Colored Petri nets for Modeling Processes of Biometric Identification System

Goharik Petrosyan[1], Andrey Avetisyan[2], Valery Burmin[3], Vladimir Knyaz[4], Armen Gaboutchian[5]

[1] Institute of Physics of the Earth named after O.Yu. Schmidt, RAS, 123242, Moscow, Russian Federation-petrosyan_gohar@list.ru
[2] Institute of Geophysics and Engineering Seismology after A. Nazarov of the National Academy of Sciences of Republic Armenia, Str.Gyumri, 3115, V. Sargsyan 5, Gyumry-avet.andrey@mail.ru
[3]Institute of Physics of the Earth named after O.Yu. Schmidt, RAS, 123242, Moscow, Russian Federation- vburmin@yandex.ru
[4]Moscow Institute of Physics and Technology (MIPT), 141700, Moscow, Russian Federation- knyaz@gosniias.ru
[5]Medical Institute, Peoples' Friendship University (RUDN), 117198, Moscow, Russian Federation - armengaboutchian@mail.ru

Keywords: Colored Petri net, token, position, transition, Biometric identification system.

Abstract

The article discusses some issues of modeling and digitalization of real-time systems using Petri nets (Petrosyan et al., 2025). The identification and secret key generation system is modeled using the Colored Petri net, which is a modern extension of the classical Petri net. The main goal of modeling the identification and secret key generation system using the Colored Petri net is to identify the presence of errors and randomness, the behaviour of the functioning, in the efficiency of the model. The Colored Petri net (CPN) graph of the identification and secret key generation system describes the workflow, logical actions much more simply, since Petri nets are convenient mechanisms for modeling, checking, validating complex systems. The article discusses some issues of modeling and digitalization of real-time systems using Petri nets. The identification and secret key generation system is modeled using the Colored Petri net, which is a modern extension of the classical Petri net (Petrosyan et al., 2025). The main goal of modeling the identification and secret key generation system using the Colored Petri net is to identify the presence of errors and randomness, the behavior of the functioning, in the efficiency of the model. The Colored Petri net (CPN) graph of the identification and secret key generation system describes the workflow, logical actions much more simply, since Petri nets are convenient mechanisms for modeling, checking, validating complex systems (Petrosyan et al., 2025).

1. Description of Petri Nets

1.1 Introduction

Petri nets were named after Karl A. Petri, who created a network-based mathematical tool for studying connections with automata in 1962 (Nagasaki, 2004), (Peterson, 1989) and (Petrosyan et al., 2025). A Petri net (also known as a place/transition net or P/T net) is one of several mathematical modeling languages for describing discrete distributed systems.

The further development of Petri nets was facilitated by the fact that they have properties that can be used to model both process synchronization and asynchronous events, parallel operations and conflicts, or resource sharing (Jensen, 1996) and (Petrosyan et al., 2025). These properties are used, for example, in industrial automation systems, communication systems, computer systems, nuclear power plants, aviation systems (Petrosyan et al., 2015) and (Petrosyan et al., 2025). Petri nets are used to design the architecture of Nokia mobile phones, in banking operations, interconnections at Hewlett Packard, in a word, they are used in real-time systems.

Using Petri nets, one can design universal or logical gates, which have firmly entered the logic of the digital life of society (Petrosyan et al., 2017) and (Petrosyan et al., 2025). Conventional Petri nets have limited properties in terms of modeling complex systems.

In 1996, a group at Aarhus University led by Professor Jensen K. created theory and software (CPN-Tools) for working with CPNs (Kristensen et al., 1998) and (Petrosyan et al., 2025).
Colored Petri nets (CPNs) are a graphically oriented language for designing, specifying, modelling and testing systems. It is particularly well suited for systems consisting of a number of processes that interact and synchronize. Typical application areas include communication protocols, distributed systems, automated manufacturing systems, workflow analysis and VLCI (very large-scale integration) chips (Jensen, 1996), and (Kristensen et al., 1998) and (Petrosyan et al., 2025).

The goal of in-depth study of various extensions of Petri nets (Colored Petri nets, Timed and Stochastic Petri nets, etc.) (Kristensen et al., 1998) and (Petrosyan et al., 2025) for modeling real-time systems (from the point of view of optimization) leads to the design of such technical means where it is necessary to minimize the costs of resources, time and maximize speed. Petri nets are similar in their properties and modeling capabilities to neural networks, Markov chains.

