Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 1 (March), pp. 150-157

Intelligent Management of the Cryptographic Keys

G. Moise, O. Cangea

Gabriela Moise, Otilia Cangea
Petroleum-Gas University of Ploiesti
Romania, 100680 Ploiesti, 39 Bvd. Bucuresti
E-mail: {gmoise,ocangea}@upg-ploiesti.ro

Abstract: With the continuous development of the computers networks, new problems have been posed in the process of keys management in the cryptographic systems. The main element in the cryptographic technologies is the keys management, as the cryptographic algorithms are known, while the keys have to be either secret (for unauthorized users that do not need them), or public (for users that need them). With an efficient cryptographic keys management system and the existing encryption techniques, there may be implemented a proper security system in the informational systems of the organizations. The process of cryptographic keys management consists in the following operations: keys generation, distribution, update, revocation, storage, backup/ recovery, import and export, usage control, expiration, and destruction. The cryptographic keys management techniques depend on the type of the keys, i.e. symmetric or public. Nowadays, the efforts of the researches in the cryptographic keys management are focused on the standardization and interoperability of the keys management. In this paper, the authors analyze the existing keys management systems and standards available for the keys management techniques, emphasizing the advantages and disadvantages of different systems. They also propose a cryptographic keys management model based on the ideas and principles of the INTERRAP architecture (a conceptual model developed by Jőrg Műller for intelligent agents). Also, there are incorporated some intelligent techniques to manage emergency situations, such as keys losing or their improper usage.
Keywords: cryptographic key management, intelligent agents, key management model.

1 Introduction

The key management is the core of a cryptographic system. The processes related to the key management consist in generation, distribution, update, revocation, storage, backup/recovery, import and export, usage control, expiration, and destruction of the cryptographic keys. Practically, the security of the information is assured by keeping secret the private cryptographic keys. Key management consists in a set of protocols that enable to establish and maintain the keying relationships between the entities of a network [6]. The concept of "keying relationship"is defined in [6] as the state wherein parties of the cryptosystems share keying material. According to the type of cryptographic algorithm used in a cryptosystem, there are two situations: key management used in a symmetric cryptosystem and key management used in an asymmetric cryptosystem. In the former case, the sender and the receiver share the same secret key or two keys computationally feasible and in the latter case, there are involved two transformations: one to generate the public key and the other to generate the private key. [6] [11] The techniques used to distribute confidential keys are: key layering, key translation center, and symmetric

key certificate techniques. Key layering comprises the following techniques: master key, key encrypting keys and data keys. *Key Translation Center* (*KTC*) consists in a trusted server, which allows two entities to establish a secure communication using long-term keys. Techniques used to distribute public keys are: point to point delivery over a trusted channel, direct access to a trusted public file (public-key registry), use of an online trusted server, use of an off-line server and certificates, and use of systems implicitly guaranteeing authenticity of public parameters [6]. The advantages of using the keys management in the situation of a public key are: use of a simple key management, on-line trusted server not-required and enhancing the functionality of the system.
In this paper, there is studied the problem of the cryptographic key management in large distributed systems, more specifically, the problems of keys distribution and generation. The main concepts used in this paper are the security domain and the keys graph. The concept of security domain is defined in [7] as "a collection of systems (servers, devices, and so on) that share a common set of keys and are attached to an administered network". In this paper the concept is used in the sense of a collection of entities (to allow an abstractive interpretation) which share a private key. The concept of key graph was introduced in [10] as an arrangement of the keys into a hierarchy and a key server manages all keys. A particular keys hierarchy is the keys tree, which enables to define key management scheme. In this paper it is proposed an intelligent key management model suited to the structure of the network.
The new ideas introduced in this paper are: combining the behaviour agent architecture into the key distribution problem and distribution based on *CRT* (*Chinese Reminder Theorem* [13]).

