15

Using Petri Net for Modeling and Analysis of a Encryption Scheme for Wireless Sensor Networks

Hugo Rodríguez, Rubén Carvajal, Beatriz Ontiveros, Ismael Soto
Universidad de Santiago de Chile
Chile

Rolando Carrasco
Newcastle University
UK

1. Introduction

Nowadays the wireless sensor networks (WSN) are increasingly being required for applications where the data reliability needs to be duly guaranteed. Due to this, many research works have been directed in this sense. However, WSN security is not an entirely solved issue.

The aim on this chapter is to use Petri Nets (PN) to model and analyze the validity of an encryption scheme applied to WSN. Usually the modeling of communication systems can become quite complex, most of the time is necessary to consider a large number of variables and mathematical models to describe them. Nevertheless, structural properties of a system and dynamic characteristics of its behavior can not be properly derived from those models. On the other hand, PN can model dynamic characteristics such as synchronization and concurrency.

PN were developed by Carl A. Petri in 1962 as a mathematical tool for the study of communication with automata. Its further development was facilitated by the fact that PN can be used to model properties such as process synchronization, concurrent operations, conflicts or resource sharing and deadlock freedom among others (Reisig, 1986), (Reisig & Rozenberg, 1998). In literature we found works where PN have been used in order to model and analyze several related aspects of a WSN. In (Chen et al., 2008) the authors use PN to determine the minimal necessary number of devices in a coal mine, in order to establish the location of miners in case of accident. There are many works related to power constrains, most recently work (Liu, Ren, Lin & Jiang, 2008) deals with the power saving problem, where the authors have proposed four sleeps scheduling schemes for different sorts of and analyze each of them by Stochastic Petri Net (SPN). By using the steady state probability matrix of the SPN models is obtained the average power consumptions and events delays. Also, there are many references devoted to communication protocols and recently in reference (Haines et al., 2007), PN are used as a formal verification technique of a MAC protocol. Where a case study is presented applying this technique to IEEE 802.11 centralised control mechanisms to support delay sensitive streams and fading data traffic. In reference (Pengand et al., 2007) author have used the PN to model the Session Initiation Protocol (SIP) and accomplished its verification.

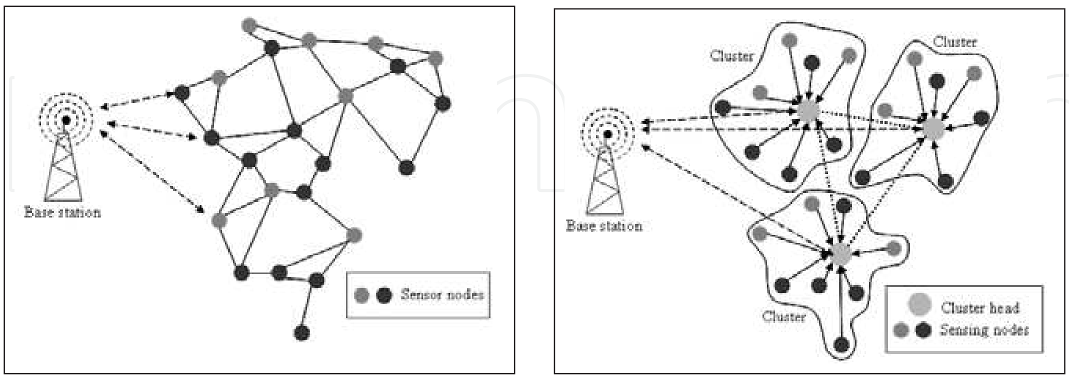
Another interesting work is presented in (Liu, Ye, Zhang & Li, 2008) where the 802.11i 4-way handshake protocol is analyzed utilizing High-level Petri Nets, confirming that the protocol is vulnerable to Denial-of-Service attack during handshake, then the authors propose an improved key management scheme.
The goal of the chapter is to present the most important part of the PN theory and to describe the possibilities of practical applications on WSN.
This paper is organized as follows: In section 2 the WSN are introduced, theirs principal characteristics and basic architectures, a brief introduction about to EC and LDPC codes and the basic concepts of PN are explained as well. In Section 3, we focus on the description of a communication model considered in order to a WSN. In Section 4, deal with the modeling phase by means of PN for our system. A PN model for the communication system and cryptography protocol are developed. Finally, we dedicated both a section to the conclusions and another to the references.-

2. Basic Concepts

In this section we developed briefly the topics that we are needed to better understand this work. For a deeper understanding look up the cited bibliography in each case.

