The System Conception of Investigation of the Communication Security Level in Networks

Henryk Piech and Grzegorz Grodzki

Czestochowa University of Technology,
Dabrowskiego 73, 42201 Czestochowa, Poland
h.piech@adm.pcz.czest.pl, grzegorz.grodzki@icis.pcz.pl

Abstract. Communication processes have to be observed because of the possibility that different kinds of threats occur in the processes of exchanging information in a network. Those threats are connected with: decryption possibility, losing jurisdiction, believing and freshness of information, message interception by intruders, etc. We monitor the communication protocol during its operation. Standard security attributes (as proposed by Barrows, Abadi, Needham and others) have been introduced to analyze the chosen aspects of security. We also employed a standard set of rules which interrelate parts of communication operations with security aspects. Our previous research introduced a system that investigates security related issues. It could be employed for auditing purposes and/or to predict failures given different kinds of communication threats. In this paper the security analysis is continued in the direction of building the model of dynamic modification of chosen factors (adequate to security aspect) with prognosis possibility.

Keywords: protocol logic, probabilistic timed automata, communication security.

1 Introduction

Information is sent in the form of a message according to protocol systems that should guarantee: encryption safety, sufficient belief level, protection against intruders, and the freshness of information elements [1],[2]. Usually, many mutually interleaved protocols are used in networks [3],[4]. Obviously, information refers to a different group of users (usually, they are grouped in a pair). Therefore, security analysis will be referred to those groups and they will form the creation basis of the so called main security factor [1],[5]. Another main factor can include a set of messages, the public key, secret, nonce, etc. Security attributes may consist of: the degree of encryption, key and secret sharing, believing in sender or receiver, believing in user honesty, and the level of a message or nonce freshness [1]. M. Kwiatkowska indicates that security attributes are presented in the figure of probability parameters [6],[7]. This form is smart and very convenient. Therefore, we base our proposition on the transformation possibility

W. Abramowicz (Ed.): BIS 2013 Workshop, LNBIP 160, pp. 148–159, 2013.

of time attributes into probability characteristics. Apart from rule and time influences, we also regard intruder threats. The influence on security attributes is realized with the help of correction coefficients that also have a probabilistic form (according to the adopted approach). The above-mentioned rules are set on the basis of conditions that form actions which really appear in communication operations. In addition, one may observe that many works consider the division of protocols into operations and operations into actions [1],[4],[5],[8]. In our model we also exploit the so called tokens with binary character. A token directly appoints the secure or threat attribute level depending on the relation to a given security threshold. This type of approach improves the assessment reaction with respect to security state changing and helps in the estimation of distribution probabilities that lead to next stages and thereby to one of the forms of prognosis creation (presented in the further section)[10],[11],[12]. The proposed system, pertaining to the investigation of communication runs, can be easily realized with mutually converted probabilistic timed automata (PTA) and colored Petri nets (proven and shown in the works of M. Kwiatkowska [6],[7],[9]). These characteristics guarantee the effective realization of a parallel model.

2 Communication Protocol Actions and Attribute Grammars

The problem consists in the definition and recognition of actions. Rule conditions should be directly connected with actions, whereas their conclusion ought to be associated with attributes. The transformation of the run operation into an action is the first stage of action recognition. Each operation is divided into actions which are adequate to their function. The action definition is as follows:

Definition 1. A tuple $\{S, R, K, M, N, Ch, Ad\}$ is an action ac_v which may contain information about the sender (S), receiver (R), message (M), the character of dealing (Ch), additional information - e.g. secrets etc.(Ad).

- The sender is represented by one user or a set of users $S = \{s(1), s(2), ..., s(ls)\}$,
- The receiver is represented by one user or a set of users $R = \{r(1), r(2), ..., r(lr)\}$,
- Sender and receiver create a group of users that can be limited for the excluding possibilities of intruder activity.

Actions are both a part of protocol operation and influences on security attributes. The set of security attributes is defined by rules (their arguments and conclusions). In order to present the same example of rules, we should define the set of predicates of the communication BAN logic [4]:

$A \leftrightarrow^K B$ - users A and B communicate via shared key K,
$\rightarrow^K A$ - user A has K as its public key,
$A \Leftrightarrow^Y B$ - users A and B share Y as a secret,
$\{X\}_K$ - the message X encrypted by key K,
$< X >_Y$ - the message X with a secret Y attached,
$A| \equiv X$ - user A believes the message X,

$A \triangleright X$ - user A sees the message X,
$A \triangleleft X$ - user A sends the message X once,
$A| \Rightarrow X$ - user A has jurisdiction over X,
$\#(X)$ - the message is fresh.

