Indonesian Journal of Combinatorics **3** (1) (2019), 14–23

INDONESIAN JOURNAL
OF COMBINATORICS

Implementation of super H-antimagic total graph on establishing stream cipher

Antonius Cahya Prihandoko[a], Dafik[b], Ika Hesti Agustin[b]

[a]*Faculty of Computer Science, University of Jember, Indonesia*
[b]*Combinatorics, Graph Theory, and Network Topology (CGANT), University of Jember, Indonesia*

antoniuscp.ilkom@unej.ac.id, d.dafik@unej.ac.id, ikahesti.fmipa@unej.ac.id

Abstract

This paper is aimed to study the use of super (a,d)-H antimagic total graph on generating encryption keys that can be used to establish a stream cipher. Methodology to achieve this goal was undertaken in three steps. First of all the existence of super (a,d)-H-antimagic total labeling was proven. At the second step, the algorithm for utilizing the labeling to construct a key stream was developed, and finally, the mechanism for applying the key to establish a stream cipher was constructed. As the result, according to the security analysis, it can be shown that the developed cryptographic system achieve a good security.

Keywords: Super H-antimagic total graph, keystream, encryption, stream cipher
Mathematics Subject Classification: 05C78
DOI: 10.19184/ijc.2019.3.1.2

1. Introduction

Cryptography is a popular solution to keep confidentiality of information that is spread over unsecure network. Confidential means that the information can only be revealed by authorized users. In this approach, information is firstly encrypted before it is distributed through the network. Legal decryption keys are given only to authorized users. Some encryption protocols for information

Received: 01 May 2019, Revised: 30 May 2019, Accepted: 15 Jun 2019.

security purposes have been proposed in many literatures [6, 7, 8]. The strength of these protocols relays on the encryption-decryption keys management. The keys must be kept secret and inaccessible to unauthorized users, as knowing information about the keys would enables someone to decrypt the protected information without constraint. Without a doubt, keeping the keys from being accessible to unauthorized users is the major consideration for many cryptographic protocols.

Research on the encryption-decryption keys management is continuously undertaken. The efforts are focused to achieve information confidentiality according to the required security level. In previous paper [9], we constructed encryption keys using super H-antimagic total labeling (SHATL). SHATL is used for this purpose because of the uniqueness of its construction but it still produces a number sequence that is difficult to see in its pattern. This condition is potential to achieve a good security of block based cryptosystem. The encryption keys produced from SHATL are then utilized to establish a block cipher. In this cipher, plaintext is split into blocks with the same length and is encrypted block by block. Unlike the ordinary block cipher, where each block is encrypted using the same key, in the scheme we develop, it is possible for different blocks are encrypted using different keys. However, many obstacles come out from this scheme. If the message is longer than the sequence yielded from SHATL, the sequence will be repeated for encrypting the remainder parts of the message. Morover, the longer the sequence produced by SHATL, the larger the storage capacity is needed.

This paper is aimed to solve the obstacles described above. While still utilizing SHATL, the keys construction is focused on establishing a stream cipher. In this scheme, SHATL is utilized to produce an initial key. The system then generates the subsequent keys based on the initial key and the defined stream function. This mechanism will reduce the possibility of key repetition. The different block will be encrypted by the different keys to achieve a stronger protection. Furthermore, the initial key is the only key that needs to be saved, so that this mechanism requires relatively small capacity of storage.

The rest of this paper is outlined as follows. Section 2 describes the concept of SHATL and proves its existence. Section 3 presents the mechanism of establishing a stream cipher using SHATL and section 4 provides the security analysis of the developed stream cipher.

2. Super H-antimagic Total Graph

A graph G is called to be an (a, d)-H-antimagic total graph if there exist a bijective function $f : V(G) \cup E(G) \rightarrow \{1, 2, \ldots, |V(G)| + |E(G)|\}$ such that for all subgraphs of G isomorphic to H, the total H-weights $w(H) = \sum_{v\in V(H)} f(v) + \sum_{e\in E(H)} f(e)$ form an arithmetic sequence $\{a, a + d, a + 2d, ..., a + (n - 1)d\}$, where a and d are positive integers and n is the number of all subgraphs of G isomorphic to H. In this case, the function f is then called an (a, d)-H-antimagic total labeling of G. Additionally, if $f : V(G) \rightarrow \{1, 2, \ldots, |V(G)|\}$, then the (a, d)-H-antimagic total labeling f is called super.

