International Journal of Bioinformatics and Biomedical Engineering
Vol. 2, No. 1, 2016, pp. 19-24
http://www.aiscience.org/journal/ijbbe
ISSN: 2381-7399 (Print); ISSN: 2381-7402 (Online)

Application of Colored Petri Nets for Modeling of Biometric Identification Protocol with Random Parameter

G. R. Petrosyan*, **L. A. Ter-Vardanyan**

Institute for Informatics and Automation Problems of NAS RA, International Scientific - Educational Centre of NAS RA, Yerevan, Armenia

Abstract

Biometric Identification Protocol with Random Parameter is modelled in the current paper by using Petri Nets, namely with the modern extension of Petri Nets-Colored Petri Nets. The purpose of modelling Biometric Identification Protocol with Random Parameter by the use of Petri Nets is to reveal the following aspects of the functioning model: The efficiency of the model; Behaviour; The existence of mistakes and accidents; Whether it is possible to simplify the system or substitute its separate components for more perfect components not breaking the general functioning of the system. The use of mathematical markings introduced in the paper shows that Colored Petri Net modeling the process of Biometric Identification Protocol with Random Parameter is identified with the Colored Petri Net modeling the process of Biometric Identification System depicted in [12]. In this paper the model of Colored Petri Net describes the identification process with Random Parameter much simpler.

Keywords

Petri Net (PN), Colored Petri Net (CPN), Position, Transition, Arc, Biometric Identification Protocol with Random Parameter, Channel with Random Parameter

Received: September 17, 2015 / Accepted: November 12, 2015 / Published online: January 8, 2016

1. Introduction

In modern society informational communication and confidential usage of it, became a common matter and is quite used by many. One of the important areas of application are biometrics and Petri Net theory. The core tasks of biometric security are person's reliable identification and authentication according to his biometric data. The main problem of Petri Net is modelling of real-time systems by optimization of the processes with clear description. Therefore, Petri Net computer system gives opportunities to study properties, use them to solve practical problems, mainly those issues that are related to information processing models, paralleled with sources and consider to be important issue [1, 2, 3, 4]. Till now, it has been shown that the task has fundamental interpretation of information theory [5, 7, 8, 9, 10, 11]. According to the theory, it is impossible to reliably identify many individuals but specified number, which is the main feature of identification system.

In such cases we need to use Petri Net. The following problems may serve as examples for problems that often occur in discrete systems in need of design and research:

- The system uses the functions for which it was intended.
- Whether it operates effectively.
- There might occur some errors and emergency situations.
- Whether there are potential barriers and
- If it is possible to simplify the system or replace individual components with more perfect components, without prejudice to its general functions.
- If it's possible to design more complicated and more

* Corresponding author
E-mail address: petrosyan_gohar@list.ru (G. R. Petrosyan), lilit@sci.am (L. A. Ter-Vardanyan)

functional systems that will meet the requirements and so on.

- These problems mainly have a qualitative, rather than quantitative nature.

So far, with the help of Petri Net various extensions, Biometric Identification Systems have not been simulated.

1.1. Petri Net

Definition. Petri Net $M(C,\mu)$ pair, where $C=(P,T,I,O)$ is the network structure and μ is the network condition. In structure C of a P-positions, T-transitions are finite sets. $I:T\to P^{\infty}, O:T\to P^{\infty}$ are the input and output functions, respectively, where P^{∞} are all possible collections (repetitive elements) of P. $\mu:P\to N_0$ is the function of condition, where $N_0=\{0,1,...\}$ is the set of integers. We determine (in a known manner) the allowed transitions of Petri Nets and the transitions from one state to another, as well the set of reachable states.

Places, transitions, tokens and arcs are the basic Petri Net components. A Petri Net can be thought of as a bipartite graph consisting of two types of nodes, places and transitions. Places are displayed pictorially as circles (or ovals) and transitions are displayed as vertical lines. Shifts of tokens are define the state of the system.

The simple presentation of system by Petri Net is based on two conceptions-events and conditions.

Event is an action which occurs in the system. The appearance of the events governs the state of the system. The state of the system can be described by multiple conditions.

The condition is a predicate or a logical description of the states of the system. The condition can admit true or false value. As the events are actions they can take place. In order the event occurs, is necessary appropriate existence of the conditions. They call these conditions the preconditions of the event. The appearance of the event can bring the a break of the preconditions and bring to other carrying of the condition-postconditions.

