Original Scientific Paper

Journal of Applied Engineering Science

doi:10.5937/jaes16-16557

Paper number: 16(2018)3, 549, 424 - 429

APPLIED POHLIG-HELLMAN ALGORITHM IN THREE-PASS PROTOCOL COMMUNICATION

*Robbi Rahim**

University Malaysia, School of Computer and Communication Engineering, Perlis, Malaysia

Three-pass Protocol is a method or technique that can be used by 2 (two) sender and recipient of the message to communicate with each other using XOR function, the problem that occurs is when the communication process there is parties who can know the messages sent from the sender to the recipient. To solve the problem we need an algorithm in this case Pohlig-Hellman algorithm, the use of Pohlig-Hellman algorithm on Three-pass Protocol ensures the security of messages sent by the sender to the receiver because each sender and receiver uses keys p, e, and d which is a number random and d is the inverse modulo of p and e of the sender and receiver, the results of this research suggest that it is impossible for the attacker or cryptanalyst to know the correct message quickly despite having adequate computer resources.

Key words: Algorithms, Recivers, Three-Pass Protocol, Pohlig-Hellman Algorithm, Sieve of Eratosthenes, Little Theorem Fermat

INTRODUCTION

The key protocol is a widely used exchange model where keys are distributed directly to recipients [1], the key distributions performed are usually performed on algorithms that have the symmetrical types [1], [2]. Cryptographic algorithm consists 2 (two) types that are symmetric and asymmetric [3]–[6], an symmetric algorithm use same key and on the asymmetric keys are used differently for each encryption and decryption[3]. The key in asymmetry type are not distributed, it's different with symmetry type but in symmetrical computing calculations are much faster than asymmetric[1], [2], [5], [6].

Asymmetric type cryptography has a distinct disadvantage to the symmetrical type where main problem lies in key distribution, in asymmetric the problem is the key that must be long to improve security and require complex and long[7]–[10].Implementation of cryptographyprotocols without key exchanges is still an area of less concern, the strength of the keyless cryptography protocol is based on padding and the exchange of keys generated especially in using prime number[11], [12], one of the algorithms using cryptography without key exchange is the three-pass protocol[2], [13].

Three-pass protocol is a framework that allows senders to send encrypted messages to recipients without having to distribute keys to message recipients[14], called a three-pass protocol because senders and recipients do not need to exchange keys and communication is perform in three directions where both parties each use the key[14]. The three-pass protocol scheme enables various types of cryptographic algorithms to be implemented, Pohlig-Hellman[13] is one of the cryptographic algorithms that can be used in the three-pass protocol scheme. Pohlig-Hellman are not type asymmetric and asymmetric algorithm because both encryption and decryption keys must be kept secret[13]. Pohlig-Hellman is different from Diffie-Hellman, Diffie-Hellman is a key exchange protocol that allows computers to generate similar secret keys for both systems and using public key as process in encryption and decryption[1], [15], [16]. The use of a three-pass protocol scheme with Pohlig-hellman is to cover the weaknesses of an algorithms that still using symmetric and asymmetric keys in the text message security process or by using session key.

METHODOLOGY

Security in a cryptographic protocol is very important [12], this is because many attacks on the protocol due to the selection of wrong algorithms in cryptographic protocol. Modular arithmetic [17], greatest common divisor [13], [17], euclidean algorithm, prime number [11] and inverse modulo[18] are used on the application of Pohlig-hellman algorithm and three-pass protocol.

Modular Arithmetic

Modular arithmetic is used in the process of encryption and decryption of the Pohlig-Hellman algorithm [13]. Encryption can be done to calculate the value of the message raised with the value of encryption key obtained then by doing modulo at predetermined prime values, the formula as follows:

$$m = n * q + r, 0 \leq r < n \tag{2.1}$$

Greatest Common Divisor

Greatest Common Divisor is used in the Pohlig-Hellman algorithm at the time of the determination of additional keys [13], [17]. The conditional additional key must be a member of the odd number in which the GCD between the odd number and the totient value obtained must be

* *Volga State University of Technology, 424000, the Republic of Mari El, Yoshkar-Ola, pl. Lenina 3, SmirnovMY@volgatech.net*

1. In the notation can be written

K_e∈odd, K_e GCD (K_e, θ) = 1 (2.2)

Greatest Common Divisor or GCD of the number of a and b is the largest integer d such that d | a and d | b. In this case we state that GCD (a, b) = d. Suppose that in determining GCD (5.2) = 1. It is found that the value of a is 5 and the value of b is 2.