A *shackle* of graph H, symbolized by $G = shack(H, v, n)$, is a graph G developed by non-trivial graphs $H_1, H_2, \ldots, H_n$, such that for every $1 \leq s, t \leq n$, with $|s - t| \geq 2$, H_s and H_t have no common vertex, but for every $1 \leq i \leq n-1$, H_i and H_{i+1} have precisely one common vertex v, called *connecting vertex*, and all $n - 1$ connecting vertices are different. A *generalized shackle* of graph, denoted by $G = gshack(H, K \subset H, n)$, is the graph obtained from $G = shack(H, v, n)$ by substituting the connecting vertex by any subgraph $K \subset H$. If H is a non-trivial connected graph and $k \geq 2$ is an integer, then $shack(H, v, k)$ containing precisely k subgraphs that isomorphic to H, has been proven as H-super antimagic [5]. This proof, however, only covered a connected version of shackle of graph and did not cover all feasible d. This paper attempt to prove the existence of a super (a, d)-H antimagic total labeling for connected or disconnected generalized shackle of graphs.

The Existence of Super (a, d)-H antimagic total labeling (SHATL)

An *integer set partition technique* presented in [1, 3] is utilized to prove the existence of super (a, d)-H antimagic total labeling of $G = \text{gshack}(\mathcal{F}_{2,m}, e, n)$ and $G = s\text{gshack}(\mathcal{F}_{2,m}, e, n)$. This technique is employed for finding out the feasible difference d. Suppose n, m, d and k be positive integers. Consider the partition $\mathcal{P}^n_{m,d}(i, j)$ of the set $\{1, 2, \ldots, mn\}$ into n columns (where $n \geq 2$) and m rows such that the difference between the sum of the numbers in the $(j+1)$th m-rows and the sum of the numbers in the jth m-rows is always equal to the constant d, where $j = 1, 2, \ldots, n-1$. Therefore, these sums develop an arithmetic sequence with the difference d. The jth m-rows in the partition with the difference d, for $j = 1, 2, \ldots, n$ is denoted by $\mathcal{P}^n_{m,d}(i, j)$. If $\sum \mathcal{P}^n_{m,d}(i, j)$ be the sum of the numbers in $\mathcal{P}^n_{m,d}(i, j)$, then $d = \sum \mathcal{P}^n_{m,d}(j + 1) - \sum \mathcal{P}^n_{m,d}(j)$.

The following lemma is helpful for analyzing the existence of super (a, d)-H antimagic total labeling of $G = \text{gshack}(\mathcal{F}_{2,m}, e, n)$. The lemma is connected to the construction of the partition $\mathcal{P}^n_{m,d}(i, j)$.

Lemma 2.1. *[4] Let n and m be positive integers. For $1 \leq j \leq n$, the sum of*

$$\mathcal{P}^n_{m,d}(i,j) = \begin{cases} ni - j & ; \mathrm{i} \equiv 1(\mathrm{mod}3), 1 \leq i \leq m, 1 \leq j \leq n-1 \\ ni & ; \mathrm{i} \equiv 1(\mathrm{mod}3), 1 \leq \mathrm{i} \leq \mathrm{m}, \mathrm{j} = \mathrm{n} \\ ni - n + j + 1 & ; \mathrm{i} \equiv 2(\mathrm{mod}3), 1 \leq \mathrm{i} \leq \mathrm{m}, 1 \leq \mathrm{j} \leq \mathrm{n} - 1 \\ 1 + ni - n & ; \mathrm{i} \equiv 2(\mathrm{mod}3), 1 \leq \mathrm{i} \leq \mathrm{m}, \mathrm{j} = \mathrm{n} \\ ni - n + j & ; \mathrm{i} \equiv 3(\mathrm{mod}3), 1 \leq \mathrm{i} \leq \mathrm{m}, 1 \leq \mathrm{j} \leq \mathrm{n} \end{cases}$$

