

# Enhancing Security in Cyber-Physical Systems Through Cryptographic and Steganographic Techniques

Laura Vegh

Faculty of Automation and Computer Science  
Technical University of Cluj-Napoca  
Romania  
Laura.Vegh@aut.utcluj.ro

Liviu Miclea

Faculty of Automation and Computer Science  
Technical University of Cluj-Napoca  
Romania  
Liviu.Miclea@aut.utcluj.ro

**Abstract**—Information technology is continually changing, discoveries are made every other day. Cyber-physical systems consist of both physical and computational elements and are becoming more and more popular in today's society. They are complex systems, used in complex applications. Therefore, security is a critical and challenging aspect when developing cyber-physical systems. In this paper, we present a solution for ensuring data confidentiality and security by combining some of the most common methods in the area of security – cryptography and steganography. Furthermore, we use hierarchical access to information to ensure confidentiality and also increase the overall security of the cyber-physical system.

**Keywords**—cyber-physical systems, hierarchical access, cryptography, steganography, multi-agent systems

## I. INTRODUCTION

### A. Overview

Today's fast advancing technology is increasing the need for protection of the information contained in a system, of confidentiality of sensitive data. Becoming more and more popular in today's society, cyber-physical systems (CPS) represent an integration of computational and physical processes.

Concerns about security are not new, but as technologies come and go, the methods used to protect data from unwanted access must evolve as well. One of the best known methods is cryptography. The goal of cryptography is to change the form of the message in such a way that it becomes unreadable by any third party trying to access it. There are several types of cryptography that can be used, according to one's needs, to the type of system that needs to be protected. One such type is the symmetric key cryptography which implies that both the sender and the receiver have the same type of key. Another category of encryption algorithms are those with public-key. These are widely used nowadays and they are based on a public-key owned by the sender of a message and used for encryption and a private key owned by the receiver and used to decrypt the message. In this category we find consecrated algorithms such as Data Encryption Standard (DES), Advanced Encryption

Standard (AES) or ElGamal. The algorithm used to encrypt data in the present paper belongs in this category.

Another method to secure data which has come back in the attention of researchers over the past few years is steganography, the art of hiding a message. Unlike cryptography, steganography does not modify the form of the message, it simply hides it. There are several types of steganography, the difference between them come from the medium used to hide the data – the cover file (also called a "stego-file"). Thus, steganography can be technical or linguistic. The technical area includes methods which use as cover files images, videos and audio files. Images are usually preferred as they are easiest to work with, both to hide the message and to transmit it. The linguistic area refers to hiding data inside text. Methods such as using the white spaces between the words, or adding more space at the end of the cover-text are usually preferred, but others are accepted as well, depending on what type of data is being hidden, or what type of system is being handled.

### B. Combining steganography and cryptography

A rather new approach in system's security is to combine cryptography with steganography within the same system in order to obtain a more robust security architecture. From its definition we know that cryptography modifies the form of the message so that it becomes unreadable without the right key to decrypt it. Most encryption algorithms rely on powerful mathematical structures and finding these keys is extremely difficult. However, the encrypted data is visible to any malicious third party. All they would need to focus on is finding the secret key and all data is revealed. With steganography the message is hidden. However, once an external party realizes the existence of the message, the system is considered compromised, regardless of the algorithm needed to extract the actual information. By combining the two methods however, the security of the system is obviously increased. For example, one of the most used ways in current research is to encrypt the message and then hide it in its encrypted form. This way, a third party looking at the system might notice the existence of secret data hidden in the cover

file. However, the system would not yet be compromised, as the extracted message is encrypted. This modality of using security methods together is particularly useful when dealing with complex systems, such as CPS.

### C. Security in cyber-physical systems

Cyber-physical systems are used in many critical applications such as transportation networks, national disaster control systems, gas and water networks and power grids. Their usage is expected to increase reliability, safety and efficiency in critical infrastructures. CPS have many advantages such as enabling individual entities to work together in order to form complex systems; they are efficient, have properties such as dynamic reconfiguration and self-configuration. For this reason security is a crucial aspect when working with CPS especially in areas such as data interpretation, distribution and control of information, availability and confidentiality [20]. When dealing with such complex systems, one cannot expect to find one universal way to ensure security. Rather, one should consider the area of application of the system, the type of data, of communication that needs to be secured and model the solutions accordingly. For example, paper [6] presents a solution for the security of cyber-physical energy systems. In the current paper we propose modeling security through hierarchical access to information, while using as both cryptography and steganography.