Euclidean Algorithm

This algorithm is used in the Pohlig-Hellman algorithm for the determination of the value of additional key numbers[19]. Suppose that there are two non-negative integers m and n where m ≥ n, the Eulidean algorithm can find the largest common divisor of m and n.

Modulo Invers

If a and m are relatively prime and m> 1, then we can find the inversion of a modulo m. The inversion of a (mod m) [13], [20], [21], also called inversion multiplication, is an integer a-1.

a* (a - 1) ≡1 (*modm*) (2.3)

The relative prime definition it is known that GCD (a, m) = 1, and according to the equation there are integers p and q, such that: p*a + q*m = 1 implying that: p * a + q * m ≡1 (modm) Since qm ≡ 0 (*modm*) then the value of p * a ≡ 1 (*modm*), this variance means that p is the inverse of a (*modm*), the above process is used to find the inverse value which will be used Pohlig-Hellman encryption and decryption process.

Pohlig-Hellman

The concept of encryption on the Pohlig-Hellman Algorithm is similar to the RSA algorithm. Basically this algorithm is one asymmetric algorithm because it uses different keys for encryption and decryption [22]. In the Pohlig-Hellman algorithm does not use the public key concept because the key can be used at the time of encryption and decryption so it must be kept confidential[13], like in the RSA algorithm to be able to perform encryption and decryption were calculated by using formula as below:

C = Pe mod n

P = Cd mod n

Provide that the value of e * d ≡ 1

Based on function that used in this implementation of Pohlig-Hellman ini Three-Pass Protocol, Figure 1 display how function key and encryption process.