2.1 The WSNs

The WSNs are defined as compound networks of a large number of tiny devices called sensor nodes, which have limited processing power, storage, bandwidth, and energy. In addition, a WSN might be often deployment on a large scale throughout a geographic region in hostile environments.
According to how sensors are grouped and how the information of sensors is routed through the network, there are two basic architectures to WSN, flat and hierarchical. In a flat architecture all nodes have almost the same communication capabilities and resource constraints and the information is routed by each sensor. In a hierarchical architecture, the sensor nodes are grouped in clusters where one of the member nodes is the "cluster head". This node is responsible for management and routing tasks. Fig. 1(a) shows a flat WSN model and a hierarchical WSN configuration is showed in Fig. 1(b).



(a) Flat WSN Model

(b) Hierarchical WSN Model

Fig. 1. Architectures of Wireless Sensor Network

Further regarding the properties of the sensing interfaces of their nodes the WSNs can be classified in homogeneous (HoWSN) or heterogeneous (HeWSN). The HoWSN are characterized by the fact that all network nodes have the same properties unlike the heterogeneous (HeWSN) where nodes of different nature like temperature and movement sensors coexist.

Fig. 1(b) shows a HeWSN. Including the base station there are three types of nodes: base nodes, header nodes and sensors nodes. The base station is the interface between an application and a sensor network. The application network implements all the programs which manage the sensor network. The header nodes are devices with capability to route information that comes from other nodes. The sensor nodes are those that perceive changes produced in the environment such as temperature, humidity, light, movement among others.

Fig. 1(b) also shows the clustered sensor network. The sensor network is divided into several clusters. The headlines represent connection between the header nodes and the base station. Then header nodes manage the sensor nodes and route messages from others nodes; the sensor nodes are scattered in the monitoring field and it implements data process such as reading, encrypting, encoding and transmission.

As shown in the Fig. 2, a sensor node is composed of four basic components: sensing unit, processing unit, transceiver unit and a power unit.

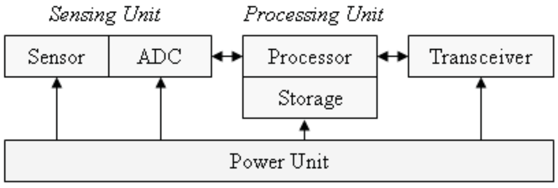


Fig. 2. Components of a Sensor

The sensing units are usually composed of two sub-units: Sensors and analogy-to-digital converters (ADCs). The analogy signals perceived by the sensor which are based in the observed phenomenon are converted to digital signals by the ADC, and then they are nourished to the unit of processing. The process unit, that is generally associated to a little storage device, manages / handles the procedures which make the sensor node collaborates with the others nodes in order to carry out the assigned sensor task.

2.2 Elliptic Curves

In our scheme we are using an elliptic curve defined over a finite field of characteristic two. In this case we only considered a representation of field elements and exist efficient way to effect arithmetic operations. The option we choose is to use optimal normal base (ONB). In which the add operation can accomplish by means of the operation XOR, without cost; squaring can be performed simply by a cycle shift of the coordinates of an element, hence, in hardware, it is almost cost free. A normal basis multiplication is not so simple but always more efficient than in other base. Reference over multiplication in ONB can be found in (Reyhani, 2003; 2006).

According to (Hankerson et al., 2004), an elliptic curve over a finite field F_{2^m} is defined as the set of points that satisfies the following equation:

$$E : y^2 + xy = x^3 + ax^2 + b \quad (1)$$

where $a, b \in F_{2^m}$ and $b \neq 0$ in F_{2^m}. The set of solutions (x, y) joined with a point at infinity, and special addition operation define an abelian group, are called the elliptic curve group.

An Elliptic Curve Cryptosystem (ECC) bases its security in the Elliptic Discrete Logarithm Problem (EDLP), that is, in the Discrete Logarithm Problem (DLP) defined on the group of

rational points of an elliptic curve.
Given an elliptic curve E over a finite field F, a point $G \in E(F)$ and another known point Q which is multiple of that point G. the problem is to find the integer n such that $nG = Q$. This problem is computationally difficult to solve.

2.3 LDPC codes

The Low-density parity check (LDPC) codes were introduced by Robert Gallager in 1962 (Gallager, 1962), but with computational capabilities available then were dismissed. In the mid 90 were rediscovered by MacKay (MacKay, 1990) reaching great popularity by his performance closely to Shannon limit.
LDPC codes are a class of linear block codes characterized to have its sparse parity-check matrix which contains only a few 1's in comparison to the amount of 0's. In other words, the degree of all nodes is low. These codes can be represented by a Tanner bipartite graph consisting of two sets of nodes $\{c_i\}$ and $\{f_i\}$, respectively called variable nodes and check nodes. See example in Fig. 3.