1.2 Description of Classical Petri net

A Petri net is a directed bipartite graph in which nodes represent transitions (i.e. discrete events that can occur, denoted by rectangles or stripes), places (i.e. conditions, denoted by circles) and directed arcs (describing which places are before and/or after the condition under which transitions occur, denoted by

arrows). Places (positions) of Petri nets can contain tokens, the presence and number of which change during the operation of the network. The state of the network is described using positions containing the corresponding number of tokens, and the actions that occurred in the network are described using transitions (Peterson, 1989) and (Petrosyan et al., 2025). Petri nets have been used to model fault tolerance and real-time safety.

A Petri net model can be described by a set of linear algebraic equations or other mathematical models reflecting the behaviour of the system. This opens the possibility of formal analysis of the model. Which in turn allows for formal verification of properties related to the behaviour of the underlying system, such as precedence relations between events, parallel operations, appropriate synchronization and elimination of deadlocks, repetitive actions and mutual exclusion of shared resources. A model based on simulation modeling can only give a limited set of states of the modeled system object and, thus, can only show the presence (but not the absence) of errors in the model and the underlying requirement specifications (Peterson, 1989) and (Petrosyan et al., 2025).

Let's give a simple example of a regular Petri net (Figure 1). In Figure 1, the Petri net initially has one token in position P_1. After the T_1 transition is triggered, the token from transition P_1 goes to position P_2 in double form, since transition T_1 is connected to transition P_2 by two arcs. Arcs describe the dynamics of the system. An arc entering a transition, as a result of actions, takes a certain number of tokens from the corresponding input positions, and an outgoing arc means that the transition assigns tokens to its output positions.

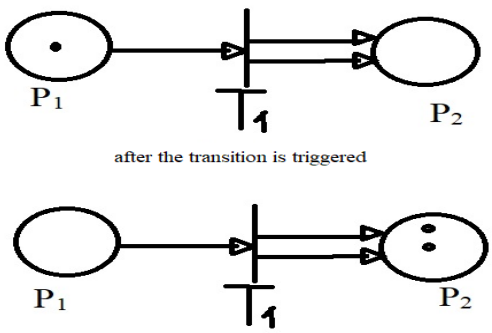


Figure 1. A simple example of a classical Petri net.

1.3 Description of Colored Petri net

Colored Petri nets are a modern extension of the regular or classical Petri nets (Jensen, 1996), (Kristensen et al., 1998), (Nagasaki, 2004) and (Petrosyan et al., 2025) adding the idea of a data type and a number of other ideas inherent in programming languages. As in classical Petri nets, each position is assigned a name. However, the names do not have any formal meaning; rather, they make the net clearer. Each position is assigned a data type, which defines the type of data in the position. Each token has a certain value related to the type of data attached to the position containing the marking. Each token is attached to an integer, which is the coefficient of this marking in the position, i.e. the number of tokens with this value in the given position is determined. For a given moment in time, the network is characterized by its state, which is the types and number of tokens in each position.

Before the CPN starts working, its positions are assigned a certain number of tokens belonging to the types of the corresponding data that describe the initial state of the system, i.e. the network is initialized (Jensen, 1996), (Kristensen et al., 1998) and (Petrosyan et al., 2025). The token types themselves are called colors.

Colored Petri nets are similar to high-level programming languages, while classical Petri nets are similar to low-level programming languages. In Colored Petri nets, unlike standard Petri nets, a position can contain tokens of arbitrary complexity: records, lists, etc., which allows creating more reliable models, simulating complex systems (Kristensen et al., 1998) and (Petrosyan et al., 2025).

