مجلة جامعة بابل / العلوم / المجلد 14/ العدد 4 : 2007

Petri Net based Cryptosystem

Alia Karim Abdul Hassan
University of Technology, Computer Science

Abstract

Every few years, computer security has to re-invite itself. New technologies and new applications bring new threats, and force use to invent new protection mechanisms. Cryptography became important when businesses started to build net worked computer systems. The expressive and representation power of Petri Nets render them ideally suited to the modeling of many complex event systems and to reveal important information about the structure and dynamic behavior of the modeled system. In this work Petri Nets is used to design a secret key cryptosystem to encrypt the secret message before send to the receiver then the receiver used it to decrypt received massage.

الخلاصة

أصبحت الاتصالات الرقمية جزء أساسي من البنية التحتية هذه الأيام, والكثير من التطبيقات معتمدة على الانترنيت وفي بعض الحالات تحتاج هذه التطبيقات ان تكون سرية وبالتالي أصبحت ضمان سرية البيانات مسألة أساسية. شبكات بتري(Petri Nets) لها القدرة لتمثيل ووصف العديد من الأنظمة وإبراز معلومات كثيرة عن هيكلة وسلوكية النظام الذي تمثله. في هذا البحث استخدمت شبكات بتري لتصميم نظام تشفير ذو مفتاح سري لتشفير رسالة سرية ويرسلها للمستلم ثم المستلم يستخدم نفس النظام لإعادة الرسالة السرية.

1. Introduction

Every few years, computer security has to re-invite itself. New technologies and new applications bring new threats, and force use to invent new protection mechanisms. Cryptography became important when businesses started to build net worked computer systems [JD01]. Since the beginnings of human communication, the desire to communicate in secrecy has existed. There have been many solutions to this problem, the most widely used and investigated being cryptography. Historically, sensitive information has been protected using encryption. Encryption uses powerful mathematics to map plaintext into an unreadable ciphrtext that is sent over a channel to the recipient. When a message is encrypted it is done so using a secret key. To decrypt a message, the secret key is used to reverse the process. For an eavesdropper to defeat the system he or she must acquire the secret key [Jer03]. Designing and managing real systems often require that relationships between organization choice and attained results be identified in order to decide on possible improvements of the system architecture and of the operational environment for achieving better performance [Bab01].

In this work Petri Net (PN) is used to design a cryptosystem to encrypt the secret message before send to receiver then receiver used it to decrypt received massage. PN is a tool for describing, studying and analyzing the flow of information and control in systems. It has a great modeling power to represent different kinds of system (hardware and software) and to reveal important information about the structure and the dynamic behavior of the modeled system [Akb95].

2. Cryptography

Secrecy is at the heart of cryptography. Encryption is a practical means to achieve information secrecy. Modern encryption techniques are mathematical transformations (algorithms. The message input to an encryption algorithm is conventionally called plaintext which may or may not be intelligible. For example, a plaintext message can be a random nonce or a ciphertext message. Therefore, plaintext and ciphertext are a pair of respective notions: the former refers to messages input to, and the latter, output from, an encryption algorithm. In order to restore information, an encryption transformation must be reversible and the reversing transformation is called decryption as shown in figure(1).

Conventionally, encryption and decryption algorithms are parameterized by cryptographic keys. An encryption algorithm and a decryption algorithm plus the description on the format of messages and keys form a cryptographic system or a cryptosystem. In a secret-key cryptosystems encryption and decryption use the same key. The principal who encrypts a message must share the encryption key with the principal who will be receiving and decrypting the encrypted message. Secret-key cryptosystems another name: symmetric cryptosystems. In a public-key cryptosystem, encryption and decryption use different keys; public-key cryptosystems another name: asymmetric cryptosystems [WM03].