Let us try to define the set of actions and attributes. In order to achieve this aim, we exploit the rules based on BAN logic:

1. Authentication rule – type I:
   if $(A| \equiv ((A \leftrightarrow^K B),\ A \triangleright X_K)$ then $(A| \equiv (B \triangleleft X)$.
   The rules can be interpreted as follows: if A and B shared key K and A sees message, then A believes that this message is from B.
2. Authentication rule – type II:
   if $(A| \equiv (\rightarrow^K B), A \triangleright X^C_{K-1}, C \neq A)$ then $A| \equiv (B \triangleleft X)$.
   The rules can be interpreted as follows: if A knows B's key, then A recognizes signed message from B.
3. Authentication rule – type III:
   if $(A| \equiv (A \Leftrightarrow^Y B), A \triangleright X^Y)$ then $(A| \equiv (B \triangleleft X)$.
   The rules can be interpreted as follows: if A and B shared secret Y and A sees message with attached secret Y, then A believes that this message is from B.
4. Nonce rule:
   if $(A| \equiv \#(X),\ A| \equiv (B \triangleleft X))$ then $A| \equiv (B| \equiv X)$.
   The rules can be interpreted as follows : if A believes that X is "current" and that B said X, then A believes that B believes X.
5. Jurisdiction rule:
   if $(A| \equiv (B| \Rightarrow X),\ A| \equiv (B| \equiv X)$ then $A| \equiv X$
   The rule can be interpreted as follows: if A believes that B has jurisdiction over X and A believes that B confirms X then A believes X.
6. Vision rule – type I:
   if $(A| \equiv (A \leftrightarrow^K B),\ A \triangleright \{X\}^C_K, C \neq A)$ then $A \triangleright X$.
   The rules can be interpreted as follows: A can see through encryption on a shared symmetric key, provided that the encryption was done by a user other than A itself.
7. Vision rule – type II:
   if $(A| \equiv (\rightarrow^K A), A \triangleright X^C_K, C \neq A)$ then $A \triangleright X$.
   The rules can be interpreted as follows: A knows its secret key, so it can decrypt message encrypted with public key.
8. Freshness rule:
   if $\#(X)$ then $\#(X, Y)$.
   The rule can be interpreted as follows: if X is fresh then $X \wedge Y$ is also fresh.

The idea of operation decomposition into actions may be presented on the basis of a simplified example. Obviously, there is a possibility to describe it as a set of actions by selecting a single operation; for example: operation $A \rightarrow B : NaK(a, b)$ (from ASF Handshake protocol) consists of actions:
$A \leftrightarrow K(a, b)B$ adequate description: $A, B, K(a, b), *, *, Shared\ key, *,$
$\rightarrow KA$ adequate description: $\{A, *, K(a, b), *, *, has\ key, *\}$,

$\#(Na)$ adequate description: $\{A, *, *, *, Na, nonce\ is\ fresh, *\}$,
where $\{*\}$ - irrelevant parameter in the described action.

In addition, we can exploit a system of coding to identify and recognize actions. This convenient approach consists of the coding weight system, e.g. binary, decimal, etc. The binary system is more extensive but it helps in describing particular actions more precisely. Generally, a simple coding system is proposed:

$$cta = \sum_{i=1}^{le} w_i * pos_e l(i) \ - \ the\ code\ of\ an\ action\ type, \tag{1}$$

where $w_i = 2^{i-1} \ - \ (or\ 10^{i-1})$ - position weight.

The mutual cooperation among recognition and attribute correction procedures is illustrated in Fig. 1. Chosen sets of attributes create security modules

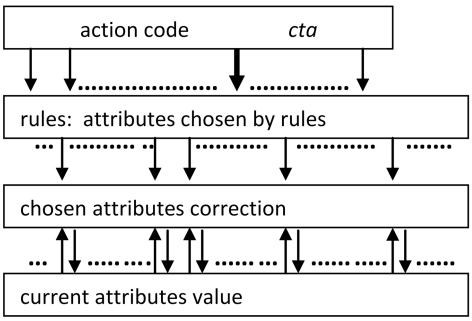

Fig. 1. Information flow in a result of action (cta) activity. Corrected attributes will become current attributes for the next action.

that concentrate around main factors like: communication protocols, keys, message service, and users.

3 Communication Security State and Node Structure

The security state is represented by an adequate security node structure.