and

$$\mathcal{P}^n_{m,d}(i,j) = \begin{cases} ni - 2j + 2 & ; \mathrm{i} \equiv 1(\mathrm{mod}3), 1 \leq i \leq m, 1 \leq j \leq n-1 \\ ni - 1 & ; \mathrm{i} \equiv 1(\mathrm{mod}3), 1 \leq \mathrm{i} \leq \mathrm{m}, \mathrm{j} = \mathrm{n} \\ ni - n + 2j - 1 & ; \mathrm{i} \equiv 2(\mathrm{mod}3), 1 \leq \mathrm{i} \leq \mathrm{m}, 1 \leq \mathrm{j} \leq \mathrm{n} - 1 \\ ni - n + 2 & ; \mathrm{i} \equiv 2(\mathrm{mod}3), 1 \leq \mathrm{i} \leq \mathrm{m}, \mathrm{j} = \mathrm{n} \\ ni - j + 1 & ; \mathrm{i} \equiv 3(\mathrm{mod}3), 1 \leq \mathrm{i} \leq \mathrm{m}, 1 \leq \mathrm{j} \leq \mathrm{n} \end{cases}$$

form an arithmetic sequence of differences of $d \in \{\frac{m}{3}, \frac{-m}{3}\}$.

The main statement associated with the existence of super $(a,d) - H$ antimagicness of the connected graph $G = \mathrm{gamal}(H, K \subseteq H, n)$, is presented in theorem 2.1.

Theorem 2.1. *[4] For $m, n \geq 3$, the graph $G = \mathrm{gshack}(H, P_2, n)$ admits a super $(a,d) - H$ antimagic total labeling with feasible d is $d = \frac{m_1}{3} - \frac{m_2}{3} + m_3^2 - m_4^2 + m_5 - m_6 + \frac{r_1}{3} - \frac{r_2}{3} + r_3^2 - r_4^2 + r_5 - r_6 + 10$.*

3. Establishing a Stream Cipher

Stream cipher is an improved variant of the block cipher. At the block cipher, the plaintext is normally divided into several blocks with the same length, and is then encrypted block by block using the same key. At the stream cipher, however, different blocks may be encrypted using different keys. The system needs to generate only the initial key that is used to encrypt the first block. The keys utilized to encrypt the subsequent blocks are constructed from the previous keys based on the defined function.

Our developed cryptosystem is 5-tuples: $[G = \mathrm{gshack}(H, P_2, n), i, b, g(k), CBC]$. This system can be described as follows.

- The source of key stream is taken from SHATL of graph $G = \mathrm{gshack}(H, P_2, n)$
- The initial block key has a length of b and starts with the ith element of the sequence produced by the labeling.
- The key stream is generated by function $g(k)$.
- Stream cipher is implemented in the mode of Cipher Block Chaining (CBC).

3.1. Generating Key Stream

To generate a key stream, we modify the SHATL algorithm for constructing encryption keys that was proposed in [9]. This algorithm yields a sequence of labels taken from super $(a, d) - H$ antimagic total labeling of the graph $G = \text{gshack}(F_{2,m}, e, n)$. The sequence is then utilized as an encryption key. Drawbacks of this mechanism are the possibility of the key repetition and the need of large storage capacity. To overcome the drawbacks, the modified algorithm is focused to construct a key stream. Let we are working on 26 English alphabets. The key stream construction is undertaken through the following algorithm.

Algorithm 1. **Algorithm for generating key stream**

```
1. Define f for labeling the graph elements
2. If f is bijection, do 3, otherwise back to 1
    3. Take a certain d for super (a,d)-HATL
    4. Take z = the number of vertices plus 26
    5. Draw the layered diagram by ignoring all labels greater
       than z
    6. Place all edge labels in sequence from left to right
       and start from the top to the bottom layer.
7. Name the sequence by s and let t = length of s
8. Use the sequence s of labels as the source of key stream
9. Determine b = length of block
10. Determine i, such that $1 \leq i \leq t - b$
11. Take $k = s_i, s_{i+1}, s_{i+2}, ..., s_{i+b-1}$ as initial block key
12. Determine stream function $k_{j+b} = g(k_j, k_{j+1}, ..., k_{j+b-1})$
```

Outputs of algorithm 1 are the initial block key k and the stream function $g(k)$. For the sake of clarity, we need to have an illustration from the example provided in [9]. The labeling function is defined as follows: $f(x_{i,j}), 1 \leq i \leq 2, 1 \leq j \leq 5$, $f(y_{i,j}), 1 \leq i \leq 5, 1 \leq j \leq 4$, $f(x_{1,j}x_{2,j}), 1 \leq j \leq 5$, $f(e_{i,j}), 1 \leq i \leq 11, 1 \leq j \leq 5$. This labeling is illustrated in Figure 1. It shows that the vertex and edge labels start from 1 to 30 and 31 to 79, respectively.