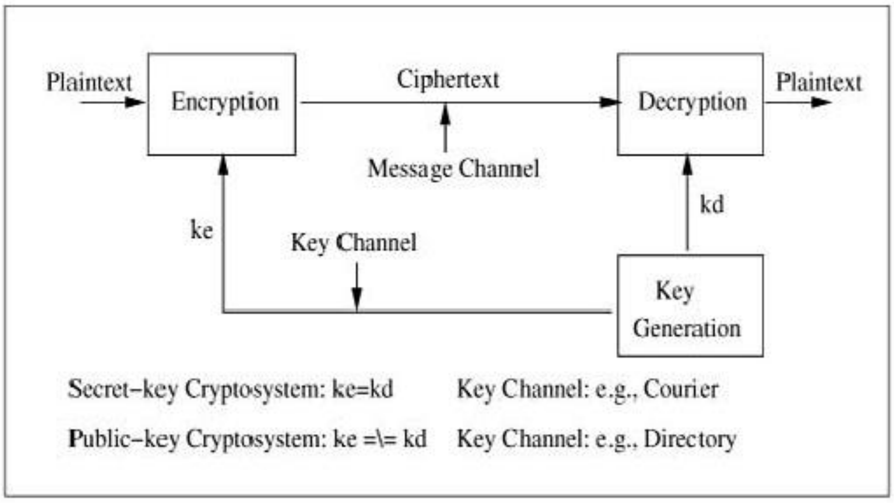

Plaintext
Encryption
Ciphertext
Decryption
Plaintext
Message Channel
ke
kd
Key Channel
Key Generation
Secret–key Cryptosystem: ke=kd
Key Channel: e.g., Courier
Public–key Cryptosystem: ke =\= kd
Key Channel: e.g., Directory

Figure (1): cryptography system

3. Petri Nets

Carl Adam Petri in the beginning of the 1960s presented the foundation of PN. Petri wanted to define a general-purpose graphical and mathematical model describing relations between conditions and events [Joh99]. The expressive and representation power of PN render them ideally suited to the modeling of many complex event systems and to reveal important information about the structure and dynamic behavior of the modeled system. This information can then be used to evaluate the modeled system and suggest improvements or changes. Yet, they are quite powerful and possess many advantages in modeling a system [Bey02]. Structure modeling is accomplished through the graphical structure and the input-output relationships among the places and transitions while behavior modeling is achieved through execution of firing rules that capture the dynamics of the modeled system over time [Nar99].

3.1 Basic Concepts and Definitions

PN is a particular kind of directed graph, together with an initial state marking, M_0. The underlying graph *N* of a PN is directed, weighted, bipartite graph consisting of two kinds of nodes, called *places* and *transitions*, where arcs are either from a place to a transition or from a transition to a place [Mur89]. Graphical representations of

PNs are as follows: places, transitions, connections, and tokens are represented by circles "O", bars " |", arcs"→", and bullets "•", respectively [KK00].

Definition1: Formal Definition of PN:

A unmarked PN is a 4-tuple $N= \{P, T, I, O\}$ Where

$P= \{p_1, p_2,...,p_n\}$ is a finite, nonempty set of places.

$T= [t_1, t_2,..., t_m]$ is a finite, nonempty set of transitions,

$P \cap T \neq Ø$ i.e., the sets P and T are disjoint.

$I:P\times T\rightarrow \{0,1\}$ is the input incidence function.

$O:P\times T\rightarrow \{0,1\}$ is the output incidence function.

A marked PN is a pair PN= (N, M_0) in which N is an unmarked N and M_0 is the initial marking [Joh99].

Definition2: (Marking-Tokens-Initial marking)

A marking *M* is a mapping *M: P→ K* that associates with each place a number of tokens [kno02].A marking (state) assigns to each place a nonnegative integer. If a marking assigns to place *P* a nonnegative integer *K*, we say that *P* is marked with k tokens. Pictorially, we place *k* black dots (tokens) in place *p*. A marking is denoted by *M*, and m-vector, where m is the total number of places. The *p*th component of M, denoting by *M (p)*, is the number of tokens in place *p* [Mur89].

Transition Enabling and Firing

The behavior of many systems can be described in terms of system states and their changes. In order to simulate the dynamic behavior of a system, a state or making in Petri nets is changes according to the following transition (firing) rules:

Definition 3: (Enabling Rule)

A transition $t_j \in T$ is enabled if all input places of t_j contain a token and all output places are empty, $M (p_i) =1 \ \forall \ p_i \ |(p_i,t_j) \in F$ and $M(p_k)= 0 \ \ \forall \ p_k|(t_j,p_k) \in F$.

Definition 4: (Firing Rule)

An enable transition may fire. On firing it removes the tokens from all its input places and places one token in each of its output palaces [Mur89].

3.2 Structural Properties of PN

Here we identify PN constructs for representing characteristics of various features [Nar99]:

Sequential Execution: In figure (2 a) , transition t_2 can fire only after the firing of t_1. This imposes the precedence constraint ' t_2 after t_1' .