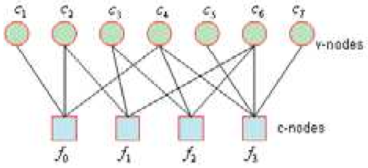


Fig. 3. Tanner Bipartite Graph

There are two kinds of LDPC codes: regular and irregular. For regular LDPC codes, all nodes of the same type have the same degree. For irregular LDPC codes, the degree of each set of nodes is chosen according to some distributions. A code vector c is obtained multiplying a message vector m by a generate matrix G

$$c = m * G \tag{2}$$

The corresponding check matrix H has the property to be constructed with independent lineally rows vectors has which form a subspace of the subspace generated by the rows vectors of G. This signifies than each vector code satisfies the condition:

$$H * c = 0 \tag{3}$$

property that enables the decoding.
In (Castiñeira & Guy, 2006) we found a chapter dedicated to the binary LDPC codes. Description, construction and decoding algorithms can be studied here. Other reference about to construction of codes is (Carrasco & Johnston, 2008).
The decoding of LDPC codes is based on sum product algorithm (SPA), also called belief propagation algorithm (BPA), or message passing algorithm (MPA) which iteratively updates the posterior probabilities of bit nodes.
If we defined $q_{i,j}(b), b \in \{0,1\}$, as the probability computed based both on the received signal y_i and the message $r_{j',i}(b)$ passed from the neighbors check nodes c_i excluding the check node j. Also, we defined $r_{i,j}(b)$ as the probability computed based on the message $q_{j',i}(b)$ passed from the neighbors variable nodes v_j excluding the variable node i .
The MPA in the probability domain then proceeds as follows.

1. Estimate the noise power σ^2 . Then for $i = 1, 2, \ldots, n$, initialize $P_i(b) = prob(c_i = b/y_i)$.
2. Set $q_{ij}(b) = P_j(b)$ if the variable node i and the check node j are connected.
3. Update $r_{ij}(b)$ using

$$\begin{cases} r_{ji}(0) = \frac{1}{2} + \frac{1}{2}\prod_{i' \in v_{j\setminus i}} (1 - 2q_{i'j}(1)), \\ r_{ji}(1) = 1 - r_{ji}(0) \end{cases}$$

4. Update $q_{ij}(b)$ using

$$q_{ij}(b) = K_{ij}P(b) \prod_{j' \in c_{i\setminus j}} (r_{j'i}(b)); \;\; b = 0, 1$$

   where K_{ij} is chosen to meet $q_{ij}(0) + q_{ij}(1) = 1$.
5. Compute the posterior probability of the code bit c_i using

$$Q_i(b) = K_i P_i(b) \prod_{j \in c_i} (r_{ij}(b)); \;\; b = 0, 1$$

   where K_i is chosen to ensure that $Q_i(0) + Q_i(1) = 1$.
6. For $i = 1, 2, ..., n$ set up:

$$\begin{cases} \widehat{c} = 1 \; if \; Q_i^{(1)} > Q_i^{(0)}, \\ \widehat{c} = 0 \; if \; Q_i^{(1)} < Q_i^{(0)} \end{cases}$$

If $\widehat{c}H^T = 0$ or the number of iterations equals the maximum limit, stop; else, go to Step 2. Here, H is the parity check matrix of the LDPC codes.

2.4 Petri nets

A PN is identified as a particular kind of bipartite directed graph populated by three types of objects. They are places, transitions, and directed arcs connecting places and transitions. Formally, a PN can be defined by:

$$\mathcal{N} = \langle \mathcal{P}, \mathcal{T}, \mathcal{F}, \mathcal{W} \rangle$$

where:

a. $\mathcal{T}$ is the set of places, $\mathcal{T}$ is the set of transitions, $\mathcal{P} \cap \mathcal{T} = \emptyset$
b. $\mathcal{F} \subseteq (\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})$ is flow relationship
c. $\mathcal{W} : \mathcal{F} \to \mathcal{N}$ assigns a weight or multiplicity to each arc.

Usually, actions are associated to transitions and conditions are associated to places. A transition is enabled if each input place of contains at least a number of tokens equal to the weight of the arc connecting to. When an enabled transition fires, it removes tokens from its input places and deposits them on its output places. PN models are suitable for representing systems that exhibit concurrence, conflict, and synchronization.

Some important PN properties include a boundness (no capacity overflow), liveness (freedom from deadlock), conservativeness (conservation of non consumable resources), and reversibility (cyclic behavior). The concept of liveness is closely related to the complete absence of deadlocks. A PN is said to be live if, no matter what marking has been reached from the initial marking, it is possible to ultimately fire any transition of the net by progressing through further firing sequences. This means that a live PN guarantees deadlock-free operation regardless of the firing sequence. Validation methods of these properties include reachability analysis, invariant analysis, reduction method, siphons/traps-based approach, and simulation (Reisig, 1986), (Reisig & Rozenberg, 1998).

3. System Description

Fig. 4 shows a block communication model we propose to communicate a header node and the base station in a hierarchical WSN. This model should use an elliptic curve encryption scheme and LDPC code in the process of messages transmission. On the left side we have the header node, transmitter of the message, and on the right side we have the base node, receiver of the message.

