THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 13, Number 2/2012, pp. 175–179

SECURITY SYSTEM VULNERABILITIES[*]

Nicolae CONSTANTINESCU

University of Craiova, Department of Informatics
E-mail: nikyc@central.ucv.ro

The development of the notion of computer system led to a parallel development of the information models of defense stored in these. In parallel it was created an arsenal of access to information without having this right. The first stage is cryptography, the second cryptographic analysis. In these two worlds "parallel in terms of declarative, but with many intersections in practice", the mathematical models were those that were imposed as the development of viable solutions. In this paper are indicated some aspects of the two worlds from the author's practice and not only, and solutions to those stated. The red thread of this world is the same as in that of war: "a castle can have walls however high, it could be won, depending on how much you are willing to pay, so how much is worth what is inside".

Key words: cryptography, cryptanalysis, system vulnerabilities.

1. INTRODUCTION IN SECURE DATA SYSTEMS

To secure a data system. Is needed. Something that we tend to. This starts from analyzing a model of transfer, storage and modification of information. The principles underlying the model defined are those that compete from the idea of access to data. Let's start from the definition of the components of a data security system. This is composed of:

– algorithms,
– hardware,
– software,
– policies.

All these elements have a well defined role and the absence of each will create a gap that will result in the cancellation of the security model targeted. The first category includes the implementations of some mathematical models that turn data from something known into something unintelligible and that applies techniques that permit a user access to information. The second category consists of structures that physically implement the idea of the one who thought the information security structures. The software is the implementation of the algorithms described at the first point. Policies apply to all these, ie the system's security policies. These represent the rules of use for this physical and software overall.

In the following we will discuss for each of the existing problems and through specific examples we will illustrate how to create holes in a system. Each model chosen to satisfy the requirement of secrecy will follow a phased standard that includes the basic idea of the system "every entity that wants to have access to the system will show it is who it says it is, depending on this it will be given a set of rights and it will be pursued what it did". This can be called basic security policies. To describe an attack model we will illustrate each part of the information defense system and we will illustrate what can occur when they are flaws in these, the examples being from real models at the analysis of which we have been part. Lets first look from the point of view of an entity (person or any other system) that wants to have access to an information. Let's start thinking about those described above. The first step is to have permission to access it. It appears here the notion of proving who you are. The best known is through username and password. This model states

[*] This paper was presented in part at the 1st Conference *Romanian Cryptology Days* – RCD2011, October 11–12 2011, Bucharest, Romania.

that is the entity user, and proves it by password. This model is transhipped in various topics based on what we know, what characterizes us or what we have. Hence the notion of authentication. The second step is to have access to shares: to read, to copy, to modify, resulting from the policies granted to the entity authenticated. The third step is to transport the information from one side to another. For this we must make it, again, to have access to it only those who can demonstrate that they have this right. Hence authentication at both ends of transmission, encryption on the communication channel, management of an encryption key system based on policies of the selected model. And above all come the use policies of the entire secure system. The entire model is described in Fig. 1.

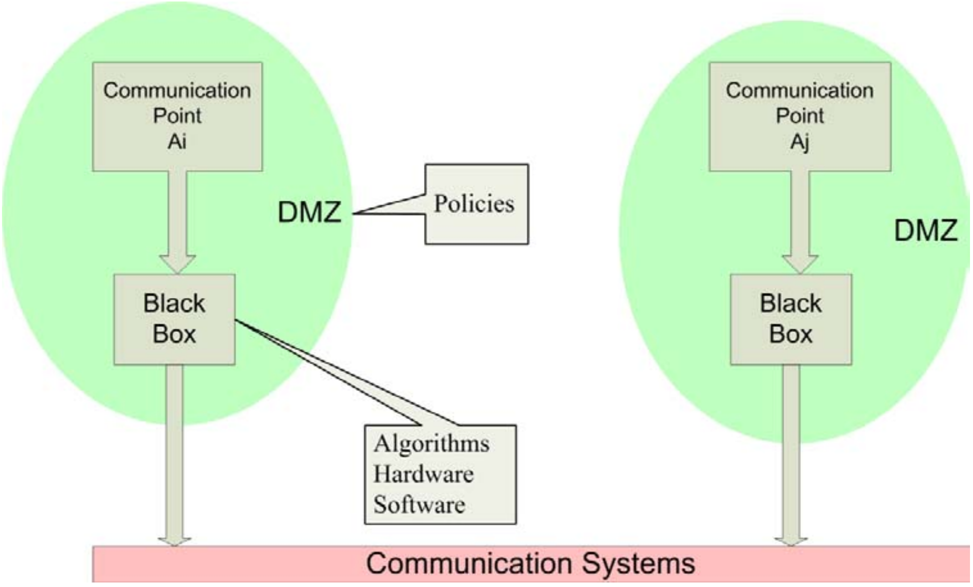

Fig. 1 – Legend – communication system structure.