Concurrency: In figure (2 b) , the transitions t_1,t_2, and t_3 are concurrent. Concurrency is an important attribute of system interactions.

Synchronization: as shown in figure (2 c) t_1 will be enabled only when each of its input places has a minimum appropriate number of tokens, so in figure (2 c) , p_1,p_2 have tokens but p_3 does not have a token. Therefore, for t_1 to be enabled, we will have to wait for a token to arrive into p_3.

3.3 Behavioral Properties of PN [Nar99]:

Definition 5: (Boundedness)

A place pi of a marked Petri net is said to be k-bounded (k>0, k integer) in marking M_0 if

$M (p_i) \leq k \quad \forall \ M\in R[M_0]$

If k=1, p_i is said to be safe. If all places of PN are k-bounded (safe) in a marking Mo, the PN itself is said to be bounded (safe) in M_0.

Definition 7: Reversible (or proper)

A marked PN with the initial marking $M_0$is said to be proper or reversible if

$M_0\in R [M_0] \quad \forall \ M\in R [M_0]$

In a proper PN, the initial marking is reachable in one or more steps from every reachable marking. Properness implies reinitializability of the system and assumes significance in ensuring recovery from failure states.

Definition 8: Liveness

A transition t_j of a marked PN is said to be alive under a marking M_0 if, for all markings $M_0 \in R\ [M_0]$, there exists a sequence of a transition firings which result in a marking that enable t_j. A PN is said to be live if all its transitions are live. Liveness of a PN implies absence of deadlocked states.

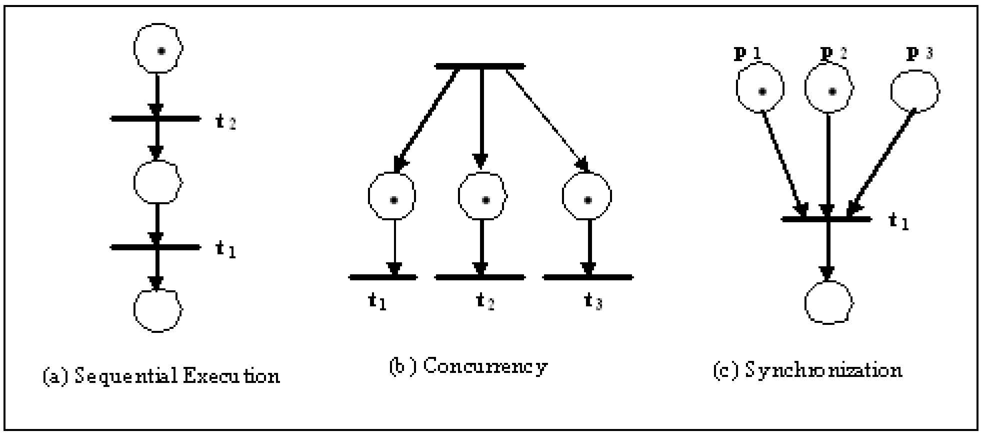


Figure (2): Some of PN properties

4. Proposed PN based Cryptosystem

In our cryptosystem a definition of the PN used in encryption stage is:

Let P represents a set of all places, T represents a set of all transitions, and B will be used to denote an 8-bit vector which represents a token. M denoted the set of all markings. The input function is I: $P\times T\rightarrow B$ and output function is O: $T\times P\rightarrow B$, then PN= (P, T, I, O) and $M_0 \in M$ is an initial marking.

Figure (3) represents the PN-model of encryption model; with an initial state, that the first place for this model is the 8-bits binary representation of a full ASCII code for a character while the final place is the new 8-bits binary representation of a full new ASCII code, and the formal description of the suggested PN-model is as follow:

$P=(p_1,p_2,p_3,\ldots,p_{14})$ and $T=(t_1,t_2,t_3,\ldots,t_{11})$

$I(t_1) = p_1,\ I(t_2) = p_2$	$O(t_1) = \{p_2,p_3,p_4\}$
$I(t_3) = p_3,\ I(t_4) = p_4$	$O(t_2) = p_5,\ O(t_3) = p_6$
$I(t_5) = p_5,\ I(t_6) = p_6$	$O(t_4) = p_7,\ O(t_5) = p_8$
$I(t_7) = p_7,\ I(t_8) = p_8$	$O(t_6) = p_9,\ O(t_7) = p_{10}$
$I(t_9) = p_9,\ I(t_{10}) = p_{10}$	$O(t_8) = p_{11},\ O(t_9) = p_{12}$
$I(t_{11}) = \{p_{11}, p_{12}, p_{13}\}$	$O(t_{10}) = p_{13},\ O(t_{11}) = p_{14}$

The PN-model of encryption executed in the following way:

The firing of t_1 causes dividing the 8-bits binary representation of a full ASCII code for a character into three parts as shown in the figure (4). The reason of division of the input code is that (BS1, BS2, and BS3) will be processed concurrently through transitions t_2, t_3, t_4, this mean each part is processed completely independent of other parts that will add more complexity to the encoding process, and this condition represents the concurrency structure property of the proposed PN-model for encryption.

