A Study of Pohlig - Hellman Cryptosystem Parameters

Dolantina Hyka[1], Katerina Zela[2], Sediola Ruko[3] ,Festim Kodra[4]

Abstract— — In this paper we are considering the parameters of the system that generates a strong or diagnostic cryptosystem, so it can be used for teaching purpose or to real life problems that need to be treated with symmetric encryption. First of all we detect the problems that can occur during the algorithm steps calculations. Secondly, the issue of selecting parameters is addressed in order for the system to be as secure as possible. in both of them are explained the reasons how the selection or non-selection of a certain element, affects the proper functioning of the system, both from a didactic point of view but also from that of increasing security. In this way it is expected to avoid all possible errors. So it can be a challenge to avoid possible attacks and to obtain any enhancement of clasic systems

Index Terms— Cryptography, diagnostic systems, education, modulus, simetric systems, strong systems, parameters, Pohlig – Hellman.

1 INTRODUCTION

In this paper is presented the Pohlig - Hellman cryptosystem with all its characteristics.

First, the cryptosystem is mentioned, as well as the way of encryption - decryption through it. Then a series of errors are given one after the other which can occur during the calculations or during the selection of the parameters in an incompletely appropriate way.

This is followed by a brief overview of the steps to follow to avoid these errors and a pseudocode that generates examples and a strong diagnostic system.

The cases reviewed are presented for the generation of strong or diagnostic systems.

If the system is given and we have to check the parameters or the result of an exercise or example it is enough to replace the initial generating conditions with the values given in the exercise and in the intermediate steps and apply the conditions for the given fixed values not autogenerated.

- *Dolantina HYKA Dr. in Algebra (Cryptography), University of Tirana, Albania. Head of Information Technology, Statistics and Institutional Coordination office at Mediterranean University of Albania. E-mail: dolantina.hyka@umsh.edu.al*
- *Katerina Zela MSc. in Mathematics, Politecnical University of Tirana, Albania. Lecturer at Mediterranean University of Albania E-mail: katerina.male@umsh.edu.al*
- *Sediola Ruko Msc.in Operation Mamangment in Informatic University of Tirana, Albania.Lecturer at Mediterranean University of Albania E-mail: sediola.ruko@umsh.edu.al*
- *Festim Kodra MSc. in Economics, Agricultural University of Tirana, Albania. Lecturer at Mediterranean University of Albania E-mail: festim.kodra @umsh.edu.al*

This way we will see if any mistake has been made even if the system is diagnostic we will see what mistake has been made and how to act to avoid it.

2 ENCRYPTION AND DECRYPTION ALGORITHM

2.1 Review Stage

To make a communication between two parties, the message center [1], [2]

a. Generates a relatively large number p which is a prime number.

b. Choose a private key such that (e, p-1) = 1.

c. The private key $d = e^{-1} \bmod p-1$ is calculated

If Alice needs to send Bob an encrypted message via the Pohlig-Hellman cryptosystem, follow these steps:

- Converts the message from alphanumeric to plain numeric text
- Encrypts the message using the encryption algorithm:

$$C = M^{e} \bmod p$$

and sends it to Bob.

He encrypts the message by applying over the encrypted text, his private key

$$d = e^{-1} \bmod p-1$$

thus reading the message

$$M = C^{d} \bmod p$$

3 Errors and mistakes to be avoided during

algorithm steps, for generating strong or diagnostic systems.

➢ The public key generation e must be such that (e, p-1) = 1. If this condition is not accomplished, it is impossible to generate the corresponding private key. This leads to the result that open text can be encrypted but will remain unreadable, as a key that enables its decryption cannot be calculated.[3]

➢ The generation of the public key must be such that (e, p-1) = 1. If this condition is not happened, it is impossible to generate the corresponding private key. This leads to the result that open text can be encrypted but will remain unreadable, as a key that enables its decryption cannot be calculated.

➢ If is chosen as the private key, a number such that (e, p) = 1 and e-1 mod p = e-1 mod p-1, then it would not be discernible whether the operations to calculate the private key were performed correctly or not according to the respective module. By setting the condition (e, p)≠1 from the beginning when the public key is generated, we avoid this error in the following steps, when building strong examples.

➢ If we want to construct diagnostic examples, this condition is set special when the private key d = e-1 mod p ≠ e-1 mod p-1 is calculated in order to see which of the errors was performed. Another way to express this condition would be d ≠ shvp (p, p-1) which is equivalent to the first condition.