## II. PROPOSED APPROACH

The goal of this project was to design a model for a secure cyber-physical system. The chosen approach was that of a hierarchical type of system, where users would have access to information based on their degree of access. Also, the data needed to be secured from unwanted external access and due to the critical type of applications in which CPS are used we chose to use both cryptography and steganography in order to obtain a practically unbreakable system. Hierarchical access to information is a practical way to model a system because it reenacts the form of many ‘real-life’ systems. For example a medical facility, or a teaching facility, they will all have some sort of restriction to access according to a person’s rank. Therefore, using hierarchy to model a cyber-physical system is a fitting choice.

In terms of implementation, we chose to model our CPS as a multi-agent system. Agents are autonomous entities with decision making capabilities. In a multi-agent system there is no global control system, data is decentralized and every agent has incomplete data to solve the tasks. The absence of a global control system means that every agent has only a ‘local view’, in other words no agent can see the entire information in the system. Agent-based modeling can help overcome some of the major issues when it comes to CPS design through agents’ flexibility and their decision making capabilities.

Hierarchical systems are somewhat different from the usual systems and require a different kind of approach, especially when designing the security part of the system. The hierarchy can be modeled in several ways. Because our system should also include an encryption module, we have chosen to model the hierarchy in the system based on a cryptographic algorithm

with divided private key, more specific, ElGamal with differentiated decryption with  $K+1$  access levels. This algorithm is an extension of the ElGamal algorithm [18], a classical public key type of algorithm, where one user – the sender – encrypts a message using the public key and the other user – the receiver – decrypts the message using a private key. The algorithm relies on powerful mathematical structures such as the discrete logarithm, to ensure a good level of security and is based on a cyclic finite group. Its security depends on the complexity of the problem defined within that group. Our algorithm, ElGamal with differentiated decryption with  $K+1$  access levels [10] is based on the same type of mathematical structures as the original. It also uses a public key for the encryption of the message. The difference is that it does not use a single key for the decryption. The private key is ‘divided’, each user receives a certain key and with it he will have access to certain messages. It is this key that defines the access level of a user, the higher the level, the more messages he will be able to decrypt using the given private key. We will not go into further detail regarding the mathematical aspects of the algorithm because this is not the main focus of the paper, but they are presented in detail in [10].

The hierarchical structure formed in this way can be viewed as a tree structure. The users having the lowest level can be seen as the leaves of the tree, while the user with the highest access level is represented as the root. Such a tree structure, with  $k=3$  or four levels of access is illustrated in Figure 1.

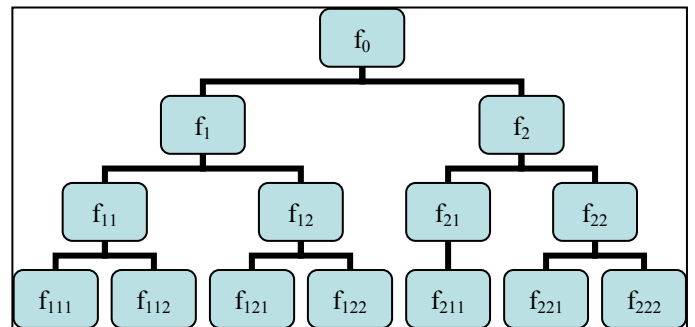


Figure 1. Hierarchical tree structure

Like any ElGamal based algorithm, the one used to secure our CPS has three main phases: key generation, encryption and decryption. However, in this case, the key generation phase is more complex and requires particular attention, as it does not mean simply computing some keys but generating the entire tree structure.

For the implementation of the entire model we have used the Java language with the JADE platform for multi-agent systems. JADE provides all the tools necessary to develop complex multi-agent systems and is FIPA compliant. Also, it is written in Java, which is also the programming language chosen to implement the encryption algorithm.

The system itself was modeled using three main classes: the first one, named ‘KeyManager’ represents the agent that will generate and distribute the keys, hence the name; a second class in which we have implemented the functionalities for all the agents in the tree structure, with the exception of the root

which is implemented in the third class. The differentiation between agents' functionalities helped this division, because the root of the tree has access to any information in the system, while the rest have only limited access.