The DMZ locations are the demilitarized zones where is thought to be safe to communicate between the involved parties.

2. ISSUES IN VULNERABILITY ANALYSIS

2.1. Algorithms Discussions

The algorithms are those behind the work little questionable, by implementers, of the chosen mathematical models. The main causes of the problems appeared in this area are based on the lack of an interface between mathematicians and informaticiens, the errors made knowingly or not by the implementers and the non compliance to the security policies of the implementation. For example we will take an example (reads): An entity, noted x, which had a transmission system according to the one in Fig. 1. For the implementation used a key generator model, adapted to the chosen security system. This generator used a library of mathematical functions that calculated the pieces required in the functions to generate the keys. These mathematical models implemented algorithms with large numbers that used two methods to generate these parameters: a primacy test method based on the classic model (Fermat primality test [1]) and Germain prime test method [2]. Both methods were based on sequences of numbers used in the construction of keys. The first implementation was done using the classic model that has been penetrated due to the differences between the properties of those two types of numbers generated. After the first audit were used algorithms of type two. When analyzing the security degree (this is the polite expression pattern that there were information leaks from the system that after the analysis were revealed to be due to the storage-transfer data system) was found that data was not confidentialised correctly. This was because the generation of keys was based on consecutive applications of numbers generated according to those previously described and how the applications were succeeded and the acquisition of the first set was done according to a linear graphic.

From here we deduce the lack of interface between the project mathematicians and implementers of the system software. In the real exhibited case, the losses were exponential due to the cost reductions caused by the close deadline that was decided. The examples may continue, this I think is eloquent in defining the role of interfacing between mathematicians and software designers.

2.2. Hardware Discussions

To implement a security system are needed modelings that take into account the needs of the entity in which will be used the informational structures. Some of these are the hardware systems. In the following we will describe a case of implementation that led to problems related to the communication system. To an entity on which was made the security degree analysis, was discovered a flaw in the communication system of the packets competing on the number generator used in authentications. After the comparative analysis, it was found that the routing system used was based on embedded software that used algorithms in security breaches. The conclusion was that the system was designed correctly in whole, but the model was taking into account as being DMZ a black box introducing vulnerabilities in the system. Here the examples that I was part of in the vulnerability degree analysis can be conclusive in the black box area sites and their risk. I will review: Routers from big name companies, Firewalls from the same entities, PCoIP implementation from more than 70% market share. The red thread of this paragraph is the need to build a system based on hardware only as a computer system, not as a black box with embedded algorithms.

2.3. Software Discussions

Considering the creation method of the software for the information security systems we can conclude that the problem can be divided into:

– our mistakes,

– our backdoors.

The first category is based on how to write software and its testing methods (involve the interfaces between the group of mathematicians and software designer and the cryptographic analysis group that validates the software). The second category of information is subject to some discussions not related to the purpose of this article.

As an example for the first category I will describe the result of an audit of an entity that aims to transfer (securely) the information pertaining to financial data. During the action have been defined the categories of existing vulnerabilities and one of them was the way of authenticating messages of report for the actions on the system. This module was based on a pattern of 112 bits of ECC. The implementation, for optimization, had a table of values used in intermediate calculations. This table was accessed through a method using large sized uncompressed data. The result was that the vulnerability given by the uncompressed data volume allowed the comparative analysis of this table with a dictionary of existing tables in the literature of specialty. From this was deduced the sequence of parameters used, from where the known results and then in terms of revealing that information. For details on the general models of software threats can be consulted [7].

2.4. Policies Discussions

The most delicate point of vulnerability analysis is to discuss the security policies. Here there is no mathematical model to follow, only simulations of all the possible actions, results and taking actions to limit the damages. A complete security policy does not exist, the one of complete classification is only the ideal, the more severe and tested to limit damage in case of a security breach. In the most common problems such as these stands out the "boss-underling" policy update and policy revision in a modifiable system. For the first I will describe the safety analysis of a system that has strict confidentiality policies, and inside that entity the tests made have described a level of vulnerability that, after the audit has received favorable rating. Following an act of replacing the entities involved, due to some details not covered by this article, there was a breach in security. The established policies were not taking into account such a breach because, according to those existent, this was not possible. From here the limitation of damages was not standardized resulting in a hierarchical level of authentication corresponding between the components and thus collapsing the whole security system. For the second category of policies, we can illustrate the methodologies underlying the upgrading of the banking systems in Eastern Europe to those of communication in Western Europe. In these studies, merging the two entities was necessary to restructure the entire existing security model (with costs that are higher than those of building from scratch). For an overview of the area can be consulted [4] and [6].

3. AUTHENTICATED KEY PAIR GENERATION

Based on experiences gained in previous descriptions, it was developed a method of access-communication-access to data that is based on a mathematical model studied and tested over the years in different entities. In this part we will describe how the protocol creates the keys involved. The entire system is based on the scheme in Figure 2 and the software was implemented in the embedded technology.