Here, too, the events of the modeled system are described by transitions. The number of removed or added chips is determined by arc expressions, which can have a simple form, for example, be a constant belonging to some type, in which case the arc can only transfer a chip with a given value. The expression can also consist of variables, contain a conditional function that determines the number of chips transferred along this arc. If the expression does not contain variables, it is called a closed expression. In arc expressions, variables must be assigned a specific value so that these expressions can be calculated or evaluated. However, not all such assigned assignments lead to the possible execution of the transition. Only in this case is the execution of the transition permissible for the corresponding input positions to contain the corresponding number of chips with such a value. The above mentioned properties make the structure of the Colored Petri net dynamic in the following sense: if the number of transmitted tokens in classical Petri nets is determined by the number of arcs, and these arcs are fixed in advance and they determine the static structure of the net, then in Colored Petri nets the number of tokens during the firing of the transition is determined by an arc expression, which, in the case of different values of the variables included in it, can transmit different quantities. In addition, in complex real-time systems it is often necessary to apply the priority principle for different orderings of transition firing, depending on the conditional function assigned to the arc. In this regard, classical Petri nets have limited capabilities for modeling and implementing such systems, compared to Colored Petri nets (Jensen, 1996), (Kristensen et al., 1998) and (Petrosyan et al., 2025).
Let's give a simple example of a Colored Petri Net (CPN) (Figure 2).

A token in a colored Petri net is an object that has a property, for example, water is cold or hot (Figure 2). The model describes the operation of an automatic coffee maker that receives an order at the input and receives ready-made coffee at the output. The first step is the process of initializing the network, the second step is grinding the beans and heating the water, where the water object changes its property, in this case, *temperature*. The grain object changes its property, in this case, *shape* (ground coffee). The third step is the process of brewing coffee. The colored Petri net in this case is quite simple, since there are no logical and trigger expressions or functions on the arcs.

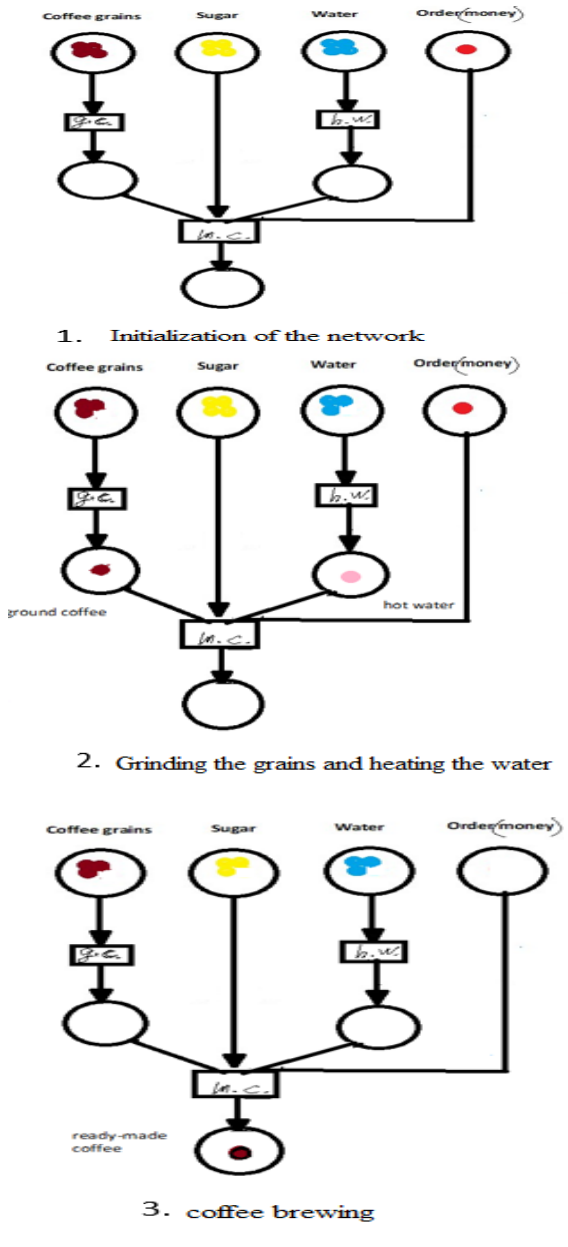


Figure 2. CPN model for automatic coffee maker.

2. Mathematical definitions of Petri nets

2.1 Mathematical definition of Classical Petri Net

The structure of the network is a quartet of elements $C = (P, T, I, O)$, where P and T are finite sets of positions and transitions, and $P \bigcap T = \emptyset$. The input function I maps the transition t_j to the set of input positions $I(t_j)$, and the output function O maps the transition t_j to the set of output positions $O(t_j)$, i.e., the transition can have both input and output positions, and with repetitions (Peterson, 1989).
Let us give a formal definition of the classical Petri net.