The paper is structured as follows:

- formalism of the key management problem, that studies the existing key management systems and introduces the concept of security domain graph;
- backgrounds of the intelligent key management model, referring to the *Chinese Remainder Theorem*, as an important calculation algorithm used in order to generate the key management model;
- intelligent key management, proposing a model of the cryptographic key management that may be used in a *SDG* type architecture, based on the principles of the INTERRAP architecture, introducing intelligent agents responsible with the key management in the cryptographic system;
- conclusions, that emphasize the importance of the key management and the advantages offered by the proposed model.

2 Formalism of the key management problem

To formalize the problem of keys management in a computers communication system, there are defined the security domain and a partially ordered relation between security domains. A security domain is a set of the entities (users, data, hardware devices, etc.) which share the same secret key. So, it can be established an equivalence relation between two entities e_1, e_2, according to the following statement:
$e_1 \equiv e_2$ if e_1, e_2 share the same secret key.
The equivalence relation between two entities produces equivalence classes, called security domains.
Let us consider n security domains $SD=\{SD_1, SD_2, ..., SD_n\}$. The set SD contains all the entities of the computer communications system. On the set of the security domains, it can be defined

a binary partially-ordered relationship, using the operator $\prec$.
The relation $SD_j \prec SD_i$ means that the entities of the security domain SD_i have a security clearance higher or equal than that of the entities of the security domain SD_j. For example, the entities from SD_i can decrypt the message received from the entities that belong to SD_j. Also, it is used the expression that the security domain SD_i dominates the security domain SD_j. In this way, one may determine a partially ordered set $(SD, \prec)$, shortly called poset. Messages (data, plain texts) from the security class SD_i are encrypted with the cryptographic key sk_i and data from the security domain SD_j are encrypted with the cryptographic key sk_j. If there is the relation $SD_j \prec SD_i$, the entities of the security domain SD_i have the right to decrypt the cipher text using the cryptographic key sk_i. In contrast, the entities which are parts of the security domains, SD_j , cannot decrypt the messages received from the entities of the security domain SD_i . Also, if the following relations between three security domains $SD_k \prec SD_j$ and $SD_j \prec SD_i$, are true, that means that the entities of SD_j can decrypt the cipher texts received from the entities of SD_k and the entities of SD_i can decrypt the cipher texts received from the entities of SD_j . Consequently, the entities of SD_i can decrypt the cipher texts received directly from the entities of SD_k.
In this manner, there are generated domains hierarchies. A Hasse diagram can represent a poset. This diagram is called in [1] a security class privilege graph *(SCPG)*, in this paper it is used the term of security domain graph *(SDG)*. An example of *SDG* is shown in figure 1 (a - security domains tree *(SDT)*, b - a general security domain graph *(GSDG))*.

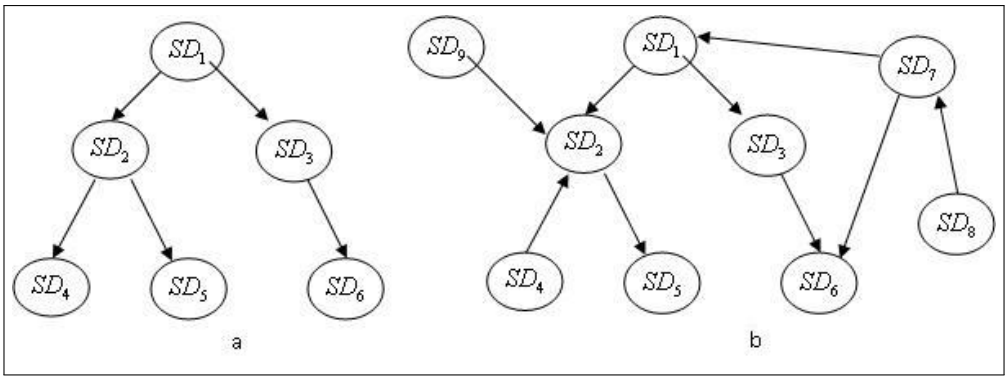

Figure 1: Security Domain Graph

where the following representations have the similar sense (figure 2)
The most simple keys management model assumes the existence of a keys server and if an entity