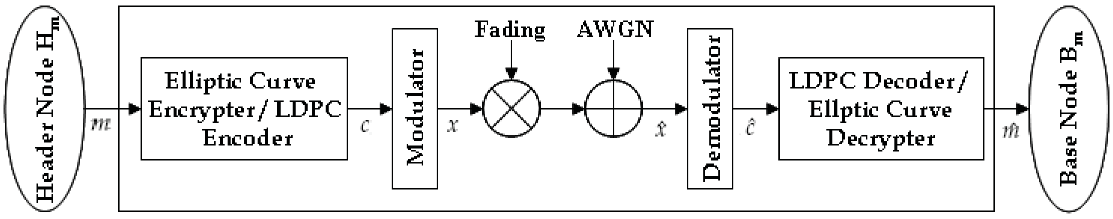


Fig. 4. Communication Model

In this model m represents the transmitted binary message sequence. This sequence gets into the system to an Elliptic Curve Encrypter/LDPC Encoder block, where a transformation function converts them to an output code c, called usually a codeword. This codeword is the input sequence to a modulator block which converts them in a modulated signal x, which goes through the communication channel. Here the signal is disturbed by the noise coming from transmission medium and suffers the fading effects own of the wireless channels. From this process derives a corrupted signal $\hat{x}$, estimated from the signal x. Then the signal $\hat{x}$ enters to a demodulator block to be converted over again to a binary sequence. This sequence is called $\hat{c}$ and is an estimate of the codeword c. The LDPC Decoder / Elliptic Curves Decrypter block receives the binary sequence and transforms them by means of an inverse function in a message sequence $\hat{m}$ which is an estimate of the original message m. The base station receives $\hat{m}$ as a valid message and the process finishes.

4. Modeling and Analysis

In this section we develop some derived PN models since the communication system we have presented in the section 3. Firstly we analyze the system from the perspective of the communication among process. Then, we model a cryptography scheme which would be based on an elliptic curve, which at the same time is defined on a finite field.

4.1 Communication System

Fig. 5, shows the PN diagram of the communication system consistent with the model we presented on Fig. 4.

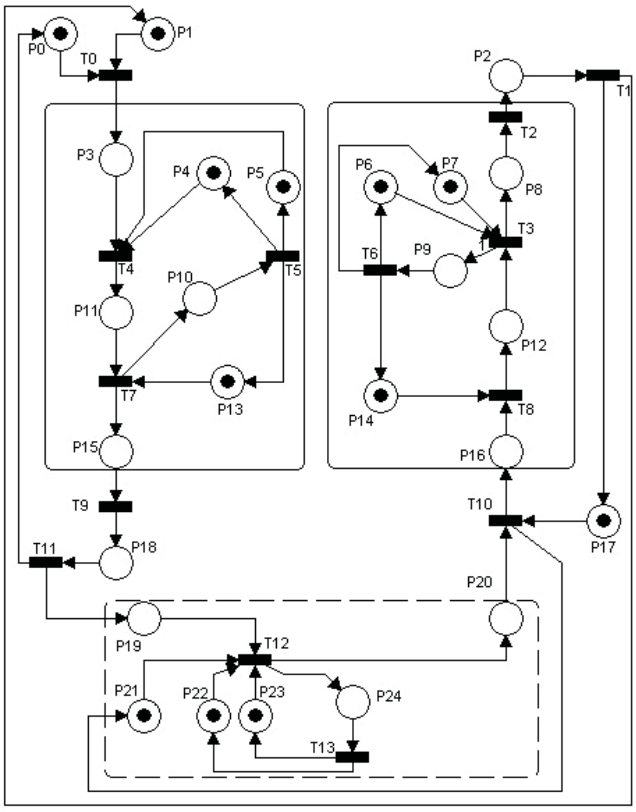


Fig. 5. Petri Diagram of the Communication System

We have made some assumptions regarding to the system in order to simplify the model.

1. Both transmitter and receiver are idle before begin the communication;
2. The channel is always available, there is not fight in access to the channel;
3. The public and private keys associated with the EC, are considered resources always available during a communication instance;
4. Both LDPC encoder and decoder are considered resources always available for the system.

The meaning of each *place* and *transition* is described in the Table 1 and 2 respectively. The initial marked which defines the system behavior denotes the following: The *places* $P0$ and $P1$ marked with one token tells us there is a new message and the transmitter is idle in order to transmit it. The places $P4$, $P5$ denotes resource availability to achieve the encryption process on the sender side. One mark in the *place* $P4$ will tell us that we count on a suitable EC and the another one in the *place* $P5$ will tell us as well that we dispose a valid public key to work on. In the same way the marks on the *places* $P6$ and $P7$ mean resource availability in order to achieve the decryption process on receiver side. In this case, one mark in the *place* $P6$ tells us that we dispose of the EC and anther one in the *place* $P7$ tells us that we count on a valid key to work on, here, the private key of the receiver.