A layered diagram rooted at label 1 is then drawn by ignoring the labels greater than 56. The diagram is illustrated in Figure 2. The source of key stream obtained from the diagram is 31, 39, 48, 52, 54, 40, 50, 49, 51, 53, 55, 47, 32, 36, 56, 43, 46, 33, 37, 42, 45, 34, 38, 41, 44, 35, or (in its equivalence modulo 26) 5, 13, 22, 0, 2, 14, 24, 23, 25, 1, 3, 21, 6, 10, 4, 17, 20, 7, 11, 16, 19, 8, 12, 15, 18, 9.

Suppose $i = 1$, $b = 5$, and the stream function is defined as $k_{j+5} = k_j + k_{j+1} \bmod 26$. We have the initial block key $k = 5, 13, 22, 0, 2$ and, thus the key stream is — 5, 13, 22, 0, 2 — 18, 9,

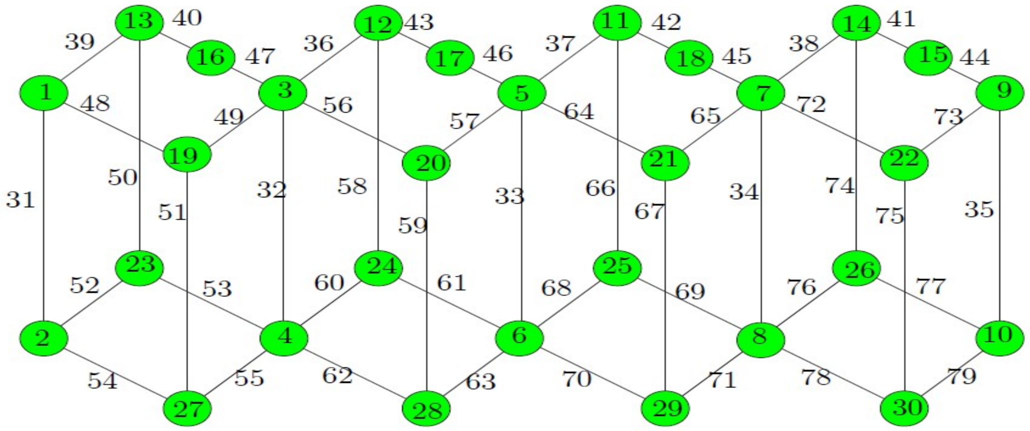

Figure 1. Super $(a, 12) - H$ ATL of the graph $G = \text{gshack}(F_{2,4}, e, 4)$.