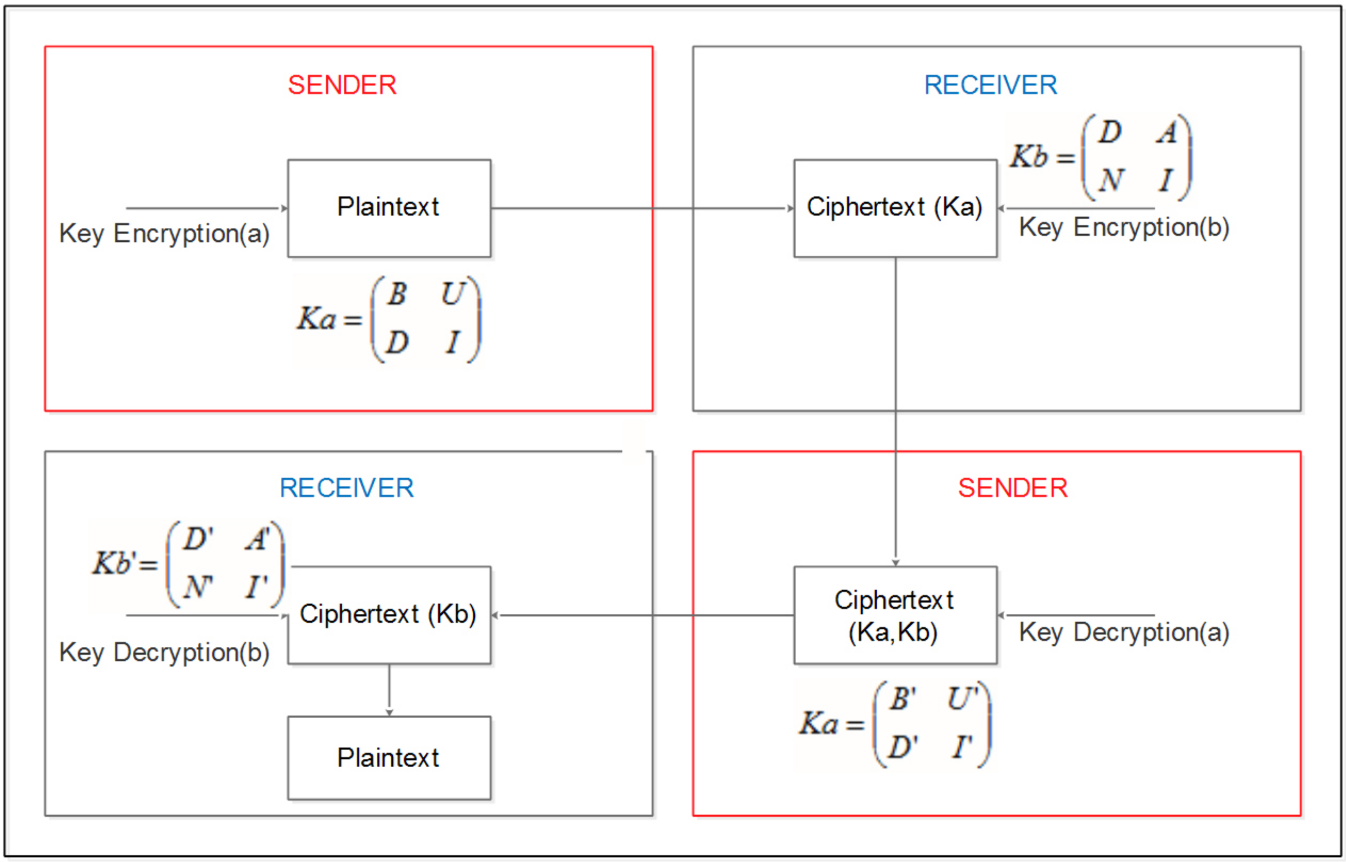

Figure 1: Pohlig-Hellman Process in Three-Pass Protocol

Three-Pass Protocol

Three Pass Protocol is a similar process of sending and receiving messages without key distributions and exchange keys so that the owner and recipient of the message does not worry that the message will change or read by a third party [17]. The Three Pass Protocol guarantees the absence of a key exchange between the party performing the encryption and decryption, each party having a private encryption key and a private decryption key [2], [13], [14], [18].