On the other hand we have the coding resources, in the graphic represented by the *places* $P13$ and $P14$. A token allocated on the *place* $P13$ denotes LDPC encoder availability whereas the another one allocated on the *place* $P14$ shows the LDPC decoder is available. The *places* $P22$ and $P23$ represent noisy and fading processes into the channel. The tokens on these *places* denote presence of those processes. The last *place* marked from the initial marked of the net is $P21$ which means that the channel is idle and free to transmit a signal.

Place	Description	Place	Description
P0	New message	P13	LDPC encoder (sender side)
P1	Transmitter idle	P14	LDPC decoder (receiver side)
P2	Received message	P15	Encoded message (codeword)
P3	Incoming message	P16	Demodulated message
P4	EC on the sender side (resource)	P17	Receiver idle
P5	Valid key (receiver public key)	P18	Modulated message
P6	EC on the receiver side (resource)	P19	Channel input
P7	Valid key (receiver private key)	P20	Channel output
P8	Decrypted message	P21	Channel Idle
P9	LDPC decoder and EC Restoring (receiver)	P22	Fading process
P10	LDPC decoder and EC Restoring (sender)	P23	Noise process
P11	Encrypted message	P24	Channel parameters restoring
P12	Decoded message		

Table 1. Places Description of PN in Fig. 5

Basically the PN shows different status that a message has to go through the communication process. The *transitions* represent functions that transform the message into different phases of process. Thus we have as follows:

1. Initially, the *transition* $T0$ is triggered, since the initial marked on the places $P1$ and $P2$ enable to do it. Then the message passes to *incoming message* status (place $P3$ is loaded with a token);

2. The *transitions* $T4$ triggers, because of the tokens in its incoming places $P3, P4, P5$ enables it. Then these ones are removed, new tokens are generated and allocated on the single output place $P11$ passing to the *encrypted message* status.

3. In the same way the *transition* $T7$ triggers removing the tokens from the places $P11$ and $P23$ and loading another two into places $P10$ and $P15$. The token in $P10$ makes conditional the resources restoring and the one in $P15$ turns the message into an *encode message* status. On the other one the *transition* $T5$ triggers tokens which restore both the curve and encode parameters.

4. A token on $P15$ enable the *transition* $T9$ which accomplishes his trigging passing to *modulated message* status, (place $P18$ is marked with a token).

5. Enabled now, the *transition* $T11$ trigger removing the token from $P18$ and allocating one into the place $P0$ and another one into the place $P19$ (*message in the channel input*).

Transition	Description	Transition	Description
T0	Transmit message	T7	Encode message
T1	Wait for message	T8	Decode message
T2	Deliver message	T9	Modulate message
T3	Decrypt message	T10	Demodulate message
T4	Encrypt message	T11	Send message to channel
T5	Restore parameters	T12	Perturb message
T6	Restore parameters	T13	Reset channel parameters

Table 2. Transitions Description of PN in Fig. 5

All these processes above encrypting-encoding-modulating belong to sender side, and in inverse proportion to this we have the receiver side; demodulating-decoding-decrypting. Nevertheless, we must as well to describe the channel process. Simply to say that if both places *P*9 and *P*21 are loaded each with a token, besides the places *P*22 and *P*23 then the transition *T*12 trigger. In other words, if there is a message in the channel input and this is idle, on the other hand there are both the noisy and fading processes, then the disturbance process is enabled which generate distortion in the message.
We perform two types of analysis over the PN model: a structural and a graph based analysis. The aim of the first is to show that the amount of states is finite (boundedness) and that all the represented activities can be done (liveness). The aim of graph based analysis is to determine the absence of dead states. The analysis was performed with the Petri net analyzer tool INA Roch & Starke. (1999).

Table 3 shows the structural analysis of the model of Fig. 5 where we can see there are eleven P/Invariants including all places, so we can conclude that the net is bounded. There are one T/Invariant (Table 7) covering all transitions, so the model is live. Finally the resulting reachability graph has 80 states large enough to draw it here.

Place Invariants	
1	*P4 + P5*
2	*P6 + P7*
3	*P6 + P9*
4	*P4 + P10 + P11*
5	*P10 + P13*
6	*P6 + P12 + P14*
7	*P2 + P8 + P22 + P16 + P17*
8	*P0 + P3 + P4 + P10 + P15 + P8*
9	*P20 + P21*
10	*P22 + P23*
11	*P22 + P24*
Transition Invariants	
1	*T1 + T2 + T3 + T4 + T5 + T6 + T7 + T8 + T9 + T10 + T11 + T12 + T13*

Table 3. Place and Transition Invariants of the Net in Fig. 7.