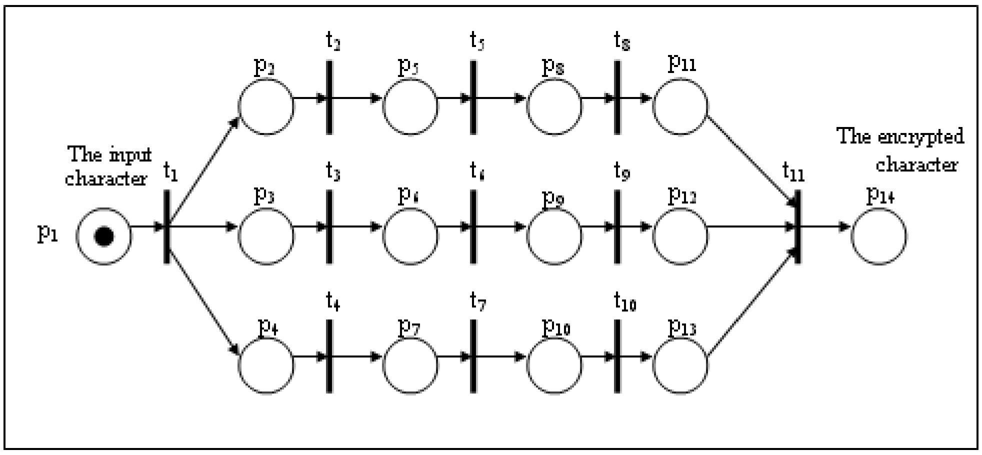


Figure (3) : Suggested PN-model for encryption method

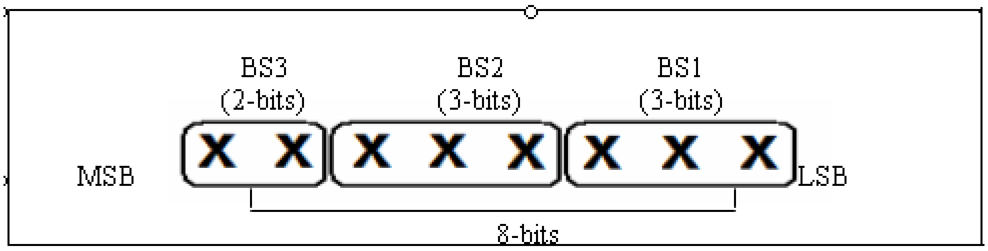


Figure (4) : The three part of 8-bits binary representation of a full ASCII code character

In proposed PN-model, three different indexed sequences of BCD of three bits as shown in (table (1), table (2), and table (3)) are generated and used in transitions t_2, t_5, t_8 and , three different indexed sequences of BCD of three bits as shown in (table (4), table (5), and table (6)) are generated and used in transitions t_3, t_6, t_9 and three different indexed sequences of BCD of two bits are generated and indexed and used in transitions t_4, t_7, t_{10} as follows in tables(table 7,table 8, and table (9)).

In each transition of PN-model, different sequences of BCD are used**,** the reason for that is the final results of encrypted code will be not sequential(non linear encryption) and also changing one sequence in each BCD sequence will changing all results of final code that will be useful to modify the method.
Just after firing t_8, t_9, t_{10} a new (BS1, BS2, and BS3) are generated and make transition t_{11} enabled, after t_{11} fired and output a token p_{14} a new ASCII code of the input character is generated. This condition represents the synchronization structure property of the proposed PN-model. Algorithm (1) illustrates how the proposed encryption model works.