As an example of Petri Net let's consider the simple problem of modelling of the automat-seller. The automat-seller is in the waiting mode until the order hasn't appeared, which it makes and sends to delivery. The conditions of the mentioned system are the following:

- P_1-the automat-seller is waiting
- P_2-the order is received and it is waiting
- P_3-the automat-seller is making the order
- P_4-the order is completed.

The events of the system are the following:

- T_1-the order is admitted
- T_2-the automat-seller begins making the order
- T_3- the automat-seller completes making the order
- T_4-the order is sent to delivery.

Such presentation of the system is easy to model by Petri Net. The conditions in Petri Net are modeling by *positions* and events are modeling by *transitions.* Moreover, the entrances of transition are preconditions of appropriate events and exits are post conditions. The appearance of the event is equal to appropriate transition of the run.

Petri Nets are appropriate to Stack Automata. Is therefore sometimes the admissibility of transition is compared with the presence of bullets in a Stack Automata. Therefore, expression of the transition commission mentioned in the literature as "A firing of an enabled transition".

The committing of the transition is presented by the existence of token positions according to that condition. The committing of the transition misses the allowed tokens, which present the committing of the preconditions and form new tokens, which present the committing of the post conditions.

The Petri Net is showed in the figure 1, which simulates the model of automat-seller.

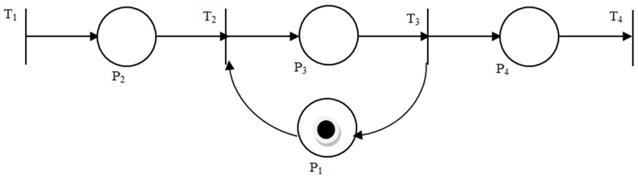


Fig. 1. The Petri Net which simulates the model of automat-seller.

Suppose T_1 transition has fired (the order is admitted), a token appears in P_2 position that is the order is received and it is waiting, now T_2 transition is allowed, as there are tokens in its corresponding incoming two positions. In the result of firing of T_2 transition, appears a token in P_3 position (the automat-seller is making the order), therefore T_3 transition is allowed, so the firing of the ordinal order is completed and again a token appears in P_1 position, so the automat-seller is waiting for the next order. Another token also appears in P_4 position, which means that the previous order is completed, therefore in the result of firing T_4 transition the order is sent to delivery.

The usage of places and transitions depends on the system being modelled. Places could represent resource status or operations. Arcs often represent the flow of data or resources. Transitions could represent the start/finish of processes. In terms of simulations, transitions can be used to model both *activities* and *events*. Activities can be thought of as the processes and logic of the system, while events occur at a single point in time and cause a change in the state of a system (White and Ingalls 2009). In fact, a transition may act as a super-process consisting of many sub-processes. In this way hierarchical nets come into play.

1.2. Colored Petri Net

Colored Petri Net (CPN) is Classical Petri Net modern expansion which was created by K. Jensen [3]. Coloured Petri Nets (CP-nets or CPNs) is a modelling language developed for systems in which communication, synchronisation and resource sharing play an important role. CP-nets combine the strengths of Classical Petri Net s with the strengths of a high-level programming language. Petri Nets provide the primitives for process interaction, while the programming language provides the primitives for the definition of data types and the manipulations of data values.

CP-nets has an intuitive, graphical representation which is appealing to human beings. A CPN model consists of a set of modules (pages) which each contains a network of places, transitions and arcs. The modules interact with each other through a set of well-defined interfaces, in a similar way as known from many modern programming languages. The graphical representation makes it easy to see the basic structure of a complex CPN model, i.e., understand how the individual processes interact with each other.

CP-nets also has a formal, mathematical representation with a well-defined syntax and semantics. This representation is the foundation for the definition of the different behavioural properties and the analysis methods. Without the mathematical representation it would have been totally impossible to develop a sound and powerful CPN language. However, for the practical use of CP-nets and their tools, it suffices to have an intuitive understanding of the syntax and semantics. The programming languages are successfully applied for those who are not familiar with the formal, mathematical definitions of the languages.

CPN models can be made with or without explicit reference to time. Untimed CPN models are usually used to validate the functional/logical correctness of a system, while timed CPN models are used to evaluate the performance of the system. There are also other languages which can be used to investigate the functional/logical correctness of a system or the performance of it. However, it is rather seldom to find modelling languages that are well-suited for both kinds of analysis.