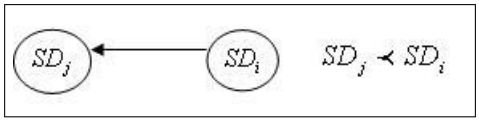

Figure 2: Graphical representation of a partially-ordered relationship between two security domains

needs to decrypt a message, it has to ask the proper key over a secure channel, or each security domain has to store all successors secret keys. These models are not really feasible, because the tendencies of increasing the networks dimensions require large storage spaces and a lot of secure channels. In this way, there is quite a challenge to define a keys management model properly adjusted to the security domain graph.

3 Backgrounds of the Intelligent Key Management Model

There were proposed a lot of cryptographic keys management architecture. To implement Data Encryption Standard, IBM proposed a Key Management Scheme for DES in the 70' years. The cryptographic keys management architecture consists in cryptographic systems connected via a communications network. Each cryptographic system has a cryptographic facility, a cryptographic key data set, a cryptographic facility access program and is using application programs. One solution based on control vector is proposed in [5]. The scheme uses a control vectors which facilities the implementation of owner key management policy and rules. This technique enables key distribution in different environments: peer-to-peer distribution, key distribution center, and key translation center. A list of keys management architecture can be found at [8].
Akl and Taylor proposed the first cryptographic keys assignment scheme to solve problems related to access control in hierarchies (AT scheme). According to AT scheme [1], each security class (the security class contains data and users with the same rights) has associated a secret key and a public parameter. For a relation $SC_j \prec SC_i$, SC_i use the public parameter T_j and the secret key k_i to derive the secret key k_j. The secret key k_i is computed according the formula $k_i = k_0^{T_i}(modM)$, where k_0 is the secret key of the *Central Authority* and M is the product of two secret large prime numbers. T_i is a public parameter with the following property: $SC_j \prec SC_i$, if T_j is a multiple of T_i. In order to generate the public parameters, for T_i it is used the formula $T_i = \prod_{SC_i NOT \succ SC_k} p_k$, where p_k are prime numbers related to each security class.The major inconvenient of the method is the fact that the value of T_i increases and will become impractical (figure 3). Another problem is represented by the question: is the arrangement of the security

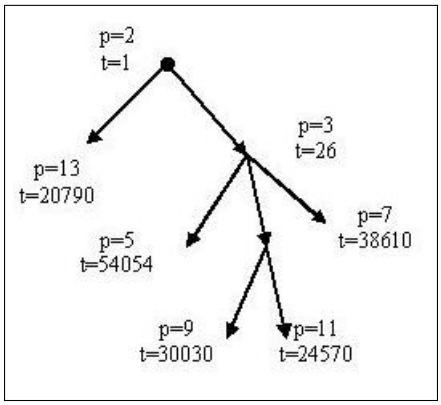

Figure 3: The diagram of generating the T parameter in an AT scheme

domains according to the *SDG*? (that is, when a security domain has more than one parent).
In order to generate a model for the key management, the authors studied the *Chinese Remainder Theorem* that is an ancient, but important calculation algorithm in modular arithmetic. It enables one to solve simultaneous equations with respect to different moduli in a considerable generality. Here is the statement of the problem that the *Chinese Remainder Theorem* solves.

Theorem 1. *Chinese Remainder Theorem [12]. Let*

$m_1, m_2, ..., m_k \in Z$ *with* $gcd(m_i, m_j)=1$,*any* $i,j=\overline{1,k}$, $i \neq j$. *Let* m *be the product* $m=m_1 \times m_2 \times ... m_k$. *Let* $a_1, a_2, ..., a_k \in Z$. *Consider the system:*

$$x \equiv a_1 (mod\ m_1)$$
$$x \equiv a_2 (mod\ m_2)$$
$$...$$
$$x \equiv a_k (mod\ m_k)$$

Then there exists exactly one $x \in Z_m$ solution of the system.