Table (1) BCD sequence in t_2

INDEX	BCD
0	010
1	100
2	111
3	101
4	001
5	000
6	011
7	110

Table (2) BCD sequence in t_5

INDEX	BCD
0	110
1	101
2	001
3	100
4	111
5	000
6	011
7	010

Table (3) BCD sequence in t_8

INDEX	BCD
0	010
1	110
2	111
3	100
4	001
5	000
6	011
7	101

Table (4) BCD sequence in t_3

INDEX	BCD
0	010
1	110
2	111
3	011
4	001
5	000
6	100
7	101

Table (5) BCD sequence in t_6

INDEX	BCD
0	111
1	010
2	101
3	011
4	110
5	100
6	001
7	000

Table (6) BCD sequence in t_9

INDEX	BCD
0	101
1	100
2	000
3	011
4	001
5	010
6	111
7	110

Table (7) BCD sequence in t_4

INDEX	BCD
0	10
1	00
2	01
3	11

Table (8) BCD sequence in t_7

INDEX	BCD
0	01
1	11
2	00
3	10

Table (9) BCD sequence in t_{10}

INDEX	BCD
0	00
1	10
2	01
3	11

Algorithm (1): Encryption Method
Input: secret message.
Output: the encrypted secret message.
Step 1: Generate 6 different BCD sequences of 3 bits and another 3 different sequences of 2 bits and documented these sequences by sender to use it in the reception station.
Step 2: Read a single character from the secret message.
Step 3: Obtain ASCII code for the read character and convert it into 8-bit binary representation, this means reaching a single token in place p_1 that represents the initial state of PN-model.
Step 4: The token in place p_1 make the transition t_1 enabled by firing t_1 the (8-bits) code resulted from step 2 is divided into three groups two of 3 bits and of 2

bits, then a token is placed in each place (p_2,p_3,p_4) and make transitions t_2,t_3,t_4 enabled.

Step 5: By firing t_2, t3, t4 the following process is work in each of these transitions to produce a token in each p_5, p6, p_7:

Search on the value of (BS1) in the BCD sequence as shown in table (1), and search the value of (BS2) in the BCD sequence as shown in table (4) and take out the index of this value in this sequence then represent it as a value, and search the value of (BS3) in the BCD sequence as shown in table (7) and take out the index of this value in this sequence then represent it as a value.

Repeat the previous search process in transitions t_5,t_6,t_7,t_8,t_9,t_{10} and used its BCD sequences as shown in tables(2), (5), (8), (3), (6), and (9).

*Step 6:*The tokens which output from firing t_8,t_9,t_{10} are represent the new value for(BS1, BS2 & BS3) ,these tokens make transition t_{11} enabled ,then by firing t_{11} the following process is work to output a token in p_{14}:

eliminate the five most significant bits from (BS1, BS2) and six most significant bits from (BS3) and concatenating the remaining bits of the new three values with the same order to generate new (8-bits), the resulted value represents the encrypted value of a character reads in step 2.

Step 8: If the characters of the secret message are ended then continue to step 9 else return to step 2.

Step 9: Save the resulted ASCII values in matrix (r) of size (64×64) and finish.

4.1 Example: Execution of PN-model for encryption

In this example a character (A) will be encrypted using the proposed method. First convert the ASCII code of character (A) into 8-bit binary representation (01000001), which considered the input place to the proposed PN-model, the PN-model is executed as follow:

Initial marking: M_0=((01000001),0,0,0,0,0,0,0,0,0,0,0,0,0)

In this current state t_1 is enabled, firing t_1 changing the marking of PN-model to M_1:

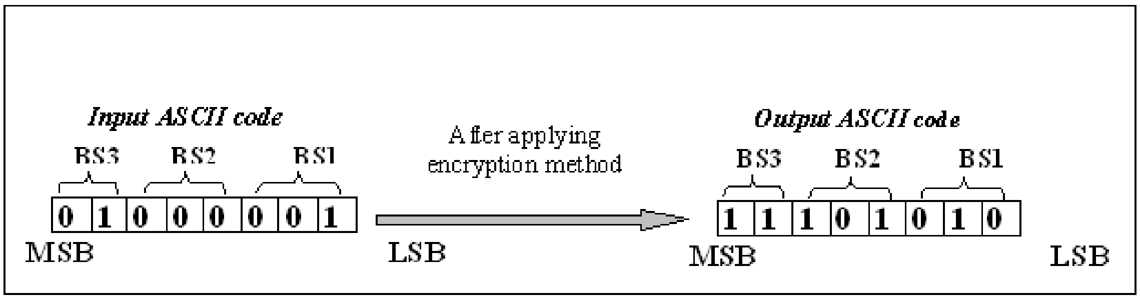
M_1=(0,(00000001),(00000000),(00000001),0,0,0,0,0,0,0,0,0,0)

In the current state t_2, t3, t4 are enabled, firing these transitions changed the marking to:

M_2=(0,0,0,0,(00000100),(00000101),(00000011),0,0,0,0,0,0,0)

PN-model Execution continued as shown in table (10) shows the markings that can obtain from initial marking and gives every possible sequence of transition firing.