Definition 1: A Petri net is a pair $M = (C, \mu)$, where $C = (P, T, I, O)$ is the structure of the net, and μ is the state of the net. In the structure C, P and T are finite sets of positions and transitions, $I : T \to P^{\infty}, O : T \to P^{\infty}$ are input and output functions, respectively, where P^{∞} are all possible (with repeating elements) sets from the set P, $\mu : P \to N_0$ is the state function, where $N_0 = \{0,1,\cdots\}$ is the set of all positive integers. In a known way, the permissible transitions of the Petri net are determined, and the transition of the network from the state μ to the state μ' is the set of achievable states (Peterson, 1989) and (Petrosyan and Ter-Vardanyan, 2016).
The state of the network is understood as the following vector: $(\mu(P_1), \mu(P_2), \cdots, \mu(P_n))$, if $P = \{P_1, P_2, \cdots, P_n\}$.
The function mm performs the following mapping: $\mu : P \to N_0$, where N_0 is the coding of the number of tokens in positions. Before starting work, the network must be in the initial state: $(\mu_0(P_1), \mu_0(P_2), \cdots, \mu_0(P_n))$, where the number of tokens in the specified positions is noted. Let us assume that we have a Petri net $M = (C, \mu)$.

We say that in the state μ the transition $t_j \in T$ is allowed to operate if $\forall P_i \in I(t_j), \mu(P_i) \geq \#(P_i, I(t_j))$, where $\#(x, A)$ is a function depending on the arguments, which specifies as its value the number of entries in the set A (Peterson, 1989) and (Petrosyan and Ter-Vardanyan, 2016).
Let's assume that in the state μ the transition t_j is allowed to fire, and it is executed. In this case the network will go to a new state $\mu' = \delta(\mu, t_j)$, which is defined as follows:

$$\forall P_i \in P, \mu'(P_i) = \mu(P_i) - \#(P_i, I(t_j)) + \#(P_i, O(t_j)).$$

Here is a function $\delta(\mu, t_j)$, depending on two arguments, which returns the network to the state it would go to when, while in the state μ, a valid transition t_j is fired.
The initial state of the network is denoted by μ. Let us formulate a set of achievable states. Let us denote it as $R(C, \mu_0)$.

1. $\mu_0 \in R(C, \mu_0)$ is the initial state;
2. if $\mu' \in R(C, \mu_0)$ and $\exists t_j$ are such a transition that $\delta(\mu, t_j) = \mu''$, then $\mu'' \in R(C, \mu_0)$;
3. other states do not belong to $R(C, \mu_0)$. $R(C, \mu_0)$ can be infinite (Peterson, 1989) and (Petrosyan and Ter-Vardanyan, 2016).

2.2 Mathematical definition of Colored Petri Net

Definition 2: A Colored Petri net is described by nine elements $CPN = (\Sigma, P, T, A, N, C, G, E, I)$, which satisfy the

following requirements (Jensen, 1996), (Kristensen et al., 1998) (Nagasaki et al., 2004) and (Petrosyan and Ter-Vardanyan, 2016:

1. Σ is a finite set of non-empty types called color sets;
2. P is a finite set of places (positions);
3. T is a finite set of transitions;
4. A is a finite set of arcs such that $P \bigcap T = P \bigcap A = T \bigcap A = \emptyset$;
5. N is a node function, a mapping from A onto $P \times T \bigcup T \times P$;
6. C is a color function, a mapping from P onto Σ;
7. G is a descent function, a mapping from T for expressions such that
$\forall t \in T : [Type(G(t)) = Bool \wedge Type(Var(G(t))) \subseteq \Sigma]$,
8. E is an arc expression function, a mapping from A for an expression such that the $\mathrm{p(a)}$ position belongs to $N(a)$;
9. I is an initialization function, a mapping from P onto closed expressions such that

$$\forall p \in P : [Type(I(p)) = C(p)_{MS}].$$

3. Biometric identification and secret key generation system

3.1 Schematic model of the biometric identification and secret key generation system

In the modern world, biometrics or biometric recognition accompanies us at almost every step, is widely used both in public administration and in private companies (military forces, border control, healthcare, financial and banking sector, physical access control or individual login, payments of social benefits, etc.) (O'Sullivan et al., 2004), (Petrosyan et al., 2025).