Definition 2. A tuple (At, Th, Tk, na), where At - security attribute set, Th - the vector of a low level of feasible attribute values (thresholds), Tk - security tokens, na - the number of attributes, is a communication security state described as follows:

1. $At = \{at_1, at_2, ..., at_{na}\} \in [0,1]^n$ - the vector of attribute activation probabilities,
2. $Th = \{th_1, th_2, ..., th_{na}\} \in [0,1]^n$ - the vector of threshold attribute activation,

3. $Tk = \{tk_1, tk_2, ..., tkna\} \in \{0, 1\}n$ - the binary vector (token) of attribute activation: if $at_i \geq th_i$ then $tk_i = 1$ else $tk_i = 0$.

Attributes may express assertion about user honesty belief, belief about message freshness, assertion about attestation, assertion about the shared key, belief that the receiver has jurisdiction over the message, etc. The attributes are corrected with the help of modification functions in cases when the current action appointed the attribute by the rule [1]. The sequence of actions influences a given set of attributes. The attribute set defines communication security level. The security value is estimated on three levels:

- global,
- in reference to the main security factor (security module),
- in reference to particular attributes.

According to the proposed development approach, we propose several structures of the security main factor (security modules), (Fig. 2,3).

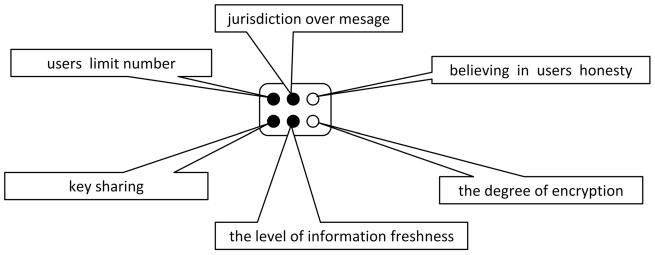

Fig. 2. The structure of the protocol security module, black - attribute activation, white - attribute has loosed activity

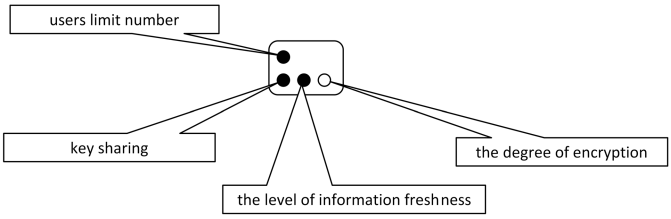

Fig. 3. The structure of the key security module

The action influences on attributes are conveniently presented and realized with the usage of equivalent tables that regard the above-mentioned rules. Therefore, the first table refers to action identifications and their characteristics, and

the second to attributes which will be corrected. In order to describe such situation, we consider two handshake operations as in Fig. 4,5 (column descriptions are adequate to action and attribute definitions):
Example:

1. $A \rightarrow B : \{N_a\}_{K(a,b)}$
2. $B \rightarrow C : \{N_b\}_{K(b,c)}$

These operations, belonging to two protocols, are decomposed into actions:

1. $A \leftrightarrow^{K(a,b)} B$ adequate description: $\{A, B, K(a,b), *, *, Shared\ key, *\}$,
2. $\rightarrow^{K} A$ adequate description: $A, *, K(a,b), *, *, has\ key, *$,
3. $\#(N_a)$ adequate description: $\{A, *, *, *, Na, nonce\ is\ fresh, *\}$,
4. $B \leftrightarrow^{K(b,c)} C$ adequate description: $\{B, C, K(b,c), *, *, Shared\ key, *\}$,
5. $\rightarrow^{K'} B$ adequate description: $\{B, *, K(b,c), *, *, has\ key, *\}$,
6. $\#(N_b)$ adequate description: $\{B, *, *, *, Nb, nonce\ is\ fresh, *\}$.

S1	*R1*	*S2*	*R2*	*M1*	*M2*	*N1*	*N2*	*K1*	*K2*	*Ch1*	*Ch2*	*Ch3*	*Ad1*	*Ad2*	*ac*
1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	1
1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	2
1	0	0	0	0	0	1	0	1	0	0	0	1	0	0	3
0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	4
0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	5
0	0	1	0	0	0	0	1	0	1	0	0	1	0	0	6

Fig. 4. Action descriptions in the binary convention - example, where column descriptions respect the action structure (definition 1), *ac* - action numbers

