate n pseudo-shadows $\forall i \in <1,n>, f(r,s_i)>$. If there is a conflict (any calculated values are the same, or any calculated values are equal 0 or 1) the new r should be drawn and pseudo-shadows should be recalculated once more.
- Create Lagrange interpolating polynomial $W(x)$ of degree k using $(k+1)$ pairs of arguments and values presented: $\{(0, m_1), (f(r,s_1), ID_1), (f(r,s_2), ID_2), \dots, (f(r,s_k), ID_k)\}$. Polynomial is created on the base of information related only to shareholders from first group G_1. Any and all arguments stay secret.
- Using prepared polynomial calculate $z_0=W(1)$.
- Using prepared polynomial calculate $(n-k)$ values $\forall i \in (k,n), z_{i-k}= W(f(r,s_i))$ for shareholders grouped in G_2.
- Calculate$(p-1)$ values $\forall i \in (1,p), m_i'=(m_1 \oplus m_i) \bmod q$
- Make public $(n+p-k+1)$ parameters: $r, z_0, z_1, \dots z_{n-k}, m_2', m_3', \dots, m_p'$.

Reconstructing algorithm:

- Collect k pseudo-shadows $\forall i \in <1,k>$, $f(r,s_i)$, from k different shareholders. Each shareholder is able to generate his pseudo-shadow on a base of held shadow and publicly know value r. The assumption is that g_1 pseudo-shadows are provided by shareholders from group G_1 and g_2 pseudo-shadows are provided by shareholders from group G_2 (finally $g_1+g_2 \ge k$).
- Reconstruct Lagrange interpolating polynomial $W'(x)$ on the base of arguments obtained in previous step and public values: $\{(f(r,s_1), ID_1), \dots, (f(r,s_{g_1}), ID_{g_1}), (1, z_0), (f(r,s_{g_1+1}), z_1), \dots, (f(r,s_{g_1+g_2}), z_{g_2})\}$.
- Calculate first subsecret $m_1=W'(0)$.
- Recalculate all remaining subsecrets $\forall i \in (1,p), m_i=(m_1 \oplus m_i') \bmod q$

3.2. Usage example

The example shows processes of data sharing and reconstructing based on our proposal. We shares multisecret m=123456789 consisting of three subsecrets: $m_1=(123)_{10}=(0001111011)_2$, $m_2=(456)_{10}=(0111001000)_2$ and $m_3=(789)_{10}=(1100010101)_2$. Subsecrets are concatenated. Other parameter are n=4, k=3 and p=3. We choose hash function $f(r,s)= (r+s)^2 \bmod 1081$ $(1081=23 \cdot 47)$ for the purpose of this example.

The data **sharing algorithm** used for above data looks as follows.

- We choose q=1123>123,456,789.
- We divide users into two groups: $G_1=\{u_1,u_2,u_3\}$, $G_2=\{u_4\}$.
- We generate random shadows: s_1=347, s_2=523, s_3=815, s_4=860.
- We choose users identifiers: ID_1=1, ID_2=2, ID_3=3, ID_4=4.
- We generate random parameter r=1020 for this session.
- We calculate pseudo-shadows: $f(r,s_1)=f(1020,347)=(1020+347)^2 \bmod 108=721$, $f(r,s_2)=f(1020,523)=487$, $f(r,s_3)=f(1020,815)=991$, $f(r,s_4)=f(1020,860)=661$.
- We uses following points: {(0, 123), (721, 1), (487, 2), (991, 3)} to interpolate polynomial of 3 degree: $W(x)=123+593x+350x^2+366x^3 \pmod{1123}$.
- We calculate rest of public parameters: $z_0=W(1)=309$, $z_1=W(661)=1021$, $m_2'=(m_1 \oplus m_2) \bmod 1123=(0111001000)_2$, $m_3'=(m_1 \oplus m_3) \bmod 1123=(1101101110)_2$.
- Finally we make public: r, z_0, z_1, m_2', m_3'.

To demonstrate data **reconstructing algorithm** we chose three of four pseudo-shadows: $f(r,s_1)$, $f(r,s_2)$ and $f(r,s_3)$. Following steps are done to reconstruct secret.

- We uses following points: {(721, 1), (487, 2), (1, 309), (661, 1021)} to interpolate polynomial of 3 degree: $W(x)=123+593x+350x^2+366x^3 \pmod{1123}$.
- We calculate first subsecret: $m_1=W(0)=(123)_{10}=(0001111011)_2$.
- We calculate other subsecrets: $m_2=(m_1 \oplus m_2') \bmod 1123 = (0110110011)_2$, $m_3=(m_1 \oplus m_3') \bmod 1123 =(1100010101)_2$.
- Finally we reconstruct whole secret: $m=m_1||m_2||m_3$=123456789.

3.3. Modification for short messages

Scheme presented in 3.1 can be simplified and adapted for the distribution of short messages, which need not be divided into subsecrets. The modification assumes that the first subsecret is the whole secret. The modified scheme does not require renewal of the shadows after secret changing.