The solution to the system above may be obtained using the following algorithm:
Step 1 For each to i=1 to k calculate $z_i = m_1 \times m_2 \times \ldots \times m_{i-1} \times m_{i+1} \times \ldots m_k$
Step 2 For each to i=1 to k calculate $y_i = z_i^{-1} (mod\ m_i)$
Step 3 Calculate $x = a_1 \times y_1 \times z_1 + \ldots + a_k \times y_k \times z_k$, Return x.

4 Intelligent Key Management

In this paper it is proposed a model of cryptographic keys management that may be used in the architecture of *SDG* type. Each entity of the security domains has associated an intelligent agent *(SDKMA)* responsible with the cryptographic key management in the system (figure 4). where $SD_i = e_{i1} \cup e_{i2} \cup e_{i3} \ldots$, where e_{ij} use k_i and each SD_i is organized according to the schema

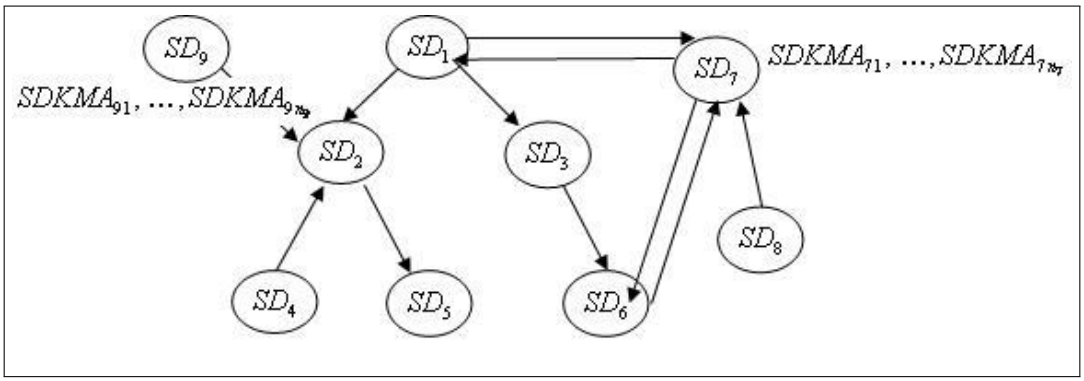

Figure 4: Security Domains Graph and Intelligent Key Management Agents

shown in figure 5.

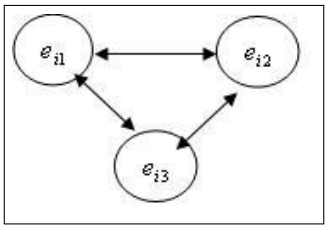

Figure 5: The entities relations in a security domain

Each *SDKMA* is structured according the INTERRAP intelligent agent architecture defined by Műller [9]. INTERRAP architecture is a layered BDI model (belief-desire-intention model) with three layers: behavior based layer, local planning layer, and cooperative planning layer. The behavior based layer contains the reactivity and procedural knowledge used in routine tasks, the local planning layer provides reasoning to realize the local tasks and to produce goal oriented behaviors, and the cooperative planning layer enables and facilities collaborative work with other agents. The structure of *SDKMA* is presented in figure 6.

Behavior planning layer acts in the emergency situations (renewal keys, delete entity, add entity, destroy keys). Local planning layer manages the cryptographic keys within the framework of the security domain. Cooperative planning layer manages the cryptographic keys between security domains.
World KB contains the procedures and functions used in the emergency situations (structure