CP-nets can also be simulated interactively or automatically. In an interactive simulation the user is in control and we can see the effects of the individual steps directly on the graphical representation of the CP-net. In this case the user can investigate the different states and choose between the enabled transitions. An interactive simulation is similar to single-step debugging. It gives a way to "walk through" a CPN model, investigating different scenarios and checking whether the model works as expected. This is different compared with many off-the-shelf simulation packages which often act as black boxes, where the user can define inputs and inspect the results, but otherwise have very little possibility to understand and validate the models on which the simulations build. The insight and detailed knowledge of a system, which the users gain during the development and validation of a simulation model, is often as important as the results that the users get from the actual simulation runs.

It is necessary to execute the CPN models as fast and efficient as possible, without detailed human interaction and inspection. The user still needs to interpret the simulation results. According this, it is often suitable to use animated, graphical representations providing an abstract, application-specific view of the current state and activities in the system.

CP-nets also gives more formal verification methods, known as state space analysis and invariant analysis. So we can prove the mathematical sense of the word, that a system has a certain set of behavioural properties. Also, industrial systems are often so complex that it is impossible or at least very expensive to make a full proof of system correctness. Hence, the formal verification methods should be seen as a complement to the more informal validation by means of simulation. The use of formal verification is often restricted to the most important subsystems or the most important aspects of a complex system.

CP-nets and their tools have been used in various practical projects within a large variety of different application areas. The CPN group at the University of Aarhus, Denmark, has developed two sets of computer tools, supporting the use of CP-nets:

- Design/CPN was developed in the late 80'ies and early 90'ies. It has been used by 800 different organisations in 60 different countries.
- CPN Tools is the next generation of tool support for Coloured Petri Nets. Today it is used by 700 organisations and individuals in 70 different countries.

Formal definition for Colored Petri Net is presented in [3, 4, 12].

Colored Petri Net is graphical oriented language which is used for modelling, analysis, description and presentation systems [3, 4, 6].

In the classical or traditional Petri Net tokens do not differ from each other, we can say that they are colourless. In difference of Classical Petri Net, in Colored Petri Net of a position can contain tokens of arbitrary complexity -a note, lists, etc., that makes reliable models more possible.

The Design/CPN system is used for modelling "NOKIA" phones, in order to find out the unwanted functional interactions [3, 4]:

If briefly describe the scope of the Colored Petri Nets, we can say:

CPNs have a graphical representation, are very general and can be used to describe a large variety of different systems, CPNs have an explicit description of both states and actions, offer hierarchical descriptions, interactive simulations where the results are presented directly on the CPN diagram, CPNs have computer tools supporting their drawing, simulation and formal analysis [1, 2, 3, 4, 6].

Colored Petri Net consist of the following components:

- The ellipses and the circles are called places. They describe the states of the system (buffers).
- The rectangles are called transitions. They describe the actions (processes).
- The arrows are called arcs. The arc expressions describe how the state of the CPN changes when the transitions occur.
- Each place contains a set of markers called tokens, each of these tokens carries a data value, which belongs to a given type.

Consider the simple example of a Colored Petri Nets, which is shown in Figure 2.

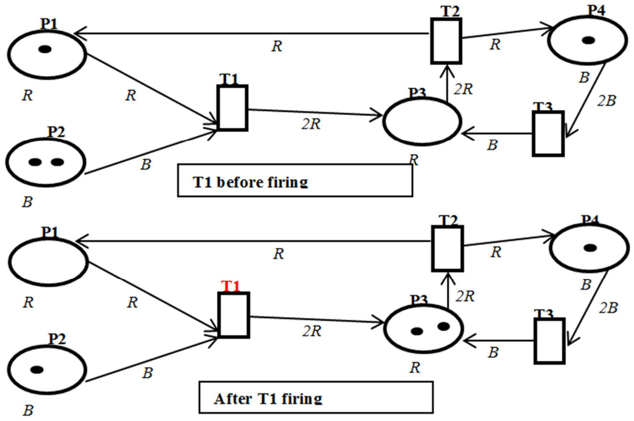


Fig. 2. Example of CPN transition.