Generating the hierarchy of the system is an operation carried out by the key manager agent. For the process to begin it is essential to establish the main parameters of the system such as the total number of agents that will be included, the number of agents on the lowest level – the leaves as we call them and how many descendants each of the remaining agents will have. This way, the wanted hierarchy is established. For programming purposes, we name each agent using the numbering convention shown in Figure 1. The root of the tree is agent '0' because it is on the first level and is thus differentiated from the other agents. On the next level we have agents '1' and '2' (the numbering can continue according to the wanted structure). We practically number the agents from left to right on their level. On the next levels, the process is more complex. We wanted to be able to determine the chain of direct ascendants of an agent starting from their name. Therefore, we take the name of the direct parent and then add a number starting from 1 to determine the order on the level. For example, if we want to name the first agent descendant from '1', he will be '11', while its second descendant will be '12'. Descendants from '12' for example, will be '121', '122' and so on. Using this exact process we name all the agents in the system.

Once the wanted structure is established, the next step is to generate and distribute the keys. Since our system does not only rely on hierarchical access to information for its security but also on cryptography and steganography, an important aspect is who secures the messages. Considering the system's structure, we know that the leaves have the lowest degree of access, they have access to the smallest part possible of the entire information. Therefore, attributing the task to secure the information before it is sent throughout the system is a good way to ensure both security and confidentiality. Having established this aspect, it is clear that only the agents on the lowest level will receive the public key for encryption. The keys are generated according to the encryption algorithm ElGamal with differentiated decryption with  $K+1$  access levels. We will not go into further details regarding these computations they can be fully reviewed in [3][10].

The private keys will be sent out to all the agents in the system and they will truly establish the hierarchy, with them each agent has access only to certain messages from the information that enters the system. The key manager agent is again responsible for generating the private keys – one for each user in the system – and distributing them in the right order, taking in consideration the agent's position in the hierarchy. Another important feature of the system derives from the fact that only the leaves can encrypt a message. In order for an agent to be able to decrypt a message, they must be an ascendant of the leaf-agent that performed the encryption. In other words there must be a direct link from the user that secured the message to the one trying to decrypt it. If such a link is not present the user will not have access to the message. The existence of this link should be verified each time a user wants to decrypt a message. This is something that

is established within the encryption algorithm, in the way that private keys are formed. At first, the verification of the existence of this link might seem redundant, since the keys establish the decryption rights, this step contributes to the efficiency of the system because it does not allow an agent to try and decrypt a message with the wrong key. Checking this link is a small process in terms of execution time as compared to performing a decryption with the wrong key.

In terms of implementation, sending and receiving the keys are viewed as sending messages between agents. Also, both are designed as '*SimpleBehaviors*'. In agent terminology, behaviors define the actions an agent will take under a certain event. The two main agent behaviors in Jade are simple and cyclic. The simple behavior is executed only one, while the cyclic one is executed as long as the agent is running. Since sending and receiving keys are one-time events, they will be modeled as simple behaviors. Sending the keys in this way is not ideal as the messages can be easily intercepted. For this reason we added a steganography layer to secure this step. Details regarding this operation will be presented in chapter 3 of the present paper.

Once the keys are sent, the key manager agent is closed and the system is ready for use. Information is sent to the system through a user interface. The receivers are the agents on the lowest level. There are two options when sending data to the system. One, the data can be intended for all the users in the system according to their access rights. That is, data is sent to the leaves that will secure the message and send it to their ascendants, so that each agent receives the parts of the message to which they have right. The second option is data being intended for a certain level. For example, in a system of 7 levels you might want data to get exclusively to level 3, bypassing levels 4-6. If that is the case, the intended level will be sent to the leaves together with the rest of the data. Once here, the message needs to be secure.

The first step towards securing the message is encrypting it. Again, the mathematical computations necessary to encrypt can be studied in detail in [3][10]. Even though the encryption algorithm used is strong and the keys are hard to guess, the message is still visible, any malicious party can tell that secret information is being exchanged. For this reason we have added another security layer using linguistic steganography. The method used will be described in chapter 3 of the present paper.

Finally, sending and receiving secured messages is the main task of the system. Regarding the implementation, this is carried out as a cyclic behavior since it is a task that will be executed as long as the agent is running. Each leaf-agent sends the part of the message they secured to all their ascendants. If the message does not have a specified receiver level, than the leaf sends the data to all its ascendants. Otherwise, the message is only sent to the ascendant on the specified level. As previously explained, when a secured message is received the first thing to do is to verify is the receiver agent is a direct ascendant to the sender agent, or the one which secured the message. An interesting aspect to note is that each level together has access to the entire information. Of course, each agent has access only to a certain part, but the information can

be ‘reconstructed’ on each level if all agents put together their part. This is a very useful aspect when working with cyber-physical systems and not only, as data can be reconstructed when all the users with the same access level work together.