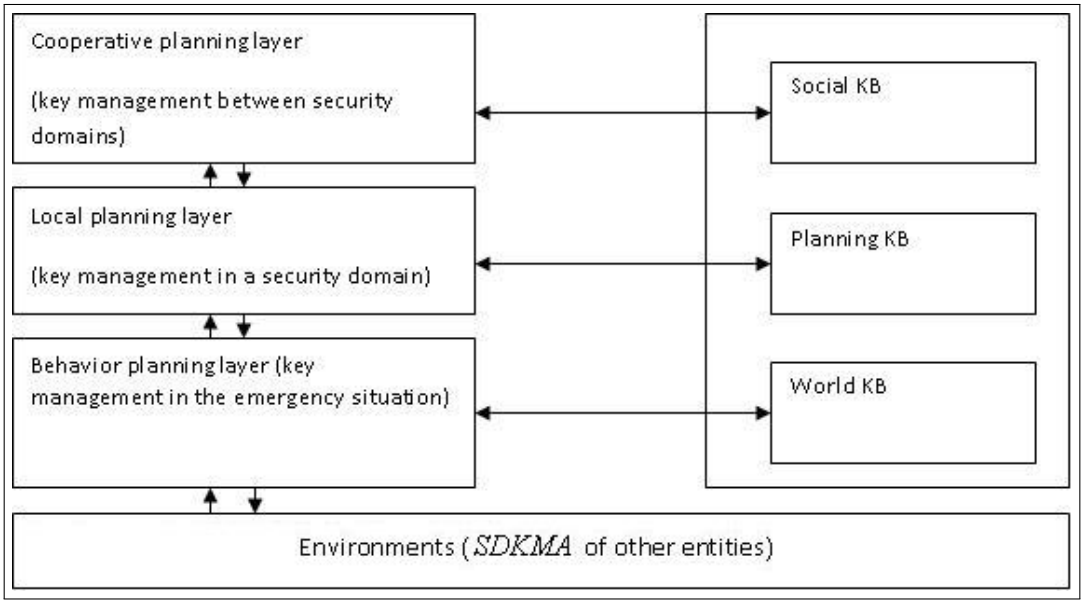

Figure 6: The structure of SDKMA

of the networks and security domains, new arrivals and leaving of the entities). Planning KB contains procedures and functions responsible with the key management within the framework of the security domain and social KB contains procedures and functions used to key management between security domains. These knowledge bases are self-updated; therefore, the keys management model has the property of flexibility that is modifications in the structure of the network cause modifications in the key management model. Keys management within the framework of security domains can be realized according to key server schema. For each security domains, it is selected, in a randomized way, an entity. Its *SDKMA* will play the role of key server. The server stores the key and the associated list of the entities from the security domain. Entities request the key to the server. Upon entity authentication, the server sends the keying material if the entity is authorized. Key management between security domains can be realized according to AT scheme if the node corresponding to the security domain has one parent. The arrangement in security domains offers the advantage of calculating not greater values of parameters T. There were provided solutions to optimize the AT schema in order to compute smaller values of parameters T [4]. If the relations between security domains are according to the scheme presented in figure 7, then one has the following situation: a node has more than one parents and a node has more than two ancestors.

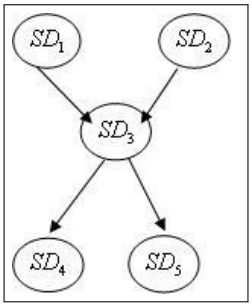

Figure 7: Security domain SD3 with two parents and two ancestors

If an entity of SD_3 broadcasts a message m, than the entities of SD_1 and SD_2 can decrypt the

cipher text c. So, the entities have to know a single decryption key. The asymmetric keys generator has to send to each security domains a private key or each *SDKMA* can compute (cooperative planning layer) the private key using the *Chinese Remainder Theorem*. For each entity (SD_1 and SD_2) it will select a large secret prime number n_1, respectively n_2. The secret key is the solution (in the space $Z_{n1\times n2}$) of the system:
$x \equiv a_1 (mod\ n_1)$
$x \equiv a_2 (mod\ n_2)$, where a_1 , a_2 are public parameters.
To calculate the solution of the system, it can be used the Lagrange method:
Step 1 Find cofactors u_{12}, u_{21} using Extended Euclidian algorithm
Step 2 Calculate $l_1 = u_{21} \times m_2$, $l_2 = u_{12} \times m_1$
Step 3 Calculate the solution $x = a_1 \times l_2 + a_2 \times l_2$