➢ The operations for the calculation of encrypted text and open text, are performed according to module p and not according to module p-1, therefore in the algorithm must be set a condition which avoids the same result of operations with the respective texts according to two different modules. This condition is:

C mod p ≠ C mod p-1 M mod p ≠ M modp-1

➢ In some cases the initial encoding A-Z = 1-26 can be obtained and in some others A-Z = 0-25 or some other definition. One way to avoid this error is to determine the transition from alphanumeric to numeric text and vice versa from the beginning by means of a function which is called by a cycle or by a certain designation throughout the algorithm so that the definition can not allow new initial encryption.

➢ The encryption key must be chosen to be different from the decryption d key because otherwise the cryptosystem would be completely fragile, as the chances of breaking the system with a statistical attack would be twice as high. Also in the case when these parameters are used in an exercise, the student can get the idea that both parties use the same key for both encryption and decryption of the message, thus misconceiving the system. In this case, the student may not distinguish between the encryption and decryption key and make accurate calculations without finding the decryption key at all. To avoid this error, the condition d = e-1mod p-1 ≠ e mod p-1 is also added to the algorithm that will generate the data.

➢ Another aspect is the case when d = e mod p. This creates again the idea that the parameters e and d should be the same, giving the student the misconception that despite the module both the encryption key and the decryption key should be the same.

➢ On the one hand it transmits to the student an error in the notion indirectly, on the other hand it significantly reduces the security of the system by making it more vulnerable to possible statistical attacks as in the case above but not only. To avoid this, it suffices to add as a condition d ≠ e mod p.

➢ One of the most common mistakes encountered by students is the use of p-1 instead of p for the calculation of open text and encrypted father during exponential calculations, so in creating the exercise can be set as a condition C = M e mod p ≠ M e mod p-1 and M = C d mod p ≠ C d mod p-1.

➢ For diagnostic examples, in each of the two conditions should be added the corresponding comment "caution actions can be performed according to the wrong module"

➢ e-1 mod p-1 ≠ -e mod p-1. The opposite element of the public key can be taken as a private key instead of its inverse, a condition that affects the misconception of the symmetric Pohlig-Hellman cryptosystem.

➢ d ≠ e because otherwise the actions would be performed without calculating the inverse of e at all and the idea can be created that it is an unnecessary element in the crypto-system. Another reason is that the cryptosystem in this case would be too weak. Also if the encryption and decryption key were the same, this could lead to other errors like the ones below [4]

➢ It may happen that instead of calculating the coded text as C = M and mod p, open text as M = Cemod p. In this case the decryption key is used instead of the decryption key. One way to avoid this error is to set the above condition in the algorithm built to generate strong examples but this would not completely eliminate this error as there are cases where condition 10 is met and yet the error mentioned at this point may occur since the actions are performed in a finite ring.

- It may happen that instead of calculating the open text as $M = C^d \bmod p$., Calculate the Encrypted text as $C = M^d \bmod p$. So to confuse the sequence of actions as well as the concept on encryption and decryption. The reasoning presented above with the error mentioned before.

- $e \neq \varphi(n)$ If it happens that the encryption key is equal to $\varphi(n)$ itself, then from Ferma theorem, instead of the encrypted text C a message would be transmitted, the characters of which would be all 1 (1111 ... 11) and so not only do we incorrectly follow the steps of the cryptosystem but we also turn the message into a meaningless text which cannot be deciphered because there is no $\varphi(n)^{-1} \bmod \varphi(n)$. So in this case it is impossible to find the decryption key. This makes the message unreadable even by the legitimate recipient.

- One way to avoid this would be to determine that at the moment of generation of e a maximum limit for it (which clearly appears to be $\varphi(n)$). Another way (in the case of the diagnostic system is more appropriate than the first way) would be to set the condition of $\neq \varphi(n)$ in the algorithm together with the corresponding comment on what are the errors to which the choice of a key can converge such public.

- It can happen that during the calculations of the private key d, the modulus $\varphi(n)$ is confused with the public key and taking as the inverse of the latter in the bezu equation, not the coefficient near e but the coefficient near $\varphi(n)$. It is clear that they will not have the same value, but during the calculations for decrypting the message it may happen that the required result is achieved even by committing this error.

4 Steps for generating strong or diagnostic systems

The following are the steps for generating strong or diagnostic systems, exercises and various such examples. The relevant pseudocode is also presented which avoids these errors treated analytically above in the case of the Pohlig - Hellman cryptosystem over Zp. [4]

- First of all, a suitable module p is initially selected.