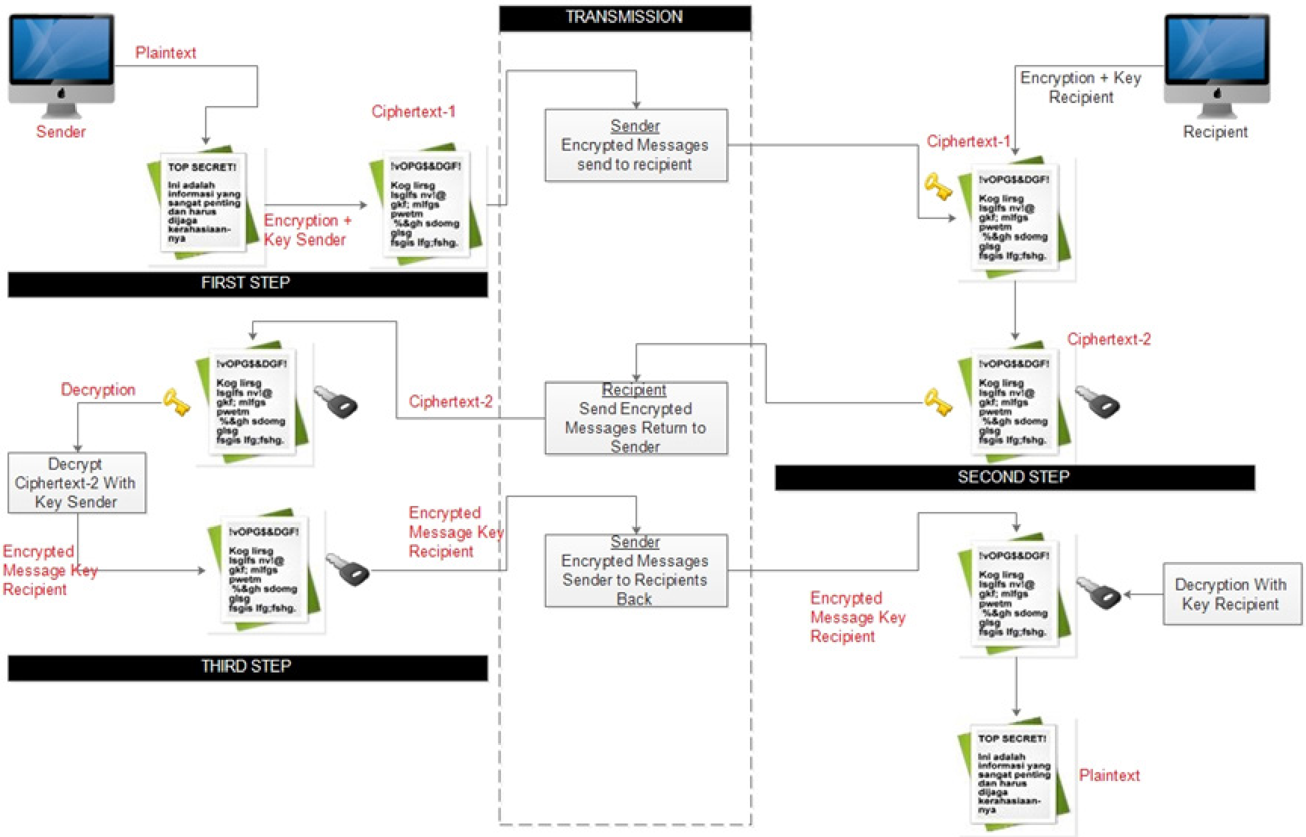

Figure 2: Three-Pass Protocol Scheme Process

RESULTS AND DISCUSSION

An experiment Pohlig-hellman algorithm with Three-pass protocol on communication is done gradually, first determining the key p, e and d on the sender and receiver of the message.

Table 1: Key of p, e and d Sender and Receiver

Sender	Receiver
p = 761351	p= 761351
e= 116733	e= 410551
d=229097	d=336601

Getting e value can be done by using Euler's totient function, here is the process:

p = 761351

totient p = 761350

Random e (1 < e < totient p -1)

e = 719498, GCD(761350,719498) = 2

e = 598658, GCD(761350,598658) = 2

e = 482416, GCD(761350,482416) = 2

e = 11832, GCD(761350,11832) = 2

e = 51575, GCD(761350,51575) = 25

e = 116733, GCD(761350,116733) = 1

Based on the above function can be value e = 116733 due to the result e (1 <e <totient p -1) = 1, and this process applied to sender and receiver e value. Getting a d value is not much different from the e value function and also using the Extended Euclidean algorithm to get a relatively prime d value.

The next test is the communication process done with the Three-pass protocol with Pohlig-Hellman algorithm after the known value of p, e and d, the message to be secure is “Robbi”.

1. Sender encrypt message

First step sender encrypt message and get the result as Table 2.

Table 2: Sender encrypt message

Char	Decimal	Encrypt	Result
R	82	82 ^ 116733 mod 761351 = 38715	38715 to byte array = [59][151][0][0]
o	111	111 ^ 116733 mod 761351 = 161101	161101 to byte array = [77][117][2][0]
b	98	98 ^ 116733 mod 761351 = 458699	458699 to byte array = [203][255][6][0]
b	98	98 ^ 116733 mod 761351 = 458699	458699 to byte array = [203][255][6][0]
i	105	105 ^ 116733 mod 761351 = 731164	731164 to byte array = [28][40][11][0]
All byte array convert to base64 and sent to receiver [59][151][0][0][77][117][2][0][203][255][6][0][203][255][6][0][28][40][11][0] = O5cAAE11AgDL/wYAy/8GABwoCwA=			

2. Receiver encrypt message

Message = O5cAAE11AgDL/wYAy/8GABwoCwA=

The message will be decode to decimal and get the result as:

[59][151][0][0][77][117][2][0][203][255][6][0][203][255][6][0][28][40][11][0]

Table 3. Receiver decrypt message

Decimal	Encrypt	Result
[59][151][0][0] to int = 38715	38715 ^ 410551 mod 761351 = 395200	395200 to byte array = [192][7][6][0]
[77][117][2][0] to int = 161101	161101 ^ 410551 mod 761351 = 417066	417066 to byte array = [42][93][6][0]
[203][255][6][0] to int = 458699	458699 ^ 410551 mod 761351 = 301152	301152 to byte array = [96][152][4][0]
[203][255][6][0] to int = 458699	458699 ^ 410551 mod 761351 = 301152	301152 to byte array = [96][152][4][0]
[28][40][11][0] to int = 731164	731164 ^ 410551 mod 761351 = 268029	268029 to byte array = [253][22][4][0]
All byte array convert to base64 and sent to sender again [192][7][6][0][42][93][6][0][96][152][4][0][96][152][4][0][253][22][4][0] = wAcGACpdBgBgmAQAYJgEAP-0WBAA=		