4.2 Cryptographic Protocol

We have used a sequence diagram (Fig. 6) to describe part of a secure communication protocol. It is based in the public key encryption scheme proposed by Menezes-Vanstone in (Menezes et al., 1993) but also we have considered (Ontiveros et al., 2006). Specifically is considered the communication between a sensor and header node.

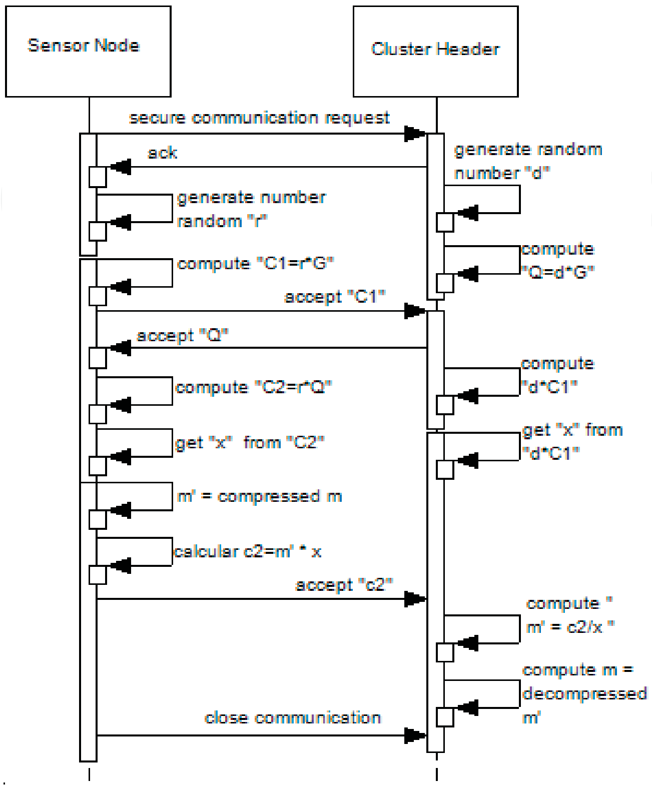


Fig. 6. Sequence Diagram of the Encryption and Decryption Process

It is important to consider the moments which operations are accomplished, with the aim of minimizing communication times. The sensor and header node should have same elliptic curve cryptoprocessor embedded in their architecture and both should know also a rational point G which is a curve points generator.

The sequence diagram of Fig. 6 describes the behaviour of the protocol, as follows: when the sensor node wishes to send a message m to their header node, sends a *security communication request*. The header node reply with a *acknowledgment of receipt (ack)* leaving the communication established. Both nodes performs concurrent tasks, the sensor node generates a random number r and computes $C1 = r * G$, meanwhile the header node generates a random number d and computes $Q = d * G$. After, when the sensor node has completed its task it sends the point computed $C1$ to the header node that receive it and sends the point Q to the sensor node. Again, both nodes accomplish concurrent task with the received components as showed in the Fig. 6.

The sensor node computes $C2 = r * Q$ and it get the coordinate x from $C2$ to compute it with the message m which compress and send it to the header node. Meanwhile, the header node B computes the point $d * C1$ received and it gets its coordinate x which it will use next to the decryption process. The next step consist in that collector node decompress and decrypts the message m computing $m = C2/x$. Finally, in order to reduce the complexity, we assume that the sensor node close the communication.

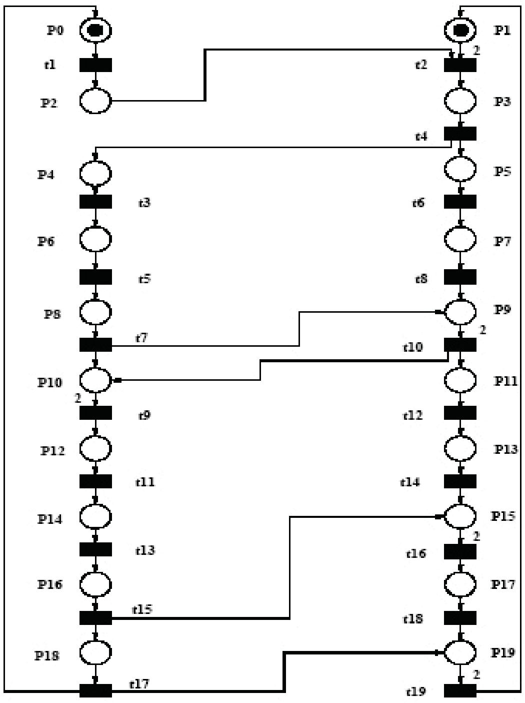


Fig. 7. Petri Diagram of the Cryptography Scheme

The Fig. 7 represents the PN model of the communication protocol discussed previously. Fig. 4 shows the meaning of each place and Table 5 shows the meaning of each transition.