- Te message have to be converted from alphanumeric to plain numeric text

- Using an encryption function or a hash function the plain text is converted into a string according to the module p.

- A public key e is chosen such that meets the conditions discussed above is selected, thus forming the pair of public keys (p, e) ..

- After that the algorithm have to calculate the privat The private key d that meets the set conditions.

- The next step is to insert the criterias for the cryptosystem and keys are set as well as relevant comments for each case of errors in order to generate diagnostic systems or exercises.

- The keys are converted to binary system to then implement the quadratic multiplication algorithm related to modular power.

- Insert the conditions for plain text and encrypted text during the encryption and decryption process as mentioned above.

5 The pseudocode to generate Strong system parameters.

quadraticmultiplication algorithm

Imput : p, M ϵ Zp, $l = \sum_{i=0}^{t} l_i \cdot 2^i$ and l_t=1

Output : $T=M^l \bmod p$

Algorithm Pohlig - Hellman

Imput: p **prime number** < Rg, M ϵZp

Output: M ϵZp **encryption and decryption of** M **with** Pohlig – Hellman **cryptosystem**.

Algorithm:

Begin

Rg=26

Key_e = True

While Key_e **do**

{

e=**integer (**1,p-1)

If (**GCD**(e,p-1)=1) **then**

{

C= **quadraticmultiplication** (M,Convertbinar(e))

If (**GCD**(e,p)=1 or C=M mod n or C=M mod (p-1) or C=C mod (p-1) or e=1 or e=(p-1)) **then**

Key_e= False

}

}

Key _d= inverse(e) mod (p-1)

While Key _d **do**

{

If (d ≠ e) **then**

{

M= **quadraticmultiplication** (C,Convertbinar(d))

If (**GCD**(d,p)=1 or d mod p =-e mod n or d=p-1 or M =C^{e} mod p or M=C^{e} mod (p-1) or M=C^{d} mod (p-1) or d=1) **then**

Key_e= False

}

}

Return (C,M)

End.

6 Conclusion

The purpose of this paper is to identify the most common errors that occur during the execution of steps in the Pohlig - Hellman algorithm.

This first not only in terms of didactics, practical exercises in teaching cryptography but also concrete steps of the algorithm in real encryption and decryption of daily life.

This paper deals with the construction of robust diagnostic systems as well as improvements possible classical asymmetric cryptosystems. Mistakes students make during calculations of steps in each of them.

We first consider the possible errors during the calculations of steps of the cryptographic algorithm and then building the pseudocode generating algorithms strong or diagnostic systems and examples, avoiding all errors treated analytically.

So if a wrong path is followed to encrypt or decrypt them not to be achieved in a precise conclusion, which serves to easily control even exercises with alternatives based on different cryptosystems. Building strong cryptographic or diagnostic systems is of particular importance not only in the teaching of cryptography, but also in the conception and creation of cryptosystems

These systems try to avoid all possible mistakes. therefore can be the impetus to avoid even possible attacks and to improve known cryptosystems or the creation of new ones.

This conclusion can be extended even further to the construction of algorithms not only the discipline of cryptography but also of many other disciplines, such as the generation of differential equations of different types without having to be checked in advance, generating matrix equations with the certainty of the number of solutions they have been without e necessary to resolve in advance, generating different situations which require solutions optimal, etc ...

REFERENCES

[1] Rahim, Robbi. (2018). Applied Pohlig-Hellman algorithm in three-pass protocol communication. Journal of Applied Engineering Science. 16. 424-429. 10.5937/jaes16-16557.

[2] V. Sklyar, O. Illiashenko, V. Kharchenko, N. Zagorodna, R. Kozak, O. Vambol, S. Lysenko, D. Medzatyi, O. Pomorova. Secure and resilient computing for industry and human domains. Volume 1. Fundamentals of security and resilient computing / Edited by Kharchenko V. S. – Department of Education and Science of Ukraine, National Aerospace University named after N. E. Zhukovsky "KhAI", 2017.

[3] "Adequate Selection Methods Using Strong/Diagnostical Systems in Public Key Cryptography", First International Conference "Mathematics Days in Tirana"11-12 December 2015, Tirana,Albania

[4] "Digital Signature,Parameters to Increase Its Viability", Third International Science Conference "Scientific Chalenges For Sustainable Development". 09 Prill 2016, Struge, Maqedoni.