3. Sender decrypt message

Message = wAcGACpdBgBgmAQAYJgEAP0WBAA=

The message will be decode to decimal and get the result as:

[192][7][6][0][42][93][6][0][96][152][4][0][96][152][4][0][253][22][4][0]

4. Receiver decrypt message

Message = yqUDAAfjBgBHrgEAR64BAJ9xAQA=

The message will be decode to decimal and get the result as:

[202][165][3][0][7][227][6][0][71][174][1][0][71][174][1][0][159][113][1][0]

Based on the above test results, the encryption and decryption results of the three-pass protocol are done twice for both the sender and the recipient of the message, the power of using the three-pass protocol is in the Pohlig-hellman algorithm which involves the inverse modulo process and the extended Euclidean algorithm and also using a key generator algorithm of up to 1,000,000 bits, in theory this would make it hard to read the messages sent and takes n billion years to decipher.

Table 4: Sender decrypt message

Decimal	Decrypt	Result
[192][7][6][0] to int = 395200	395200 ^ 229097 mod 761351 = 239050	239050 to byte array = [202][165][3][0]
[42][93][6][0] to int = 417066	417066 ^ 229097 mod 761351 = 451335	451335 to byte array = [7][227][6][0]
[96][152][4][0] to int = 301152	301152 ^ 229097 mod 761351 = 110151	110151 to byte array = [71][174][1][0]
[96][152][4][0] to int = 301152	301152 ^ 229097 mod 761351 = 110151	110151 to byte array = [71][174][1][0]
[253][22][4][0] to int = 268029	268029 ^ 229097 mod 761351 = 94623	94623 to byte array = [159][113][1][0]
All byte array convert to base64 and sent to receiver again yqUDAAfjBgBHrgEAR64BAJ9xAQA=		

Table 5: Receiver decrypt message

Decimal	Decrypt	Result
[202][165][3][0] to int = 239050	239050 ^ 336601 mod 761351 = 82	82 = R
[7][227][6][0] to int = 451335	451335 ^ 336601 mod 761351 = 111	111 = o
[71][174][1][0] to int = 110151	110151 ^ 336601 mod 761351 = 98	98 = b
[71][174][1][0] to int = 110151	110151 ^ 336601 mod 761351 = 98	98 = b
[159][113][1][0] to int = 94623	94623 ^ 336601 mod 761351 = 105	105 = i

CONCLUSION

The implementation of Pohlig-hellman algorithm to the three-pass protocol can be done well, the use of algorithms such as euler totient, GCD, Rabin Miller, Extended Euclidean in the encryption process and decryption Pohlig-hellman algorithm can improve security especially from the key side used. The development of the Pohlig-helman algorithm is particularly possible at the time encryption and decryption combined with other algorithms or also the Three-pass protocol can be upgraded by combining with other protocols such as secret sharing or blind signatures.

REFERENCES

1. Abdalla, M., & Pointcheval, D. (2005). Simple Password-Based Encrypted Key Exchange Protocols. Lectures Notes in Computer Science, 3376, 191–208. https://doi.org/10.1007/978-3-540-30574-3_14
2. Al-Khalid, R. I., Al-Dallah, R. A., Al-Anani, A. M., Barham, R. M., & Hajir, S. I. (2017). A Secure Visual Cryptography Scheme Using Private Key with Invariant Share Sizes. Journal of Software Engineering and Applications, 10(01), 1–10. https://doi.org/10.4236/jsea.2017.101001
3. Alam, M. I., & Khan, M. R. (2013). Performance and Efficiency Analysis of Different Block Cipher Algorithms of Symmetric Key Cryptography. International Journal of Advanced Research in Computer Science and Software Engineering, 3(10), 2277–128.
4. Blumenthal, M. (2007). Encryption: Strengths and Weaknesses of Public-key Cryptography. CSRS 2007, 1–7. https://doi.org/PA 19085 CSC 3990 – Computing Research Topics
5. Bruce, S. (1996). Applied cryptography. John Wiley & Sons. https://doi.org/10.1017/CBO9781107415324.004
6. Guo, M., Bhattacharya, P., Yang, M., Qian, K., & Yang, L. (2013). Learning mobile security with android security labware. In Proceeding of the 44th ACM technical symposium on Computer science education - SIGCSE ’13 (p. 675). https://doi.org/10.1145/2445196.2445394
7. Hoffstein, J., Pipher, J., & Silverman, J. H. (2014). Diffie–Hellman key exchange. An Introduction to Mathematical Cryptography, 65–67. Retrieved from http://www.math.brown.edu/~jhs/MathCrypto/SampleSections.pdf