We perform two types of analysis over the PN model: a structural and a graph based analysis. The aim of the first is to show that the amount of states is finite (boundedness) and that all the represented activities can be done (liveness). The aim of graph based analysis is to determine the absence of dead states. The analysis was perfomed with the Petri net analyzer tool INA Roch & Starke. (1999).

Place	Description	Place	Description
P0	Idle	P10	Waiting Q
P1	Waiting for communication request	P11	Q sent
P2	Ready to transmits	P12	$C2$ computed
P3	Security communication request received	P13	$d * C1$ computed
P4	Communication request reply received	P14	x obtained
P5	Communication request sent	P15	Waiting $c2$
P6	Random Number r generated	P16	$c2$ computed
P7	Random Number d generated	P17	$c2$ decompressed
P8	$C1$ computed	P18	$c2$ sent
P9	Q computed and waiting $C1$	P19	Communication closed

Table 4. Meaning of Places in Fig. 7

Table 3 shows the structural analysis of the model of Fig. 7 where we can see there are eleven P/Invariants, including all places, so we can conclude that the net is bounded. There are one T/Invariant (Table 6 covering all transitions, so the model is live. Finally the resulting reachability graph represented by the set shown in Table 7 has 42 states.

Place	Description	Place	Description
t1	Start communication request	t11	Get coordinate x from $C2$
t2	Processing communication request	t12	Compute $d * C1$
t3	Generate random number r	t13	Compute $c2m * x$
t4	Sending ack	t14	Get x from $d * C1$
t5	Compute $C1 = r * G$	t15	Compress and Send $c2$
t6	generate random number d	t16	Decompress $c2$
t7	Sending $C1$	t17	Close communication
t8	Compute $Q = d * G$	t18	Compute $m = c2/x$
t9	Compute $C2 = r * Q$	t19	Return to wait communication request
t10	Sending Q		

Table 5. Transitions Description of PN Fig. 7

	Place Invariants
1	$P0 + P2 + P3 + P4 + P5 + P6 + P7 + P8 + P9 + P10 + P12 + P14 + P16 + P18$
2	$P0 + P1 + P2 + P3 + P4 + P5 + P6 + P7 + P8 + P9 + P10 + P11 + P12 + P13 + P14 + P15 + P16 + P17 + P19$
	Transition Invariants
1	$t1 + t2 + t3 + t4 + t5 + t6 + t7 + t8 + t9 + t10 + t11 + t12 + t13 + t14 + t15 + t16 + t17 + t18 + t19$

Table 6. Place and Transition Invariants of the Net in Fig. 7

M0	=	*PO + P1*	M21	=	*P18 + P19*
M1	=	*P1 + P2*	M22	=	*P0 + 2P15 + P19*
M2	=	*P3*	M23	=	*P2 + 2P15 + P19*
M3	=	*P4 + P5*	M24	=	*P13 + P15 + P18*
M4	=	*P5 + P6*	M25	=	*P0 + P13 + P15 + P19*
M5	=	*P5 + P8*	M26	=	*P2 + P13 + P15 + P19*
M6	=	*P7 + P8*	M27	=	*P14 + P15*
M7	=	*P7 + P9 + P10*	M28	=	*P11 + P16*
M8	=	*2P9 + P10*	M29	=	*P11 + P15 + P18*
M9	=	*2P10 + P11*	M30	=	*P0 + P11 + P15 + P19*
M10	=	*P11 + P12*	M31	=	*P2 + P11 + P15 + P19*
M11	=	*P11 + P14*	M32	=	*P12 + P13*
M12	=	*P13 + P14*	M33	=	*P12 + P15*
M13	=	*P13 + P16*	M34	=	*2P10 + P13*
M14	=	*P15 + P16*	M35	=	*2P10 + P15*
M15	=	*2P15 + P18*	M36	=	*P8 + P9*
M16	=	*P17 + P18*	M37	=	*P5 + P9 + P10*
M17	=	*P0 + P17 + P19*	M38	=	*P6 + P7*
M18	=	*P2 + P17 + P19*	M39	=	*P6 + P9*
M19	=	*P2 + 2P19*	M40	=	*P4 + P7*
M20	=	*P0 + 2P19*	M41	=	*P4 + P9*

Table 7. Reachability Set of PN in Fig. 7

5. Conclusion

We have used PN to model and formally analyze a communication system which holds a elliptic curve encryption scheme and it could works on a WSN. The results obtained shows formally that our approach are correct. However, we plan increase the complexity of models. We should consider critical factors related to constrained capacities of sensors nodes. We have analyze a more elaborate encryption scheme keep in mind adding a intermediate entity for the keys management.