JM1	*JM2*	*Bh1*	*Bh2*	*Nf1*	*Nf2*	*De1*	*De2*	*Kr1*	*Kr2*	*Ul1*	*Ul2*	*ac*
1	0	1	0	0	0	0	0	1	0	0	0	1
1	0	0	0	0	0	0	0	1	0	0	0	2
1	0	1	0	1	0	0	0	1	0	0	0	3
0	1	0	1	0	0	0	0	0	1	0	0	4
0	1	0	0	0	0	0	0	0	1	0	0	5
0	1	0	1	0	1	0	0	0	1	0	0	6

Fig. 5. The appointment of attribute corrections according to BAN rules, where JM_i - jurisdiction over i-th message, Bh_i -believing in i-th user honesty, NF_i - the freshness of i-th nonce, De_i - the degree (over one) of i-th message encryption, Ks_i - i-th key sharing, Ul_i - the exceeding limit number of users seeing i-th message in an encrypted form

The connection between tables (through action numbers) allows us to realize appointed security attribute corrections. The set of attributes is chosen on the

basis of rules as well as time and heuristic functions [9]. In order to correct attributes we will use correction coefficients $CC(at(i))$ that were, previously predetermined for each attribute (Fig. 6). We present initial values of attributes by continuing the description of the example. Let us assume that the initial values of all attributes (obviously despite De) will be equal to 1(as a maximum value of trust probability). After 6 above-described actions (adequate 2 run operations), we may observe the following levels of attributes (Fig. 7). Attributes Nf and Ks are treated as timed attributes. Hence, the following formula is used for their correction: $at(k)(i) = 1 - ek - lf(at(i))$, where k - operation number [10].

correction coefficient of											
JM1	*JM2*	*Bh1*	*Bh2*	*Nf1*	*Nf2*	*De1*	*De2*	*Ks1*	*Ks2*	*Ul1*	*Ul2*
0,95	0,95	0,8	0,8	3	3	0	0	4	4	0,75	0,75

Fig. 6. Correction coefficient - example values. Attributes Nf and Ks are timed attributes, therefore their lifetimes $lf(N)$ and $lf(K)$ are given. Let us pay attention to De coefficient which is used for the *den*-th times blocking the correction of the adequate Ks attribute (obviously, only in the case when $De > 0$), where *den* - the degree of encryption.

attribute values after 2 communication eperations											
JM1	*JM2*	*Bh1*	*Bh2*	*Nf1*	*Nf2*	*De1*	*De2*	*Ks1*	*Ks2*	*Ul1*	*Ul2*
0,857	0,857	0,64	0,64	0,632	0,865	0	0	0,865	0,95	1	1

Fig. 7. The states of security attributes after two example operations

By considering the structure of tokens (binary structure) and the established threshold for all attributes on the level equal to $0,7$, it is possible to depict a security situation regarding different security modules (main factors) (Fig. 8). By treating all attributes with the same validity, we can estimate the level of security for all modules. This problem can be realized by the multiplication of specified component attribute probability values:
$SL(protocolA, B) = 0,875 * 0,64 * 0,632 * 1 * 0,865 * 1 = 0,306,$
$SL(protocolB, C) = 0,875 * 0,64 * 0,865 * 1 * 0,95 * 1 = 0,46,$
$SL(messageA, B) = 0,875 * 0,64 * 0,632 = 0,354,$
$SL(messageB, C) = 0,875 * 0,64 * 0,862 = 0,483,$
$SL(keyA, B) = 0,632 * 1 * 0,865 * 1 = 0,547,$
$SL(keyB, C) = 0,865 * 1 * 0,95 * 1 = 0,821,$
$SL(usersA, B) = 0,64 * 0,632 * 0,865 * 1 = 0,350,$
$SL(usersB, C) = 0,64 * 0,865 * 0,955 * 1 = 0,404.$
Similarly, token variant can be calculated on the basis of the estimation percent of active token participation in the full token set for a given security module (the main security factor). We may graphically present the security spectrum on the basis of the results estimated above.