### III. STEGANOGRAPHY

To increase the level of security of our system, we have added steganography in two of its main parts, which seemed to be more vulnerable: sending the keys and sending the encrypted messages respectively. For this we have used two different types of steganography: with image as a cover file for the distribution of the keys and with text as a cover for the distribution of the encrypted text. We will describe in detail both operations in the following subchapters.

#### A. Image steganography for key distribution

Image steganography means hiding the secret information behind an image file. This operation can be performed in many ways. The most common and also the one used in the present paper, is the least significant bit method (LSB) [1][4][8]. It is probably the easiest method and it suits the main requirement of steganography: hiding information in such a way that the change in the cover file is not visible to the naked eye. LSB felt like the best choice for our complex multi-agent system, precisely because it is easy to use and it does not add very much to the overall execution time. Also, the encryption algorithm uses very large numbers for the mathematical computations that take up space and resources. Therefore, we wanted our steganographical algorithm to use as little resources as possible.

Within the system, the distribution of the keys is carried out by an external entity, a “key manager”, which is an important factor to increasing security, as this agent can shut down once the its task is completed. However, sending the keys is carried out as sending a simple message between agents, making the operation very vulnerable to external attacks. A more secure way to perform this task is embedding the keys within images.

In order to embed the keys in an image using the LSB method, the first step is to find their binary representation. This representation will then be written into the LSB of the carrier image. This phase is carried out by the Key Manager agent, who will generate the keys through the mathematical steps of the algorithm and will then embed these keys before sending them to the other agents. In order to send images from one agent to another, they have to first be transformed into bytes. In Jade, this task is made easy by the command *SetByteSequenceContent()* which allows to send an arbitrary sequence of bytes over the content of an *ACLMessage* (messages sent between agents). The agent who receives such a byte sequence can easily make the transformation back to image.

The extraction phase is performed by the receivers, in our case the agents within the hierarchical system. For this, one has to copy the LSBs of the carrier image’s bytes and recombine them in a text file for example. To make the operation easier, we have embedded not also the key, the also its length. This is the first to be embedded and it will also be the first to be extracted at the receiver’s end. In this way, the receiver will

know exactly the length of the data to extract, making the system more secure and solid.

When using image steganography, besides deciding how to carry out the embedding and extraction phase respectively, another important part is choosing the type of image to use. The least significant bit method allows us to work with any type of image: *.gif*, *.png*, *.jpeg* and so on. At first, we chose to work with *.png*. However, this choice was not ideal for our system, because the data to hide – the keys – can be very large. Therefore, the final choice was made for *.jpeg*. The final step in added image steganography to our system was finding a way to differentiate between the types of keys. We know that the system works with both public and private keys. While it would seem less crucial to protect the public key from unwanted view, we believe that exposing even a single key could compromise the system, by giving a third party a view of how these keys look and the possibility to find a way to compute the rest. As stated, in the embedding phase, before embedding the key itself we first added its length. To differentiate between the keys, we have modified an additional bit, after the key. Therefore, when an agent receives an image holding a key, he will first extract the length of the key, followed by the key itself and will then read an additional bit which will let him know whether the key is public or private. The addition of this step makes the entire system more secure because the keys are much harder to detect, the entire process looks like a system exchange of images between agents.

#### B. Text steganography for message protection

Protecting information, or otherwise said, the message sent from one agent to another, is a crucial aspect of our system. To begin with, there is the encryption algorithm, which changes the form of the message making it impossible to read without the proper secret key. However, due to the critical nature of the applications in which cyber-physical systems are used, we wanted to add an additional layer of protection for the data communicated inside the system. Thus, we have added steganography, hiding the encrypted message before sending it to the receiver.

For this step, we have chosen text steganography, which implies hiding information inside a text [7]. Linguistic steganography has many forms and classifications, depending on how the data is hidden. A few examples are line shifting methods – hiding secret messages by vertically shifting the text lines; word shift – messages are hidden by horizontally shifting the words [2]; syntactic methods which use punctuation to hide bits. Another way to categorize linguistic steganography based on how the cover text is manipulated. Here we include format based methods which use the physical formatting of the text and usually modify it by adding white spaces, resizing the text or deliberately misspelling. There are also methods based on random and statistical generation. These methods are used to avoid comparison with the original text and basically mean generating your own cover text.