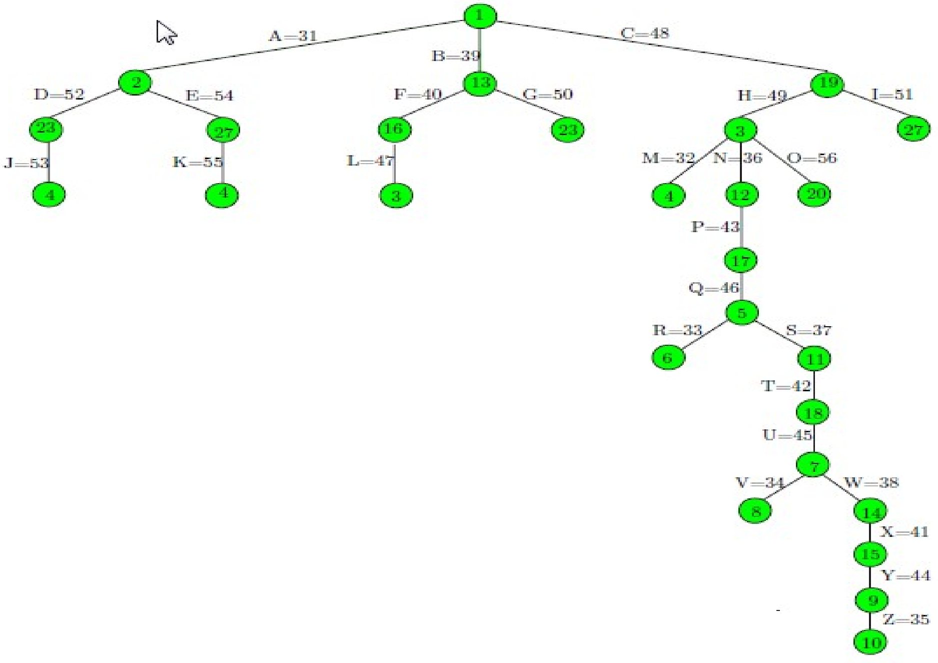

Figure 2. The layered diagram rooted at label 1

22, 2, 20 — 1, 5, 24, 22, 21 — 6, 3, 20, 17, 1 — 9, 23, 11, 18, 10 — 6, 8, 3, 2, 16 —...

3.2. Encryption Algorithm

The key stream produced by the algorithm 1 is implemented to establish a stream cipher in the mode of Cipher Block Chaining (CBC). Encryption process is undertaken using algorithm 2.

Algorithm 2. **Stream Cipher in CBC Mode**

1. Let the plaintext $P = (p_i), 1 \leq i \leq h$
2. Divide P into blocks of the length b.
3. For $i = 1$ to $\lceil \frac{h}{b} \rceil$, compute the ciphertext blocks using equation 1.

$$C_n = C_{n-1} + P_n + K_n \bmod 26 \tag{1}$$

where P_n, K_n, and C_n are the n-th block of plaintext, key sequence, and ciphertext, respectively. For $n = 1$, C_{n-1} is a null vector.

Table 1 exhibits how the key stream obtained from the algorithm 1 is utilized to encrypt the plaintext "indonesianjournalcombinatorics" and yields the ciphertext "NAZOPJBDQWTUVDEZIRIRJNPAUDMAEB". The decryption process can be done in the reverse direction.

Table 1. Example of Encryption Process.

plaintext	i	n	d	o	n	e	s	i	a	n	j	o	u	r	n
P_i	8	13	3	14	13	4	18	8	0	13	9	14	20	17	13
C_{i-1}	0	0	0	0	0	13	0	25	14	15	9	1	3	16	22
P'_i	8	13	3	14	13	17	18	7	14	2	18	15	23	7	9
K_i	5	13	22	0	2	18	9	22	2	20	1	5	24	22	21
C_i	13	0	25	14	15	9	1	3	16	22	19	20	21	3	4
ciphertext	N	A	Z	O	P	J	B	D	Q	W	T	U	V	D	E
plaintext	a	l	c	o	m	b	i	n	a	t	o	r	i	c	s
P_i	0	11	2	14	12	1	8	13	0	19	14	17	8	2	18
C_{i-1}	19	20	21	3	4	25	8	17	8	17	9	13	15	0	20
P'_i	19	5	23	17	16	0	16	4	8	10	23	4	23	2	12
K_i	6	3	20	17	1	9	23	11	18	10	6	8	3	2	16
C_i	25	8	17	8	17	9	13	15	0	20	3	12	0	4	2
ciphertext	Z	I	R	I	R	J	N	P	A	U	D	M	A	E	B

4. Security Analysis

Our cryptosystem combines SHATL and CBC. SHATL is utilized to produce the source of key stream, while CBC is the mode to encrypt the plaintext. The objectives of this combination are to achieve two principles - confusion and diffusion principles - that are mostly referred to strengthen block cipher. The combination makes the relation between plaintext, ciphertext and the key is hidden (*confusion principle*), and spreads the effect of a digit plaintext or key to as many as possible the ciphertext (*diffusion principle*). To analyze the security of the developed cryptosystem, we simulate four main possible attack models.

4.1. Ciphertext Only

In this attack model, an attacker only knows the ciphertext. The attacker may utilize a brute-force scenario, that is applying all possible keys to decrypt the ciphertext to find a meaningful plaintext. However, the plaintext of length h is divided into blocks of the length b and different blocks are encrypted by different keys, there are 26^b possible keys for each block, or totally there exists $(26^b)^{\lceil \frac{h}{b} \rceil}$ possible keys. Furthermore, in CBC mode, the keys for the 2nd to $\lceil \frac{h}{b} \rceil$-*th* blocks are confused by previous cipher blocks.