More recently, Trucel (Trucel, 2006) analyzed the trade-off between the capacity of a biometric identification system and the amount of memory (compression ratio) required for biometric templates. It should be noted that Trucel's method implements a kind of privacy protection scheme (Trucel, 2006). Recall that the privacy level, introduced by Ahlswede and Csiszar (Trucel, 2006), can be considered as the amount of total secret information that can be obtained in an authentication system in which the supporting data is available (public). Interestingly, this secrecy, equal to the mutual information between biometric registration and authentication sequences in biometric settings, is equal to the identification power found by O'Sullivan and Schmidt (O'Sullivan et al., 2004), (Petrosyan et al., 2025) and (Willems et al., 2003).

In this system, two terminals monitor the biometric sequences of registration and identification of a group of individuals (Petrosyan and Ter-Vardanyan, 2016). The first terminal generates a secret key for each registered person and stores the corresponding supporting data in a public database. This auxiliary data, on the one hand, contributes to the reliable recovery of the secret key, and on the other hand, makes it possible to determine the identity of the person for the second terminal based on the presented biometric identification sequence. The database assumes that all supporting data is publicly available. Because the biometric secrets created by the first terminal are used, for example, for data encryption.

In this paper, the system under consideration has two terminals monitoring biometric sequences of registration and identification of a group of persons.

Let us consider a biometric schematic model of the identification and secret key generation system proposed by F. Willems (Willems et al., 2003), (Figure 3).

The following notations are used in Figure 3. Capital letters are used for random variables (RV). X, Y are values from finite sets X, Y respectively and lowercase letters x, y for implementations. Small bold letters are used for $\mathrm{x} = (x_1, ..., x_N) \in \mathsf{X}^N$ vectors of N-length. The power of the set X is denoted by $|\mathsf{X}|$ (Petrosyan and Ter-Vardanyan, 2016). The notation $|\mathrm{a}|^+$ will be used for $\max(a,0)$. Lowercase letters m, s are used for the secret key and auxiliary data.

As already mentioned above, the model of the biometric identification and secret key generation system consists of registration and identification procedures (Figure 3).

At the registration stage, ν individuals are observed. For each person $\nu \in \{1,2,..., \mathcal{V}\}$ in the system, the biometric data source creates a biometric registration sequence

$$\mathrm{x}(\nu) = (x_1, ..., x_N), \text{ where } x \in \mathsf{X}^N, n = \overline{1, N}.$$

It is assumed that all these sequences are generated randomly, with a given probability distribution

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

that is, the symbols x_n, $n = 1,2,..., N$ are independent and identically distributed (O'Sullivan et al., 2004), (Trucel, 2006), (Willems et al., 2003), (Petrosyan et al., 2015), (Petrosyan and Ter-Vardanyan, 2016), (Petrosyan et al., 2017).

During the registration procedure, for each person $\nu \in \{1,2,..., \mathcal{V}\}$, the biometric sequence $\mathrm{x}(\nu)$ is encrypted, encoded in the auxiliary data $m(\nu)$ and the secret key $s(\nu)$, therefore,

$$e(x(\nu)) = (m(\nu); s(\nu)), \text{ for } \nu \in \{1,2,..., \mathcal{V}\},$$

where $e(x(\nu))$ is the encoder mapping (this is a digital circuit that converts a set of binary inputs into a unique binary code) (Petrosyan and Ter-Vardanyan, 2016).
The auxiliary data $m(\nu)$ is then stored in the (publicly accessible) database at position, and the generated secret key

$s(v)$ is transmitted to the individual. The auxiliary data stored in the database enables reliable identification. It should contain only a small amount of information about the corresponding secret key (Petrosyan and Ter-Vardanyan, 2016).

During the identification procedure, a biometric identification sequence $\mathrm{y} = (y_1,..., y_N)$ is observed, consisting of N symbols of the finite alphabet Y (Petrosyan and Ter-Vardanyan, 2016).