6. References

Carrasco, R. A. & Johnston, M. (2008). *Non Binary Error Control for Wireless Communication and Data Storage*, John Wiley & Sons Ltd, United Kingdom.

Castiñeira, J. & Guy, P. (2006). *Essentials of Error Control Coding*, John Wiley & Sons Ltd, England.

Chen, S., Ge, Q., Shao, Q. & Zhu, Q. (2008). Modelling and performance analysis de wireless sensor network systems using petri nets, *International Technical Conference on Circuits/Systems, Computers and Communications* .

Gallager, R. G. (1962). Low-density parity-check codes, *IRE Transactions on Information Theory* (No. 1): 21–28.

Haines, R., Clemo, G. & Munro, A. (2007). Petri-nets for formal verification of mac protocols, *IET Software* **Vol. 1**: 39–47.

Hankerson, D., Menezes, A. & Vangtone, S. (2004). *Guide to Elliptic Curve Cryptography*, Springer Professional Computing.

Liu, B., Ren, F., Lin, C. & Jiang, X. (2008). Performance analysis of sleep scheduling schemes in sensor networks using stochastic petri net, *International Conference on Communications (ICC'08)* (No.): 4278 – 4283.

Liu, J., Ye, X., Zhang, J. & Li, J. (2008). Security verification of 802.11i 4-way handshake protocol, *International Conference on Communications (ICC'08)* **Vol. 1**: 1642–1647.

MacKay, D. J. C. (1990). Good error-correcting codes based on very sparse marices, *IEEE Transactions on Information Theory* (No. 2, pp. 399).

Menezes, A. J., Okamoto, T. & Vanstone, S. (1993). Reducing elliptic curve logarithms to logarithms in a finite field, *IEEE Transactions on Information Theory* (No. 5).

Ontiveros, B., I.Soto & Carrasco, R. (2006). Construction of an elliptic curve over finite fields to combine with convolutional code for cryptography, *Transactions on Computers Circuits, Devices and Systems* pp. 1642–1647.

Pengand, Y., Zhanting, Y. & Jizeng, W. (2007). Petri net model of session initiation protocol and its verification, *International Conference on Communications (ICC)* **Vol. 1**: 1861–1864.

Reisig, W. (1986). *Petri Nets. An introduction*, Springer-Verlag, Berlin Heidelberg.

Reisig, W. & Rozenberg, G. (1998). *Lectures on Petri Nets I: Basic Models*, Springer-Verlag, Berlin Heidelberg.

Reyhani, A. (2003). Efficient multiplication beyond optimal normal bases, *IEEE Transactions on Computers* **Vol. 52**(No. 4): 1642–1647.

Reyhani, A. (2006). Efficient algorithms and architectures for field multiplication using gaussian normal bases, *Transactions on Computers* **Vol. 55**(No. 1): 1642–1647.

Roch, S. & Starke., P. H. (1999). *INA: Integrate Net Analizer*, Humboldt-Universität zu Berlin, Berlin.

Petri Nets Applications
Edited by Pawel Pawlewski

ISBN 978-953-307-047-6
Hard cover, 752 pages
Publisher InTech
Published online 01, February, 2010
Published in print edition February, 2010

Petri Nets are graphical and mathematical tool used in many different science domains. Their characteristic features are the intuitive graphical modeling language and advanced formal analysis method. The concurrence of performed actions is the natural phenomenon due to which Petri Nets are perceived as mathematical tool for modeling concurrent systems. The nets whose model was extended with the time model can be applied in modeling real-time systems. Petri Nets were introduced in the doctoral dissertation by K.A. Petri, titled "„Kommunikation mit Automaten" and published in 1962 by University of Bonn. During more than 40 years of development of this theory, many different classes were formed and the scope of applications was extended. Depending on particular needs, the net definition was changed and adjusted to the considered problem. The unusual "flexibility" of this theory makes it possible to introduce all these modifications. Owing to varied currently known net classes, it is relatively easy to find a proper class for the specific application. The present monograph shows the whole spectrum of Petri Nets applications, from classic applications (to which the theory is specially dedicated) like computer science and control systems, through fault diagnosis, manufacturing, power systems, traffic systems, transport and down to Web applications. At the same time, the publication describes the diversity of investigations performed with use of Petri Nets in science centers all over the world.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Hugo Rodriguez, Ruben Carvajal, Beatriz Ontiveros, Ismael Soto and Rolando Carrasco (2010). Using Petri Net for Modeling and Analysis of a Encryption Scheme for Wireless Sensor Networks, Petri Nets Applications, Pawel Pawlewski (Ed.), ISBN: 978-953-307-047-6, InTech, Available from: http://www.intechopen.com/books/petri-nets-applications/using-petri-net-for-modeling-and-analysis-of-a-encryption-scheme-for-wireless-sensor-networks

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

www.intechopen.com