P1 and P3 positions include tokens of R type in Fig. 2, and P2, P4 positions include tokens of B type. After T1 transition firing transfers a token of R type by (P1, T1) arc, and transfers a token of B type by (P2, T1) arc (that is, unlike Classical Petri Nets the tokens aren't homogeneous). Two tokens of R type are transfer by (T1, P3) arc, and there appear two tokens in P3. Before T1 transition firing, the state of the system is described by vector S1=(1,2,0,1), and after T1 transition firing will be S2=(0, 1, 2, 1). Therefore, in the result of permissible transitions of firing we will have the set of reachable states of the system.

2. The Model of Biometric Identification Protocol with Random Parameter

Biometrics is often used by different companies, governmental systems, military forces, border control, hospitals, banks etc. to either verify a person's identity, for physical access control, computer log-in, welfare disbursement, international border crossing and national ID cards, e-passports, allowing access to certain building area or to identify individuals to retain information about them, i.e. criminals, forensics, etc. In automobiles, biometrics is being instaled to replace keys for keyless entry and keyless ignition [7].

The objective of a biometrical identification system is to identify individuals on the basis of physical (passive or active) features. One of the oldest and probably best known of such features is the human fingerprint. One can safely say that for a long time fingerprinting-based identification and biometrical identification have been seen as one and the same thing.

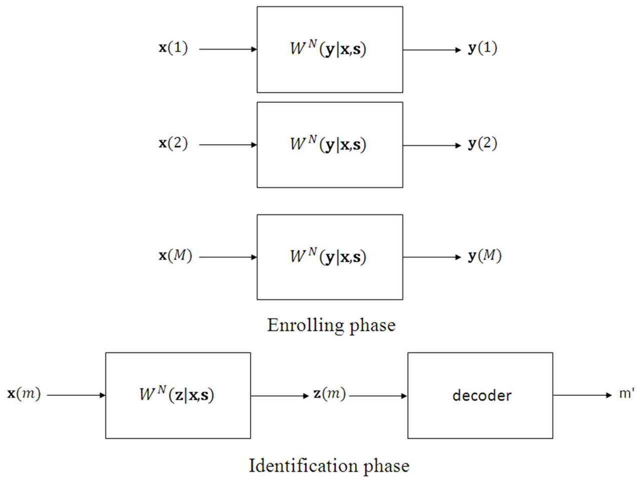


Fig. 3. The Model of Biometric Identification Protocol with Random Parameter.

The last decade other human features have become practical, and there is now an active research community on iris-based recognition, face recognition and others.

Biometric identification systems were studied by O'Sullivan and Schmid [8] and Willems et al. [9]. They assumed storage of biometric enrollment sequences in the clear and determined the corresponding identification capacity. Later

Turcel [10] analysed the trade-off between the capacity of a biometric identification system and the storage space (compression rate) required for the biometric templates. It should be noted that Turcel's method realizes a kind of privacy protection scheme. Recall that secrecy capacity introduced by Ahlswede and Csiszar [11] can be regarded as the amount of common secret information that can be obtained in an authentication system in which helper data are (publicly) available. Interestingly this secrecy capacity, which is equal to the mutual information between enrollment and authentication biometric sequences in the biometric setting, equals the identification capacity found by O'Sullivan and Schmid and Willems et al.

3. Notations and Definitions

Let X, Y, Z, S be finite sets and W is a family of discrete memoryless channels $\mathrm{W_s} : X \rightarrow Y$, with input alphabet X and output alphabet Y. The s is the channel state, varying independently in each moment of the channel action with the same known PD $\mathrm{Q(s)}$ on S.

There are M individuals and each individual has an index $m = \{1, 2, ..., M\}$. A biometric data sequence $X(m) = \{x_1, x_2, ..., x_n\}$, where $x_n \in X, n = \overline{1, ..., N}$ corresponds to each individual m. All these sequences are supposed to be generated at random with a given probability distribution

$$P^N(x) = \prod_{n=1}^{N} P(x_n), \quad x \in X^N$$

Enrollment phase. Let we have the stationary and discrete memoryless channel $W(y \mid x, s)$ with random parameter. In this phase all biometric data sequences $\mathrm{x}(m)$ are observed via this channel. The state of the channel is changed by following probability distribution $\mathrm{Q}^N(s)$, it is mean

$$W^N(y \mid x, s) = \prod_{n=1}^{N} W(y_n \mid x_n, s_n), \ \mathrm{Q}^N(s) = \prod_{n=1}^{N} Q(s_n),$$
$$x \in X^N, y \in Y^N, s \in S^N$$

The resulting $\mathrm{y}(m)$ enrollment output sequences for all $m = \{1, 2, ...M\}$ are stored in a database.