If one entity of SD_4 and one entity of SD_5 broadcast in the same time two encrypted messages c_4 and c_5, the *SDKMA* (the cooperative planning layer) of each entity of SD_3 manage a waiting queue and decrypt the messages using randomized priorities.
The advantages offered by the proposed model are: more securely, because the intelligent property changes any time the parameters used in keys generation, and more efficiently, through the use of a security domain based structure of the network. The dynamism of the network can be easily implemented because the knowledge bases related to each layer of the *SDKMA* are updated with actual information in any moment. Also, in order to generate the keys, the procedures and functions may contain different algorithms and the *SDKMA* can change any time the key distribution algorithms according to the dimensions of the networks or the necessary security level.

5 Conclusions

Starting from the analysis of the existing key management systems and standards available for the keys management techniques, in this paper it is proposed an intelligent cryptographic key management model between security domains, *SDKMA* type, based on the ideas and principles of the INTERRAP architecture, emphasizing the advantages referring to security, efficiency and feasibility. As future directions, one may consider designing a fuzzy model of the cryptographic key management system, but only if this approach would bring important improvements for the model, regarding to the procedures and functions used to generate the keys. Nevertheless, the most secure situation is to hold the keys in secure hardware and perform all processing there [3], being impossible to achieve this goal in large scale networks.

Bibliography

[1] Akl, S.G., Taylor, P.D., Cryptographic solution to a problem of access control in a hierarchy,*ACM Transactions on Computer System*,3 (1), 1983.

[2] Hassen, R. H., Bouabdallah A., Bettahar, H., Challal, Y., Key management for content access control in a hierarchy, *Computer Networks*, 51 3197-3219, 2007.

[3] Lin, J. C., Huang, K. H., Lai, L., Lee, H. C., Secure and efficient group key management with shared key derivation, it Computer Standards and Interfaces, 31, 2009.

[4] MacKinnon, S. , Taylor, P., Meijer, H., Akl, S., An optimal algorithm for assigning cryptographic keys to control access in a hierarchy, *IEEE Transactions on Computers*, C-34 (9), 1985.

[5] Matyas, S. M., Le, A.V. Abraham, D. G., A Key-Management Scheme Based on Control Vectors, *IBM Systems Journal*, Vol. 2, Issue 3, 1991.

[6] Menezes, A., van Oorschot, P., Vanstone, S., Handbook of Applied Cryptography, *CRC Press*, 1996.

[7] Michener, J. R., Acar, T., Security Domains: Key Management in Large-Scale Systems,*IEEE SOFTWARE*, 2000.

[8] Savard, J. J. G., A Cryptographic Compendium, http://www.quadibloc.com/crypto/jscrypt.htm, accessed on the December 5th, 2009.

[9] Müller, J. P., The Design of Intelligent Agents: A Layered Approach. Lecture notes in computer science, *Lecture notes in artificial intelligence*, 1177, Springer-Verlag, 1996.

[10] Wong, C.K., Gouda, M., Lam, S., Secure groups communication using key graphs,*Proceedings of the ACM SIGCOMM'98*, 1998.

[11] Key management in cryptography, http://www.netlab.tkk.fi/opetus/s38153/k2003/Lectures/g33keymgmt.ppt, accessed on December 10, 2009.

[12] Chinese Remainder Theorem, http://www.math.tamu.edu/∼jon.pitts/courses/2005c/470/supplements/chinese.pdf, accessed on December 10, 2009.

[13] Zhou, J., Ou, O. H., Key Tree and Chinese Remainder Theorem Based Group Key Distribution Scheme, *Proceedings of the 9th International Conference on Algorithms and Architectures for Parallel Processing*, ISBN:978-3-642-03094-9, 2009.