4.2. Known Plaintext

This attack model assumes that an attacker has information a part of the ciphertext and its corresponding plaintext. In our crptosystem, however, knowing only several pairs of ciphertext - plaintext is not adequate to reveal the whole blocks. This because the cryptosystem is a polyalphabetic cipher and the different blocks are encrypted using different sequence of keys.

4.3. Chosen Plaintext/Ciphertext

These attack models assumes that an atacker having a temporary access to the encryption/decryption machine. The attacker attempts to encrypt or decrypt a number of dummy plaintext/ciphertext and observes the results to derive the encryption/decryption keys. In our cryptosystem, a new key stream can be generated using SHATL everytime an encryption process is started. Therefore, a temporary access to the encryption/decryption machine is not sufficient to break the system, since at the subsequent times, the machine uses new SHATL-generated keys.

Conclusion

The open problem mentioned in [9] has been solved in this paper. A stream cipher can be established using a super (a, d)-H antimagic total labeling. The existence of super antimagicness of generalized shackle of graph $G = \text{gshack}(H, P_2, n)$ has been proven to guarantee the feasibility of the developed cryptosystem. The generated key stream guarantees that the different blocks were encrypted by the different keys. However. all block keys were connected one another, since a block key was generated from previous block key. Additionally, CBC mode makes encryption of a block was connected with encryption in the previous block. This research comes up with a more secure cryptosystem compared to previous one presented in [9] and ordinary block cipher. Furthermore, since only the initial block key that need to be save, the cryptosystem developed in this research requires less storage capacity compared to previous one.

This research, however, has not applied a randomness test to the produced key stream. This kind of test is needed to ensure that there is no specific pattern of the key stream that can be utilized by an attacker to guess the key.

Acknowledgement

Thanks to all colleagues at the Combinatorics, Graph Theory, and Network Topology (CGANT) research group, the University of Jember, Indonesia, for establishing this collaboration research.

References

[1] M. Bača, L. Brankovic, M. Lascsáková, O., Phanalasy, A. Semaničová-Feňovčíková, On d-antimagic labelings of plane graphs, *Electr. J. Graph Theory Appl.*, **1**(1) (2013), 28-39.

[2] Dafik, A. K. Purnapraja, R Hidayat, Cycle-Super Antimagicness of Connected and Disconnected Tensor Product of Graphs, *Procedia Computer Science*, **74** (2015), 93-99.

[3] Dafik, Slamin, D. Tanna, A. Semaničová-Feňovčíková, M. Bača, Constructions of H-antimagic graphs using smaller edge-antimagic graphs, *Ars Combinatoria*, **100** (2017), In Press.

[4] Dafik, M. Hasan, Y. N. Azizah, I. H. Agustin, A Generalized Shackle of Any Graph H Admits a Super H-Antimagic Total Labeling, *Mathematics in Computer Science Journal*, (2016). Submitted

[5] N. Inayah, R. Simanjuntak, A. N. M. Salman, Super $(a, d) - H$-antimagic total labelings for shackles of a connected graph H, *The Australasian Journal of Combinatorics*, **57** (2013), 127-138.

[6] A. C. Prihandoko, H. Ghodosi, and B. Litow, Obfuscation and WBC: Endeavour for Securing Encryption in the DRM Context, *Proceedings of the International Conference on Computer Science and Information Technology*, **CSIT** (2013), 150–155, ISBN 978-979-3812-20-5 2.

[7] A. C. Prihandoko, H. Ghodosi, and B. Litow, Secure and Private Content Distribution in the DRM Environment, *Proceedings of the Information System International Conference*, **ISICO** (2013), 659–664, ISBN 978-979-18985-7-7. Available online at Open Access Journal of Information Systems.

[8] A. C. Prihandoko, H. Ghodosi, and B. Litow, Deterring Traitor Using Double Encryption Scheme, *Proceedings of the IEEE International Conference on Communication, Network and Satellite*, **COMNETSAT** (2013), 100–104, ISBN 978-1-4673-6054-8. Available online at IEEE Xplore Digital Library.

[9] A. C. Prihandoko, Dafik, I. H. Agustin, D. Susanto, A. I. Kristiana, Slamin, The Construction of Encryption Key by Using a Super H-antimagic Total Graph, *Program and Abstract the Asian Mathematical Conference*, **AMC** (2016), 408, ISBN 978-602-74668-0-7.