3.4. Comparison

Our proposal belongs to multi-secret sharing schemes group and in our opinion most important was optimization of scheme for long secrets ($p>k$). Our scheme saves main features of alternate methods, amongst them their simplicity an efficiency. It uses same number of public parameters like other analyzed schemes (for $p>k$). Interpolation polynomial degree is clearly lower, even in relation to the best of analyzed methods (Li et al.'s). Summary of this parameters is presented in Tab. 1.

Tab. 1. Characteristics of selected schemes
Tab. 1. Charakterystyka wybranych schematów

Yang et. al [14]	Pang et. al [10]	Li et. al. [8]	Our proposal
Number of public parameters			
$(n+1)$ for $p\le k$	$(n+p-k+1)$	$(n+p-k+1)$	$(n-k+2)$ for $p\le k$
$(n+p-k+1)$ for $p>k$			$(n+p-k+1)$ for $p>k$
Polynomial degree			
$(k-1)$ for $p\le k$	$(n+p-1)$	n	k
$(p-1)$ for $p>k$			

The number of public parameters affect the amount of memory needed to store them and, indirectly, the number of operations related to them (generation, reading, writing). Public parameters therefore affect both the memory cost and the CPU cost. The polynomial degree directly affects only CPU cost (coefficients computation).

4. Results

Described threshold algorithm was implemented and tested. For comparison purpose alternative approach based on 3DES encryption was implemented also (Fig. 2). In this approach the long message m is encrypted using 192-bit key (concatenation of three 64-bit DES keys) and cryptogram c is made publicly available. The key is shared using Shamir's scheme into shadows which are delivered to users in a secure manner. The reconstruction of the message m requires in first step reconstruction of key and then decryption of cryptogram c. This scheme, as well as our proposal, does not require shadows re-generation for new secret (changes publicly available cryptogram only). The functionality of both implemented algorithms is therefore the same.

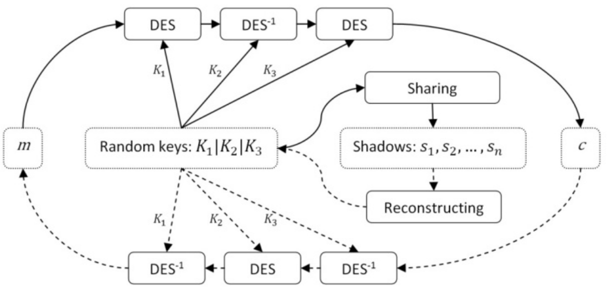

Fig. 2. Threshold sharing scheme of long message m based on 3DES
Rys. 2. Progowy schemat podziału długiej wiadomości m oparty na 3DES

Both implementations were realized In C language. GNU MP (Multiple Precision) library were used for big numbers. DES implementation was based on ready solution from GNU PG (Privacy Guard) library. Our proposal turned out faster than 3DES based solution. The time of secret sharing in both cases depended linearly on the size of the multi-secret, but our scheme was almost twice faster. The results are shown in Fig. 3.

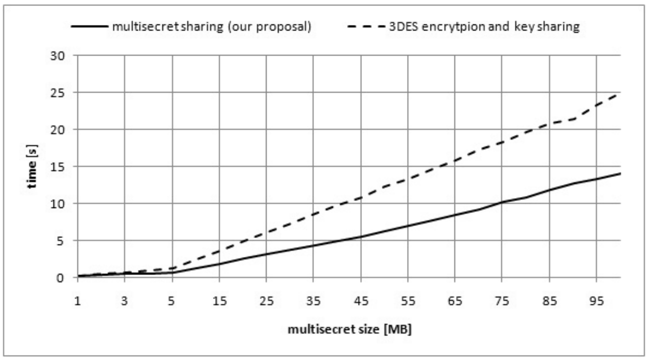

Fig. 3. Multi-secret sharing - efficiency comparison between proposed scheme and 3DES based scheme
Rys. 3. Podział sekretu złożonego – porównanie proponowanego schematu i schematu opartego na 3DES

The multi-secret reconstruction algorithms had similar efficiency like sharing algorithms and also linearly depended on multi-secret size. In this case proposed method based on Shamir's generalized scheme turned out faster too. The results are shown in Fig. 4.

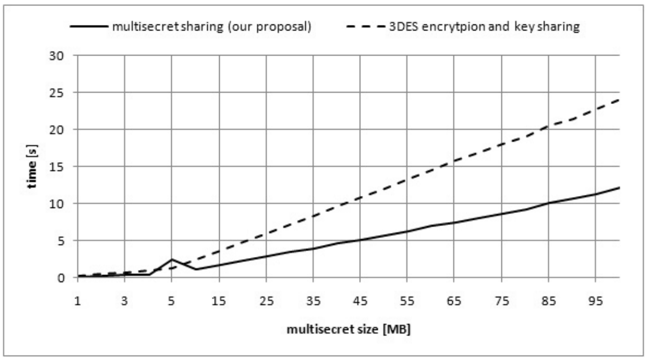

Fig. 4. Multi-secret reconstructing - efficiency comparison between proposed scheme and 3DES based scheme
Rys. 4. Odtwarzanie sekretu złożonego – porównanie proponowanego schematu i schematu opartego na 3DES