Identification phase. In the identification phase the biometric data sequence of an unknown individuals observed via a same memoryless channel $W(z \mid x, s)$ with random parameter.

$$W^N(z \mid x, s) = \prod_{n=1}^{N} W(z_n \mid x_n, s_n), \ z \in Z^N, x \in X^N, s \in S^N$$

The resulting identification output sequence **z** is compared to the sequences $\mathrm{y}(m)$, $m = \{1, 2, ..., M\}$

$$W^*(y \mid x) = \sum_{s \in S} \mathrm{Q}(s) W(y \mid x, s), \ W^*(z \mid x) = \sum_{s \in S} \mathrm{Q}(s) W(z \mid x, s) \ .(*)$$

And

$$P^* = \{P^*(y) = \sum_{x} W^*(y \mid x) P(x), x \in X, y \in Y,$$

$$W^*(z \mid y) = \frac{\sum_{x} W^*(y \mid x) W^*(Z \mid X) P(x)}{P^*(y)} .$$

The channel $\mathrm{W_s} : Y \rightarrow Z$ is memoryless:

$$W^N(z \mid y) = \prod_{n=1}^{N} W(z_n \mid y_n), \ z \in Z^N, y \in Y^N .$$

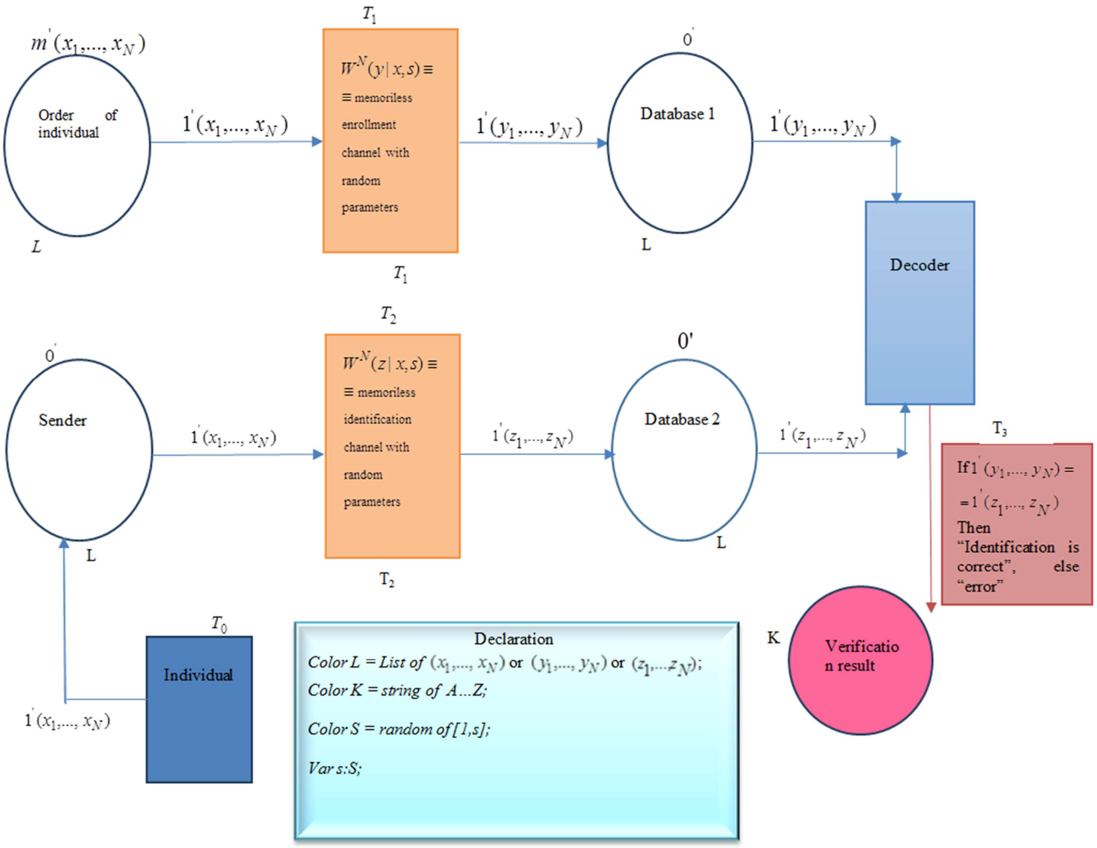
From (*) appointments follows that Biometric Identification Protocol with Random Parameter is brought to Biometric Identification system [12].