This sequence is the output of a biometric channel whose input was the sequence $\mathrm{x}(v)$ or registering an unknown person v. If individual v has been observed, then sequence $\mathrm{y}(v)$ occurs with probability:

$$Q_c{}^N(\mathrm{y} \mid \mathrm{x}) = \prod_{n=1}^{N} Q(y_n \mid x_n) \quad x \in \mathrm{X}^N,\ y \in \mathrm{Y}^N,$$

because the biometric channel $Q_c{}^N(\mathrm{y} \mid \mathrm{x})$ has no memory.

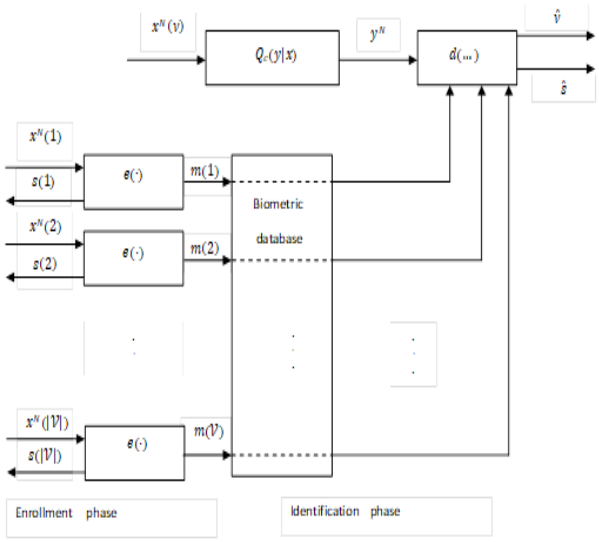


Figure 3. Schematic model of a biometric identification and secret key generation system.

During the identification process, when observing a biometric identification sequence y, the decoder generates an estimate $\hat{v} \in \{1,2,..., V\}$ of the identity of the observed person, as well as an estimate $\hat{s}(v)$ of his secret key:

$$(\hat{v}, \hat{s}(v)) = e(y, m(1), m(2),..., m(V)),$$

where e is the decoder mapping (Petrosyan and Ter-Vardanyan, 2016).

Moreover, the decoder's evaluation of the secret key $\hat{s}(v)$ assumes values from the same alphabet as the secret key generated during registration, hence $\hat{s}(v) \in \{1,2,.., V\}$.

3.2 CPN model for biometric identification and secret key generation system

Figure 4 shows the operation of the Colored Petri Net. Persons stand in a queue (registration stage), and in this case, the system is characterized as follows:

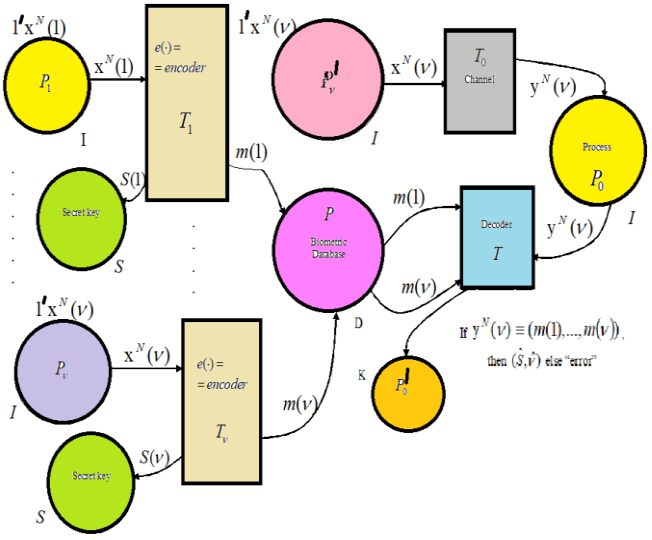
With the help of states (positions) of the $P_1, P_2, \cdots, P_v$ biometric data are registered with the corresponding encoder, when the transitions $T_1, T_2, \cdots, T_v$ are triggered, as a result of the implementation, a representation of each individual $S(1), S(2), \cdots, S(v)$ secret keys is generated, simultaneously and auxiliary data for each person.

Public auxiliary data $m(1), m(2), \cdots, m(V)$ are accumulated in the biometric database (position P).

Now let's describe the process of the identification stage:

in position P_v' there is a token $X^N(v)$, transition T_0 is a channel through which biometric data passes. According to the noise of the channel, a noisy version comes out and accumulates in position P_0 (process).