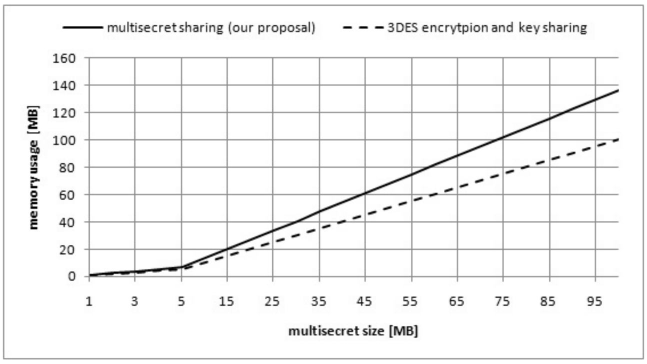

Fig. 5. Memory required for public parameters storage (comparison between proposed scheme and 3DES based scheme)
Rys. 5. Pamięć potrzebna do przechowywania parametrów publicznych (porównanie proponowanego schematu i schematu opartego na 3DES)

Fig. 5 shows the amount of memory allocated to store the public parameters of both algorithms. In this test a better solution was scheme based on the 3DES algorithm. Implementation of the proposed algorithm requires an average of almost 37% more memory to store the public parameters than alternate one.

5. Summary and Future Works

The paper provides a proposition of the novel threshold secret sharing scheme, intended to distribute a multi-secret. The algorithm presented in the paper was implemented in practice. Because the multi-secret is divided into groups (parts) of a fixed size, hence the time complexity of this algorithm depends mainly on the number of these groups.

In the proposed method and in other methods based on the generalized Shamir scheme, the problem is a large number of public parameters. This results in increased memory consumption and indirectly in decreased algorithm effectiveness (more operations are executed on these parameters). Obtaining a lower number of public parameters with the same functionality, probably requires to use not conventional mathematic approach to the subject.

The proposed sharing scheme is not resistant to reordering of generated values m_i', $\forall i \in (1,p)$ and to substituting them with other values. Reordering these values or simply replacing theirs values may prevent a proper content reconstruction.

These problems can be resolved using methods of m_i value linking (see [17]) and using verifiable secret sharing schemes (e.g. [6, 7]). It requires introducing slight modifications to the schema proposed in Section 3. These modifications are a subject of our ongoing and future works.

6. References

[1] Asmuth C. and Bloom J.: A modular approach to key safeguarding. IEEE Trans. On Information Theory IT-29 (1983), 208–211.

[2] Blakley G. R.: Safeguarding cryptographic keys. In: Proc. AFIPS 1979 National Computer Conference, AFIPS, 1979, 313–317.

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

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

[5] Green, J. W., Hellman, M. E. and Karnin, E. D.: On secret sharing systems. IEEE Trans. on Information Theory IT-29 (1983), 35–41.

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

[7] Herzberg A., Jarecki S., Krawczyk H., and Yung M.: Proactive secret sharing or how to cope with perpetual leakage. In: Advances in Cryptology—CRYPTO'95, Lecture Notes in Computer Science, Springer-Verlag, 1996, 339–352.

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

[9] Mignotte M.: How to share a secret. In T. Beth, editor, Cryptography-Proceedings of the Workshop on Cryptography, Burg Feuerstein, 1982, volume 149 of Lecture Notes in Computer Science, pages 371–375. Springer-Verlag, 1983.

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

[11] Chor B., Goldwasser S., Micali S. and Averbuch B.: Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults (Extended Abstract). FOCS85, pp. 383-395.

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

[13] Stadler M.: Publicly verifiable secret sharing. In: Advances in Cryptology EUROCRYPT'96. Lecture Notes in Computer Science, Springer-Verlag, 1997, 190–199.

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

otrzymano / received: 10.09.2010
przyjęto do druku / accepted: 01.11.2010 *artykuł recenzowany*

INFORMACJE

Newsletter PAK

Wydawnictwo PAK wysyła drogą e-mailową do osób zainteresowanych Newsletter PAK, w którym są zamieszczane:

- spis treści aktualnego numeru miesięcznika PAK,
- kalendarz imprez branżowych,
- ważniejsze informacje o działalności Wydawnictwa PAK.

Newsletter jest wysyłany co miesiąc do osób, które w jakikolwiek sposób współpracują z Wydawnictwem PAK (autorzy prac opublikowanych w miesięczniku PAK, recenzenci, członkowie Rady Programowej, osoby które zgłosiły chęć otrzymywania Newslettera).

Celem inicjatywy jest umocnienie w środowisku pozycji miesięcznika PAK jako ważnego i aktualnego źródła informacji naukowo-technicznej.

Do newslettera można zapisać się za pośrednictwem:

- strony internetowej: www.pak.info.pl, po dodaniu swojego adresu mailowego do subskrypcji,
- adresu mailowego: wydawnictwo@pak.info.pl, wysyłając swoje zgłoszenie.

Otrzymywanie Newslettera nie powoduje żadnych zobowiązań ze strony adresatów. W każdej chwili można zrezygnować z otrzymywania Newslettera.

Tadeusz SKUBIS
Redaktor naczelny Wydawnictwa PAK