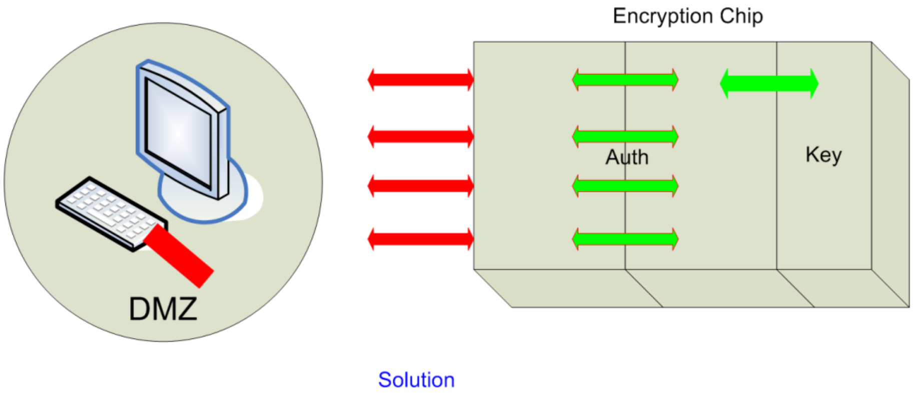

Fig. 2 – Legend – DMZ enrollment.

The choice of description of this algorithm was made because of the degree of its applicability in systems that requires multiple authentication and the reducing of the volume and complexity of the security policies necessary for choosing a model based on this structure. The general description of a variant of this algorithm is found in [3].

3.1. First T Step

1. generates a pseudorandom number $d_T \in [1, n-1]$
2. calculates $T_1 = d_T \left(P^{-1} + Q\right) = \left(x^T_{1, y_1^T}\right)$. Let $x = x_1 \bmod \ n$

   If $x = 0$ then goto step 1
3. calculates $T_2 = h\left(P_T \mid T_1\right)$
4. calculates $T_3 = e_T^S \left(S_{K_T}, T_2\right)$
5. The first communication step (from T to R)

   T sends to R$\left(T_1 \mid T_2\right)$

3.2. R Step

1. calculates $S_1^R = h\left(P_T \mid T_1\right)$

2. calculates $S_2^R = e_T^P\left(P_{K_T}, T_3\right)$. If $S_1^R \neq S_2^R$ terminates the protocol with failure

3. generates pseudo random number $d_R \in [1, n-1]$

4. calculates $R_1 = d_B\left(P^{-1} + Q\right) = \left(x_1^R, y_1^R\right)$. If $x_1^R = 0$ go to step 3 of R's steps

5. calculates $R_2 = h\left(P_R \mid R_1\right)$

6. calculates $R_3 = e_R^S\left(S_{K_R}, R_2\right)$

7. $K_R = d_R T_1 = \left(x_2^R, y_2^R\right)$

8. $x = x_2^R \bmod n$. If $x = 0$ then goto step 3 of R's steps

9. The second communication step from R to T. R sends to T $\left(R_1 \mid R_3\right)$

3.3. Second T step

1. Calculates

$$S_1^T = h\left(P_R, R_1\right)$$

$$S_2^T = e_R^P\left(P_{K_R}, R_3\right)$$

2. If $S_1^T \neq S_2^T$ terminates the protocol run with failure

3. $K_T = d_T R_1$

4. CONCLUSIONS

I conclude by saying that in this area the optimism is for those who worked only on building security systems, not in their cryptanalysis (see the white paper or the various large entities such as a study described in [5]). The others only work. There are good cryptographic systems but the security systems can only be optimal for different applications of an entity.

Defining the field is how to create and implement the security policies in the event of a breach in the overall security system.

REFERENCES

1. RICHARD CRANDALL and CARL POMERANCE, *Prime Numbers: A Computational Perspective*, Chapter 3: "Recognizing Primes and Composites", pp. 109–158, and Chapter 4: "Primality Proving", pp. 159–190, 2nd ed., Springer, 2005.
2. HARVEY DUBNER, *Large Sophie Germain Primes*, Mathematics of Computation, American Mathematical Society, **65**, *213*, pp. 393–396, 1996.
3. NICOLAE CONSTANTINESCU and GEORGE STEPHANIDES, *The GN-authenticated key agreement*, Journal of Applied Mathematics and Computation, **170**, *1*, pp. 531–544, 2005.
4. DAVID WATTS, *Security & Vulnerability in Electric Power Systems*, NAPS 2003, 35th North American Power Symposium, University of Missouri-Rolla in Rolla, Missouri, October 20–21, 2003, pp. 559–566.
5. *** *Common Cyber Security Vulnerabilities Observed in Control System Assessments by the INL NSTB Program*, NSTB, U.S. Department of Energy, 2008.
6. *** *Information Security Policy*, SANS, 2006.
7. IVAN VICTOR KRSUL, *Software Vulnerability Analysis*, PhD Thesis, Purdue University, 1998.

Received March 13, 2012