Firing the transition t_{11} will produce the place p_{14} which represents the encrypted 8-bit representation of the input ASCII code; figure (5) shows a character (A) before and after the execution of PN-model.



Figure(5): Shows the ASCII code of a character (A) before and after the execution of PN-model.

Table (10) PN-model Execution

CURRENT MARKING	ENABLED TRANSITION	NEW MARKING
M_0=((01000001),0,0,0,0,0,0,0,0,0,0,0,0,0)	t_1	M_1=(0,(00000001),(00000000),(00000001),0,0,0,0,0,0,0,0,0,0))
M_1	t_2,t_3,t_4	M_2=(0,0,0,0,(00000100),(00000101),(00000011),0,0,0,0,0,0,0))
M_2	t_5,t_6,t_7	M_3=(0,0,0,0,0,0,0,(00000011),(00000110),(00000011),0,0,0,0))
M_3	t_8,t_9,t_{10}	M_4=(0,0,0,0,0,0,0,0,0,0,(00000110),(00000101),(00000011),0))
M_4	t_{11}	M_5=(0,0,0,0,0,0,0,0,0,0,0,0,0,(11101110))

4.2 Decryption Process

In this stage the secret message is decrypted using the inverse operations followed in the algorithm (1). This stage represents the second and final step in the reception station of restoring the secret message.

The proposed PN model in encryption stage must be documented by sender to use it the same as in the reception station. PN-model represent the secret key in our proposed cryptosystem. The receiver use PN-model the same which used in sending station except the order of BCD sequences must be changed for transitions where the , three different indexed sequences of BCD of three bits that shown in (table (1), table (2), and table (3)) are used in transitions t_3, t_6, t_9 and , the three different indexed sequences of BCD of three bits that shown in (table (4), table (5), and table (6)) are generated and used in transitions t_2, t_5, t_8. The Input place of PN-model is the extracted 8-bit binary representation of ASCII code for a character and the output place is 8-bit binary representation of original ASCII code for input the character. Figure (6) presents the PN-model of decryption method. Algorithm (2) illustrates how the proposed decryption method works:

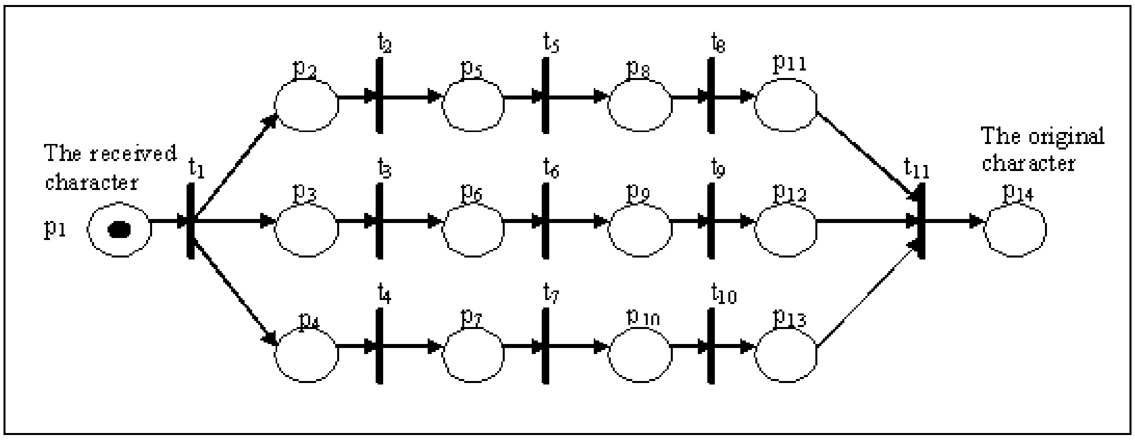


Figure (6): PN-model for decryption the secret message

Algorithm (2): Decryption Process
Input: Encrypted secret message ASCII codes (matrix(r)).
Output: The secret message.

```
Step 1: Generate the same six BCD sequences
Step 2: Read ASCII code value of a character from matrix(r).
Step 3: Convert ASCII to 8-bit binary representation.
Step 4: Divide (8 bits) code resulted from step 3 into three groups two of 3 bits and
        one of 2 bits .
Step 5: Search on the value of (BS1) in the first sequences from the first BCD
        sequence and take out the index of this value in this sequence then represent
        it as a value and repeat the previous search operation in the second and third
        BCD sequences from the same stage.
Step 6: Repeat step 5 for the value of (BS2) for the sequences tables(4, 5 &6) in the
        second stage also for the value of (BS3) for the sequences tables(1, 2 &3) in
        the third stage.
Step 7: The output of step 5 & 6 is the encrypted value for (BS1, BS2 & BS3),
        eliminate the five most significant bits from (BS1, BS2) and six most
        significant bits from (BS3) and concatenating the remaining bits of the new
        three values with the same order to generate (8-bits), the resulted value
        represents the original ASCII code of readied character.
Step 8: If the characters of the secret message are ended then continue to step 9 else
        return to step 2.
Step 9: Covert ASCII values to characters then save the final text message in new file
        and finish.
```

4.2 Example

A complete real example is implemented here. All the operations needed are discussed in algorithms (1 and 2) but here these algorithms are implemented typically on real secret message. The secret message used in this example is listed in Figure(7). In algorithm 1, each character in the secret message is converted to its ASCII value as shown in table (11), then after applying algorithm 1 a matrix (r) of size (64×64) pixel (that represents the encryption ASCII values of the secret message's characters) is obtained as shown in table (12) while the corresponding message showed in figure(8) .This matrix represents the encryption ASCII values. Note that in this example the message selected contains (660) characters and the reminder element of matrix can be set to (000) and after encryption the value (000) is encrypted to (044).

Algorithm (2) is applied to decrypt the ASCII code of (table (12)) secret message's characters; finally the character corresponding to each ASCII code is taken and gets the original secret message which is transmitted by sender. The resultant message is completely the same as the message listed in Figure (7) , so we retrieve the secret message without loss.

Table 11 ASCII values for the characters of secret message in example

	1	*2*	*3*	*4*	*5*	*Rows*	*63*	*64*
1	**083**	**116**	**101**	**120**	**103**		**111**	**110**
2	**099**	**101**	**097**	**108**	**032**		**114**	**101**
3	**118**	**101**	**110**	**116**	**032**	………………………	**111**	**108**
4	**118**	**101**	**115**	**032**	**118**		**101**	**032**
5	**111**	**116**	**104**	**101**	**114**		**032**	**097**
. . *Columns* . .	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	………………………	. . . . .	. . . . .
63	**000**	**000**	**000**	**000**	**000**		**000**	**000**
64	**000**	**000**	**000**	**000**	**000**		**000**	**000**

Steganography also differs from cryptography, which does not conceal the communication itself but only scrambles the data to prevent eavesdroppers understanding the content. Cryptography involves various methods and implementations. Steganography, on the other hand, is a relatively new area of study, as reflected in the research focus of published papers. Steganography and cryptography may be considered complementary and orthogonal. Anyone engaging in secret communication can always apply a cryptographic algorithm to the data before embedding it to achieve additional security. In any case, once the presence of hidden information is revealed or even suspected, the purpose of steganography is defeated, even if the message content is not extracted or deciphered.

Figure(7) secret message

Table (12) ASCII values for the characters of encrypted secret message in example

	1	*2*	*3*	*4*	*5*	*Rows*	*63*	*64*
1	**245**	**248**	**201**	**202**	**206**		**210**	**211**
2	**201**	**201**	**206**	**218**	**012**	………………	**255**	**201**
3	**251**	**201**	**211**	**248**	**012**	………	**210**	**208**
4	**251**	**201**	**253**	**012**	**251**		**012**	**210**
5	**248**	**212**	**201**	**212**	**201**		**210**	**120**
. *Columns* . .	. . .	. . .	. . .	. . .	. . .	……………… ………	. .	. .
63	**044**	**044**	**044**	**044**	**044**		**044**	**044**
64	**044**	**044**	**044**	**044**	**044**		**044**	**044**