Through the transition T, the random $Y^N(v)$ is compared with each stored $m(1), m(2), \cdots, m(V)$ auxiliary data, and the data is decoded by the decoder, and then if they are equal, then the vector $(\hat{v}, \hat{s}(v))$ is obtained, where $\hat{s}(v)$ is the secret key and $\hat{v}$ is the individual, otherwise a failure occurs.



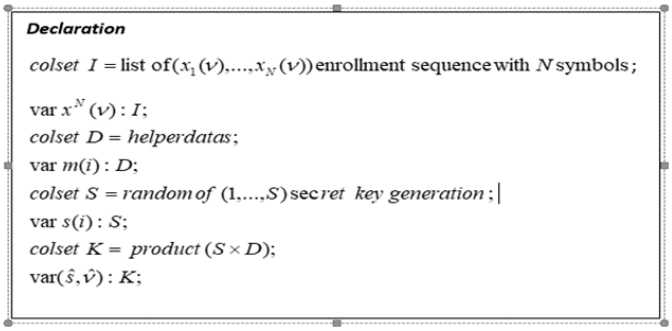


Figure 4. CPN model for biometric identification and secret key generation system.

In Figure 4 in Declaration those variables and types that participate in the Colored Petri Net are declared (Petrosyan et all, 2025).

4. Conclusions

The article describes the network modeling of the biometric identification and secret key generation system using the Colored Petri net and describes in detail the process of registration and identification of individuals' data, as well as the generation of secret keys (Petrosyan et all, 2025) and (Petrosyan and Ter-Vardanyan, 2016). Colored Petri nets are ideal for modeling real systems, studying their individual elements, checking, validating the system, detecting and preventing emergencies, for implementing and replacing individual system components with optimal options. This means that before creating such a system and implementing the process described above, Colored Petri nets are a very convenient simulation system in terms of process optimization and cost minimization.

References

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

Kristensen, M., Christensen, S., Jensen, K., 1998. The practitioner's guide to Coloured Petri Nets - Springer-Verlag.

Nagasaki, M., Doi, A., Matsuno, H., Miyano, S., 2004: A Versatile Petri Net Based Architecture for Modeling. Simulation of Complex Biological Processes, Genome Informatics, 15(1).

O’Sullivan, J. A., and Schmid, N. A., 2004: “Performance prediction methodology for biometric systems using a large deviations approach”, IEEE Trans. On it Signal Proc., vol. 52, no. 10, pp. 3036-3045.

Peterson, J.L., 1989: Petri Net Theory and the Modeling of Systems, PRENTICE-HALL, Englewood Cliffs, N.J.

Petrosyan, G., Avetisyan, A., Burmin, V., 2025: Simulation of Biometric Identification System by Colored Petri Nets. Biomedical Journal of Scientific & Technical Research, vol. 60, issue 3, pp. 52570-52573. ISSN: 2574-1241. http://dx.doi.org/10.26717/BJSTR.2025.60.009449

Petrosyan G.R., Ter-Vardanyan L.A., Gaboutchian A.V., 2015: “Modeling of biometric identification system using the Colored Petri Nets”, *The International Society for Photogrammetry and Remote Sensing WGV/5 and WGIII/3 Workshop Photogrammetric techniques for video surveillance, biometrics and biomedicine*, Moscow, RF, pp. 23-25.

Petrosyan G.R., Ter-Vardanyan L.A., 2016: “Modelling of identification and secret-key generation system with Colored Petri Net”, *International Conference on Control, Decision and Information Technologies (CoDIT)*, Saint Julian's, Malta, pp. 239-245. DOI: 10.1109/CODIT.2016.7593567.

Petrosyan G.R., Ter-Vardanyan L.A., Gaboutchian A.V., 2017: “Modelling of Biometric Identification system with given parameters using Colored Petri nets”, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLII-2/W4, 2nd International ISPRS Workshop on PSBB*, Moscow, RF, pp. 145-150.

Trucel, E., 2006: “Capacity/storage tradoff in high-dimentional identification system”, *IEEE International Symposium on Information Theory*, Washington, USA, pp. 1929-1933.

Willems, F., Kalker, T., Goselig, J. and Linnartz, J.-P., 2003: “On the capacity of a biometric identification system”, *Intenational Symposium on Information Theory*, Yokohama, Japan, p. 82.