The text we use to hide the secret message is created by the system using the knowledge that the encrypted message has up to 1024 bits and is formed uniquely of numbers, it will never have any other characters. From here the code is generated

simply by assigning each number from 0 to 9 a word. In order for the code to generate as little suspicions as possible, when generating the code the area of application of the CPS should be kept in mind and the words should be from that domain. For example, if we are modeling a system for medical facility, we should use term from the medical field to generate our code. This step is not mandatory, but it is recommended, because we want the text to look like just a simple message that is being sent over the network so that a third party will not notice there is a hidden message behind the one they see. Words that are within the system's area of application have a better chance of not raising suspicions than random words.

In terms of implementing the hiding operation, this is carried out by the leaf agents after encrypting the message. Each number of the encrypted message will be transformed into its equivalent word. To make the usage of the code easier, after each word a white space is added. The result is then added to a text file. We chose to work with text files because of the dimension of the encrypted message, which combined with our steganographical approach might result in a text too large to fit into a simple String. Once the hiding process is completed, the text file is sent to the receivers.

At the receivers end, upon receipt of a message, the first thing to do is to check the existence of a link to the sender, as described in chapter 2 of the current paper. If such a link exists, the receiver has permission to view the message and the next step is to uncover the secret message. For this, the operations to be performed are the reverse of those performed for hiding the message. Using the known stego-code, each word is transformed back into its equivalent number. To note that, for a faster and easier usage, adding the words to the file and the numbers to form the encrypted message respectively is done from left to right. This way we will always have the correct order, no interchange operation will be necessary. Once the encrypted message is revealed, the decryption process is carried out as described in the previous chapter.

To make the secret code as unbreakable as possible, the code is changed at certain periods of time. For testing purposes we change it every 10 minutes. However, depending how long the system will be running, this time span can be adjusted. For example, if we have a system that should be up for days or weeks, changing this code could be done once a day for example. Also, one should consider an estimate of how often a third party might listen in on the communication channel. If it is expected that the system will be submitted to several attacks in the course of the day, than the code should be changed as often as possible to ensure a high security level.

The functionalities of the system, step by step are as follows:

- Start Key Manager agent
- Establish the system architecture: the number of levels, the number of users and of leaves
- Generate the hierarchical system by starting all the agents and naming them using the established convention
- Generate keys

- Hide each key inside an image
- Distribute the keys
- Upon receipt of a key, the key is extracted from the image
- Close the Key Manager and begin using the system

The following operations are repeated as long as the system is running:

- When an unencrypted message is received, it is sent to the leaves
- Encrypt the message using the algorithm ElGamal with differentiated decryption with  $K+1$  degrees of access
- Hide the message using steganography with the predefined code
- Send hidden message
- Upon receipt of an encrypted message a verification of a link between receiver and sender is performed
- If a link exists, uncover the hidden message
- Decrypt message

#### IV. CONCLUSIONS

The system described in the current paper brings a new approach towards the security of cyber-physical systems. More and more present in critical applications, CPS integrate both physical and computational process and require new and complex approaches when it comes to security. Here we aim at creating a system that ensures data security and confidentiality by combining some of the most common methods in the area of security – cryptography and steganography. Furthermore, we have used hierarchical access to information to ensure confidentiality and also increase the overall security of the cyber-physical system. There are security architectures which rely on hierarchical access, more specifically a tree structure similar to the one we presented, in [14]. Unlike these other systems, the one we designed has an outside entity that establishes hierarchy. Having an outside authority establish the access rights of each user is closer to a real-life hierarchical system and as a consequence is more suitable for the design of cyber-physical systems.

Another distinguishing aspect in our system design is how cryptography is used. The encryption algorithm, ElGamal with differentiated decryption with  $K+1$  access levels is used not only to encrypt messages but also to establish the hierarchy. This is done by assigning each user in the system a private key. With this key they will have access only to certain messages. The higher the level, the more messages they can decrypt with their key. Finally, security was reinforced by using steganography. Defined as the art of hiding information, steganography was used in two main areas of our system: first to hide the secret keys when they are sent to their owners, having as cover medium images; second, to hide the encrypted message, using as a cover medium text. For the image steganography we chose a classical method – the least significant bit as it proved to be a suitable choice for our case,

while to linguistic steganography we used a personal approach – a code created by the system’s owner that uses words from the area of application of the CPS. The system designed in this way is robust, reliable and flexible. There is no fixed number of users as long as a balanced tree structure can be formed.