8. Li, N. (2010). Research on diffie-hellman key exchange protocol. In ICCET 2010 - 2010 International Conference on Computer Engineering and Technology, Proceedings (Vol. 4). https://doi.org/10.1109/ICCET.2010.5485276
9. Mesran, M., Abdullah, D., Hartama, D., Roslina, R., Asri, A., Rahim, R., & Ahmar, A. S. (2018). Combination Base64 and Hashing Variable Length for Securing Data. Journal of Physics: Conference Series, 1028, 012056. https://doi.org/10.1088/1742-6596/1028/1/012056
10. Nurdiyanto, H., & Rahim, R. (2017). Enhanced pixel value differencing steganography with government standard algorithm. In 2017 3rd International Conference on Science in Information Technology (ICSITech) (pp. 366–371). IEEE. https://doi.org/10.1109/ICSITech.2017.8257140
11. Nurdiyanto, H., Rahim, R., & Wulan, N. (2017). Symmetric Stream Cipher using Triple Transposition Key Method and Base64 Algorithm for Security Improvement. Journal of Physics: Conference Series, 930(1), 012005. https://doi.org/10.1088/1742-6596/930/1/012005
12. Putera, A., Siahaan, U., & Rahim, R. (2016). Dynamic Key Matrix of Hill Cipher Using Genetic Algorithm. International Journal of Security and Its Applications, 10(8), 173–180. https://doi.org/10.14257/ijsia.2016.10.8.15
13. Rahim, R. (2017). Man-in-the-middle-attack prevention using interlock protocol method. ARPN Journal of Engineering and Applied Sciences, 12(22), 6483–6487.
14. Rahim, R., Dahria, M., Syahril, M., & Anwar, B. (2017). Combination of the Blowfish and Lempel-Ziv-Welch algorithms for text compression. World Transactions on Engineering and Technology Education, 15(3), 292–297.
15. Rahim, R., & Ikhwan, A. (2016a). Cryptography Technique with Modular Multiplication Block Cipher and Playfair Cipher. International Journal of Scientific Research in Science and Technology (IJSRST), 2(6), 71–78.
16. Rahim, R., & Ikhwan, A. (2016b). Study of Three Pass Protocol on Data Security. International Journal of Science and Research, 5(11), 102–104. https://doi.org/10.21275/ART20162670
17. Rahim, R., Winata, H., Zulkarnain, I., & Jaya, H. (2017). Prime Number: an Experiment Rabin-Miller and Fast Exponentiation. Journal of Physics: Conference Series, 930(1), 012032. https://doi.org/10.1088/1742-6596/930/1/012032
18. Schneier, B. (1996). Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition. Network. https://doi.org/10.1016/S0740-624X(96)90083-0
19. Sklavos, N., Papadomanolakis, K., Kitsos, P., & Koufopavlou, O. (2002). Euclidean algorithm VLSI implementations. In Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems (Vol. 2, pp. 557–560). https://doi.org/10.1109/ICECS.2002.1046226
20. Sriadhi, S., Rahim, R., & Ahmar, A. S. (2018). RC4 Algorithm Visualization for Cryptography Education. Journal of Physics: Conference Series, 1028(1), 012057. https://doi.org/10.1088/1742-6596/1028/1/012057
21. Uchoa, A. G. D., Pellenz, M. E., Santin, A. O., & Maziero, C. A. (2007). A Three-Pass Protocol for Cryptography Based on Padding for Wireless Networks. In 2007 4th IEEE Consumer Communications and Networking Conference (pp. 287–291). IEEE. https://doi.org/10.1109/CCNC.2007.63
22. Yang, L., Wu, L.-A., & Liu, S. (2002). A quantum three-pass cryptography protocol. In Proceedings of SPIE - The International Society for Optical Engineering (Vol. 4917, pp. 106–111). https://doi.org/10.1117/12.483035

Paper submitted: 14.02.2018.
Paper accepted: 17.08.2018.