ICAI 2023
12th International Conference on Applied Informatics
Eger, Hungary, 2023

A new symmetric cryptographic system and key exchange protocol

Hussain Ahmad[a], **Carolin Hannusch**[b]

[a]Department of Computer Science, Faculty of Informatics, University of Debrecen,
Kassai út 26, 4028 Debrecen, Hungary
hussain.ahmad@inf.unideb.hu

[b]Department of Computer Science, Faculty of Informatics, University of Debrecen,
Kassai út 26, 4028 Debrecen, Hungary
hannusch.carolin@inf.unideb.hu

Abstract

We provide a new cryptographic system based on latin squares and error-correcting codes. The required key for decrypting can be computed by the encryption key, therefore the key has to be kept as a secret and a secure key exchange scheme is required. In the current paper, we introduce a scheme for the key exchange, where the key consists of permutations on a set with n elements. We determine the requirements for the cryptosystem and analyze its security.

Cryptographic system

Let L be a latin square of order n and denote its row permutations by $\sigma_1, \dots, \sigma_n$ and its column permutations by $\tau_1, \dots, \tau_n$ respectively. Further, let G be a generator matrix of a binary linear code C, whose decoding algorithm can correct t errors.

Key Choose a subset $I_1 \times I_2 \subseteq \{1, \dots, n\} \times \{1, \dots, n\}$ and compute

$$\rho = \prod_{i \in I_1} \sigma_i \prod_{j \in I_2} \tau_j.$$

Then the secret key of the cryptographic scheme is (G, ρ).

Encryption Let m be the message to be encrypted. Then the encrypted message is

$$Enc(m) = (mG)^{\rho},$$

which means we apply the permutation ρ to the binary vector mG.

Decryption From the secret key, we can compute ρ^{-1}, such that $\rho \cdot \rho^{-1} = \rho^{-1} \cdot \rho = 1_{id} \in S_n$. Thus if $y = Enc(m)$, then we first compute $z = y^{\rho^{-1}}$ and then apply the decoding algorithm of C to z.

It is clear that $z = y^{\rho^{-1}} = (mG)^{\rho\rho^{-1}} = mG$ and thus decoding z, we get back the message m.

Key exchange protocol

We give a protocol for the key exchange of a permutation using a latin square. A man in the middle can only access the whole key if both directions of the key communication are attacked.

Alice		*Bob*
1. Choose $I_1 \subseteq \{1, \dots, n\}$ 2. Choose $\mid I_1 \mid$ permutations of S_n such that they are the first $\mid I_1 \mid$ rows of a latin square		
3. Send $\sigma_1, \dots, \sigma_{\mid I_1 \mid}$ to Bob	$\xrightarrow{\sigma_1,\dots,\sigma_{\mid I_1 \mid}}$	
		4. Determine the first $\mid I_1 \mid$ positions of the column permuations $\tau_1, \dots, \tau_n$ 5. Extend each one to permutations of S_n such that they are columns of a latin square 6. Choose $I_2 \subseteq \{1, \dots, n\}$ 7. Compute $\rho^\star = \prod_{i \in I_2} \tau_i$
	$\xleftarrow{\rho^\star}$	8. Send this permutation to Alice
9. Compute the key $\rho = \prod_{i \in I_1} \sigma_i \cdot \rho^\star$		9. Compute the key $\rho = \prod_{i \in I_1} \sigma_i \cdot \rho^\star$

Security analysis

Let's assume that the attacker penetrated one direction of key communications and get the permutations $\sigma_1, \dots, \sigma_{|I_1|}$ or $\rho^\star$. It has to compute the other direction permutations to get ρ, i.e. it must be able to find the used latin square in order to determine the other permutations, or can guess a permutation on n elements. Since

$\rho \in S_n$, the number of permutations for n elements is $n!$, so an algorithm to produce all $n!$ permutations would have time complexity $O(n!)$. We direct the reader for some literature about these algorithms to [3], [2] and [1]. It's quite easy to see that the factorial is approximately exponential in behaviour. a factorial algorithm may be practical in a few special cases. i.e. where n is extremely small, but becomes impractical very quickly as n grows. The time complexity of multiplication of two permutations $\in O(n)$, which means the time required to multiply n permutations is $O(n^2)$, thus we have the time complexity is $O(n^2 * n!)$. So if we choose n large enough, the ability to compute ρ even the attacker got $\sigma_1, \dots, \sigma_{|I_1|}$ or $\rho^\star$ will be not possible.

References

[1] Y. Bassil: *A Comparative Study on the Performance of Permutation Algorithms*, arXiv preprint arXiv:1205.2889 (2012), pp. 7–19, arXiv: 1205.2888, URL: http://arxiv.org/abs/1205.2888.

[2] D. F. Holt, B. Eick, E. A. O'Brien: *Computation in Finite Permutation Groups*, in: Handbook of Computational Group Theory, 2005, pp. 77–148, ISBN: 9781420035216, DOI: 10.1201/9781420035216.

[3] Á. Seress: *Permutation Groups: A Complexity Overview*, in: Permutation Group Algorithms, Cambridge Tracts in Mathematics, Cambridge University Press, 2003, pp. 48–54, DOI: 10.1017/CBO9780511546549.003.