#### REFERENCES

- [1] Anwar H. Ibrahim, Waleed M. Ibrahim, “Text Hidden in Picture using Steganography: Algorithm and Implications for Phase Embedding and Extraction Time”, *International Journal of Information Technology & Computer Science*, Volume 7, No. 3, January/February 2013
- [2] Abdelraham Altigani, Bazara Barry, “A hybrid approach to secure transmitted messages using advanced encryption standard (AES) and word shifting protocol”, *International Conference on Computing, Electrical and Electronic Engineering (ICCEEE)*, 2013
- [3] Laura Vegh, Stelian Flonta, Liviu Miclea, “Secure multi-agent system using the ElGamal decryption algorithm with (K+1) degrees of access”, *The 6th International Conference on Security for Information Technology and Communications*, Bucharest, Romania, June 2013
- [4] Mamta Juneja, Parvinder Sandhu, “An improved LSB based Steganography with Enhanced Security and Embedding/Extraction”, *3rd International Conference on Intelligent Computational Systems*, Hong Kong, China, January 2013
- [5] Riadh W. Y. Habash, Voicu Groza, Kevin Burr, “Risk Management for Power Grid Cyber-Physical Security”, *British Journal of Applied Science & Technology*, Volume 3, Issue 4, July 2013
- [6] Khan Usman, Stakovic Aleksandar, “Security in cyber-physical energy systems”, *Workshop on Modelling and Simulation of Cyber-Physical systems (MSCPES)*, 20-23 May 2013
- [7] M. Grace Venice, Prof. Tv. Rao, “Hiding the Text Information using Steganography”, *International Journal of Engineering, Research and Application*, Vol. 2, Jan-Feb 2012, pp. 126-131
- [8] Padmashree G., Venupogapala P. S., “Audio Steganography and Cryptography: Using LSB algorithm at 4th and 5th LSB layers”, *International Journal of Engineering and Innovative Technology*, Volume 2, Issue 4, October 2012
- [9] Oluwaseyitanfunmi Osunade, “A Java-based Data Encryption Application for Network Communication”, *International Journal of Engineering Science and Innovative Technology (IJESIT)*, Volume 1, Issue 2, November 2012
- [10] S. Flonta, V. V. Patriciu, L. C. Miclea, “Metode criptografice pentru sisteme structurale”, U.T. Press, Cluj-Napoca, 2011
- [11] Tayana Morkel, “Image Steganography Applications for Secure Communication”, *Dissertation Thesis*, University of Pretoria, May 2012
- [12] Sonsare Pravin, “Stegano-Cryptosystem for Enhancing Biometric-feature Security with RSA”, *International Conference on Information and Network Technology*, Singapore, 2011
- [13] Mustafa AL-Fayoumi, Ja’afar AL-Saraireh, “An Enhancement of Authentication Protocol and Key Agreement (AKA) For 3G Mobile Networks”, *International Journal of Security (IJS)*, Volume 5, Issue 1, 2011, pp. 35-51
- [14] V. Valli Kumari, D.V. NagaRaju, K. Soumya, K.V.S.V.N. Raju , “Secure Group Key Distribution Using Hybrid Cryptosystem”, *Machine Learning and Computing*, pp. 188-192, February 2010
- [15] Dipti Kapoor Sarmah, Neha Bajpai, “Proposed System for Data Hiding using Cryptography and Steganography”, *International Journal of Computer Applications*, 2010
- [16] Partha Pal, Rich Shantz, Kurt Ruhloff, Joseph Loyall, “Cyber-Physical Systems Security – Challenges and Research”, *BBN Technologies, Cambridge*, Available at: [http://cimic.rutgers.edu/positionPapers/CPSS\\_BBN.pdf](http://cimic.rutgers.edu/positionPapers/CPSS_BBN.pdf)
- [17] Dr. Clifford Neuman, “Challenges in Security for Cyber-Physical Systems”, Available at: <http://cimic.rutgers.edu/positionPapers/CPS-Neuman.pdf>
- [18] ElGamal Encryption: [http://en.wikipedia.org/wiki/ElGamal\\_encryption](http://en.wikipedia.org/wiki/ElGamal_encryption)
- [19] Iuon-Chang Lin , Chin-Chen Chang, “A Novel Digital Signature Scheme for Application of Document Review in a Linearly Hierarchical Organization”, *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, August 2008, , Proceedings pp. 1367-1370
- [20] A. A. C’ardenas, S. Amin, “Secure Control: Towards Survivable Cyber-Physical Systems”, *The 28th International Conference on Distributed Computed Systems Workshops*, 2008, pp. 495-500