In Fig. 4 Colored Petri Net consists of the following positions: Order of individual, Sender, Database 1, Database 2, Verification result and the following transitions: T_0, T_1, T_2, T_3, which have corresponding names. Order of individual includes N-dimensional biometric data from m-dimensional set. Element $(x_1, ...x_N)$ passes through the channel to T_1 transition, and its corresponding encoded data $(y_1, ...y_N)$ is placed in Database 1.

After the firing of T_0 transition, the element of m-dimensional set is placed in *Sender* position, after that the result of firing of T_2 transition, through the respective channel, it is obtained the encoded data $(z_1, ...z_N)$, which is placed in the *Database 2* position. Through the firing of T_3 transition, it is checked the correspondence of vectors $(y_1, ...y_N)$ and $(z_1, ...z_N)$, gets the response: if the identification is correct, or there is an error in the system. The arc, which is directed from T_3 transition to the *Verification result* position, is bound by a corresponding logical expression.

Then firing of T_0 transition, the next element can queue up and the cycle can be repeated again.

Order of individual, Sender, Database 1, Database 2 positions that are attached to the L type, which is a set of N-dimensional vectors. The K type is attached to *Verification result* position, which is presented as a type string. The S type is presented random parameters. The corresponding information about the types is shown in the declaration table.



Declaration

Color L = *List of* $(x_1,\dots,x_N)$ or $(y_1,\dots,y_N)$ or $(z_1,\dots z_N)$;
Color K = *string of A...Z;*
Color S = *random of [1,s];*
Var s:S;

Fig. 4. The model of Biometric Identification Protocol with Random Parameter by Colored Petri Net.

4. Conclusion

As a result of the work, we presented Colored Petri Net that is modeled a biometric identification protocol with random parameter, which will explore verification, validation, error detection and functional interactions problems in the Biometric Identification Systems.

The obtained results of the study will be used for processing of biometric data used in medical equipment, reducing time and resource consumption.

References

[1] Peterson, James Lyle (1981). Petri Net Theory and the Modelling of Systems. Prentice Hall. ISBN 0-13-661983-5.

[2] Tadao Murata. "Petri nets: Properties, Analysis and Applications." Proc. of the IEEE, 77(4), 1989.

[3] K. Jensen. Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Springer - Verlag, Berlin, 1992.

[4] Jensen K. Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Springer, 1996. Vol. 1–3.

[5] S. Pankanti, R. M. Bolle and A. Jain, "Biometrics-The Future of Identification", IEEE Computer, vol. 33, no.2, pp. 46-49, February, 2002.

[6] J. D. Ullman, "Elements of ML Programming," Prentice- Hall, Upper Saddle River, 1998.

[7] S. Pankanti, R. M. Bolle and A. Jain, "Biometrics – The Future of Identification", IEEE Computer, V33, N2, pp. 46-49, 2002.

[8] J. A. O'Sullivan and N. A. Schmid, "Performance prediction methodology for biometric systems using a large deviations approach", IEEE Trans. On it Signal Proc., vol. 52, no. 10, pp. 3036-3045, 2004.

[9] F. WIllems, T. Kalker, J. Goselig, and J.-P. Linnartz, "On the capacity of a biometric identification system", Intenational Symposium on Information Theory, Yokohama, Japan, p. 82, 2003.

[10] E. Trucel, "Capacity/storage tradoff in high-dimentional identification system", IEEE International Symposium on Information Theory, Washington, USA, pp 1929-1933, 2006.

[11] R. Ahlswede and Csiszar, "Common randomness in information theory and cryptography – Part I: Secret sharing", IEEE Trans. Information Theory, vol. IT-39, pp. 1121-1132, July 1993.

[12] G.R. Petrosyan, L.A. Ter-Vardanyan, A.V. Gaboutchian, "Modeling of Biometric Identification System using the Colored Petri Nets" - The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XL-5/W6, 2015 Photogrammetric techniques for video surveillance, biometrics and biomedicine, 25–27 May 2015, Moscow, Russia.