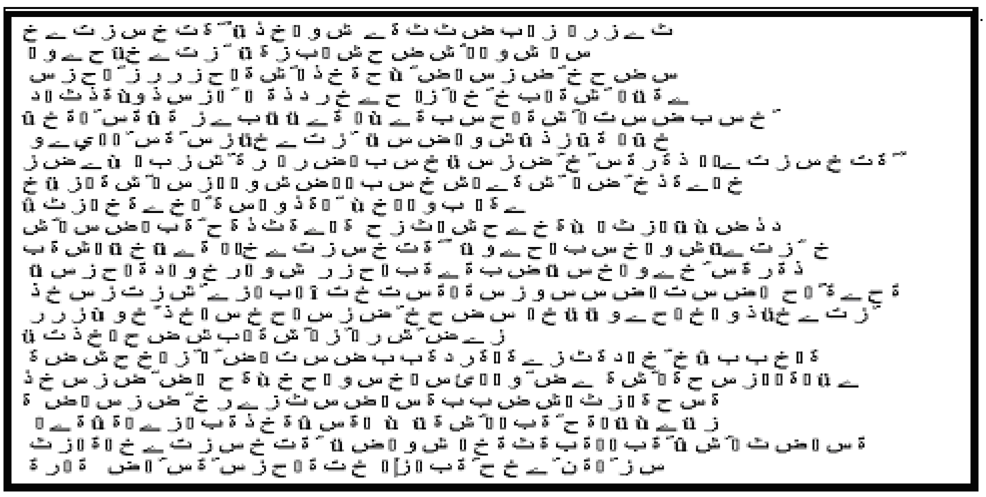


Figure 8 encrypted message

5. Conclusions

The proposed cryptosystem performs new approach for data encryption by using PN models. The following conclusions can be derived from this work:

1. The proposed cryptosystem can be defined as a secret key cryptosystem where the same key has been used by the sender and recipient.
2. PN model represent the secret key of the cryptosystem which must be documented between them.
3. By using PN, the proposed encryption method is reliable, verified and analyzed because the PN itself provides a formal method to analyze and verify models.
4. The division of the input code to three parts (BS1, BS2, and BS3) add more complexity to the encryption process.
5. Each transition of PN-model, different sequences of BCD are used**,** this made the final results of encrypted code will be not sequential (non linear encryption).
6. Changing one sequence in each BCD sequence will changing all results of final code that will be useful to modify the method.

References

[Akb95] Akbas I. A. " **Modeling Logic Programming Using Petri Nets**", Ph.D. thesis, Department of computer Science, University of Technology, Iraq, 1995.

[Bab01] Bablo G., "***Introduction to Stochastic Petri Net***", (Eds.): FMP2000, LNCS 2090, pp. 84-115. Springer-Verlag Berlin Heidelberg 2001.

[Bey02] Beythoon R., "***Automatic Synthesis of Petri Nets using Genetic programming***", Ph.D. thesis, Department of computer Science, University of Technology, Iraq, 2002.

[JD01]Johnson N. F., Duricn Z., Jajodia S., "***Information hiding: Steganography and watermarking attack and countermeasures***", kluwer Academic publishers, USA, 2001.

[Jer03] Jeremiah J. H., "***Steganalysis Of Additive Noise Modelable Information Hiding*"**, M. SC., Thesis**,** Rensselaer Polytechnic Institute, 2003.

[Joh99] Johnsson C., "***A Graphical Language for Batch Control***", Ph. D. thesis, Department of automatic Control, Lund Institute of Technology, Lund University, 1999.

[KK00] Keigo K., Kengo I., and Toshimistu U., " ***LLP Control with Timed Petri Net Models in Mobile Robots***", Department of system and human Science, Osaka University 1-3 Machaneyam, Toyonaka, Osaka 560-8531, Japan, , URL: http://www.kklabb.ces.kyutech.ac.jp/~kobayasi/smc01.pdf.

[kno02] Knorr K., "***Multilevel Security and Information Flow in Petri Net Workflows***", Department of Information Technology, University of Zurich, 2002, URL: http://www.ifi.unizh.ch/~knorr/documents/p_icots2001.pdf

[Mur89] Murata T., "***Petri Net: Properties, Analysis, and Application***", Pro. IEEE, vol.77, pp.541-579, April 1989.

[Nar99] Narahari Y., "***Petri Nets: Overview and foundations***", Resonance, pp 58-69,August 1999.

[WM03] Wenbo, " ***Modern Cryptography: Theory and Practice"***, Prentice Hall PTR, Mao Hewlett-Packard Company, 2003.