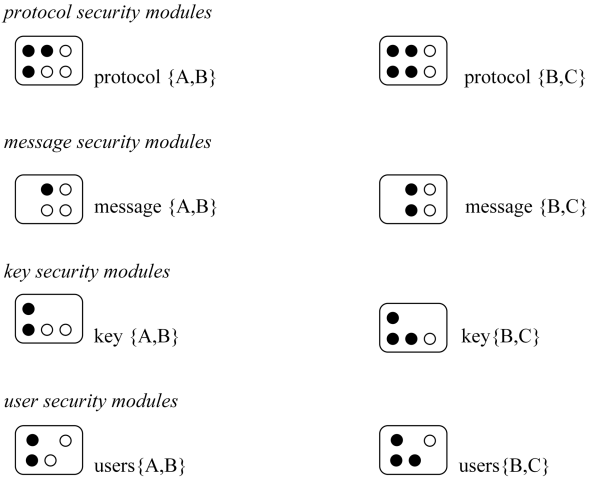

Fig. 8. The states of security modules, where A,B,C users

4 Probabilistic - Time Automata as Communication Security Investigation Model

We propose to use probabilistic - time automata (PTA) and converted to them colored Petri nets as a main tool for the investigation of communication security according to selected main factors, like: protocols, users, keys, messages, etc. The nodes (presented in Fig. 3,4 as examples) will be fundamental part of PTA. The global structure of proposed PTA is presented in Fig.9. If any attribute is decreased to an unacceptable level then there is not possibility to improve its value and security features cannot be increased. To regard the time parameter with intrinsic characteristic according to security aspect, we propose the following definition:

Definition 3. A probabilistic timed automaton PTA is a tuple in the form $(L, l', X, \sum, inv, p)$ where:

- L is a finite set of locations,
- $l' \in L$ is the initial location,
- X is a finite set of clocks (for each attribute),
- $\sum$ is a finite set of possible steps, of which $\sum_c \in \sum$ are declared as being current possible,
- the function $inv : L \to CC(X)$ is the invariant condition,
- the finite set $p \subseteq L \times CC(X) \times \sum \times Dist(2^X \times L)$ is the probabilistic edge relation.

A time state of a probabilistic timed automaton is a pair (l, v) where $l \in L$ and $v \in T^X$ are such that $v \in inv(l)$. Informally, the behavior of a probabilistic timed automaton can be understood as follows. The model starts in the initial

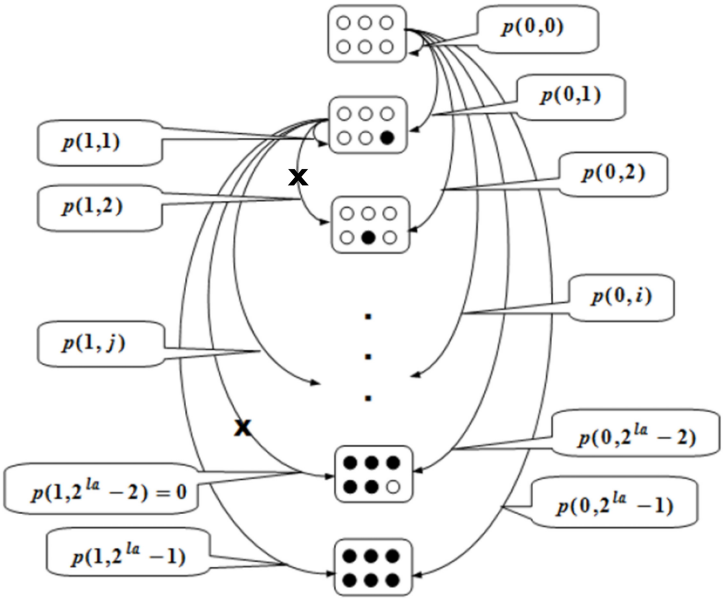

Fig. 9. Scheme of probabilistic - time automaton for communication security investigation, where $p(i, j)$ - the probability of state changing: from state i to j; $j \geq i$

location l' with all clocks set to 0, that is, in the state $(l', 0)$. In this, and any other state (l, v), there is a nondeterministic choice of either (1) making a discrete transition or (2) letting time pass. In case (1), a discrete transition can be made according to any probabilistic edge $(l, g, \sigma, p^*) \in p$ with source location l which is enabled; that is, the zone g is satisfied by the current clock valuation v. Then the probability of moving to the location l" and resetting all of the clocks in X to 0 is given by $p^*(X, l")$. In case (2), the option of letting time pass is available only if the invariant condition $inv(l)$ is satisfied while time elapses and an enabled probabilistic edge with a current step does not exist. Note that a timed automaton [2] is a probabilistic timed automaton for which every probabilistic edge $(l, g, \sum, p^*)$ is such that $p^* = \mu(X, l")$ (the point distribution assigning probability 1 to $(X, l")$) for some $(X, l") \in 2^X \times L$.

Generally, the above formalisms are adequate to a single secure attribute (or action) connected with a single message. In order to adapt them to real communication runs, indexes are to be introduced to distinguish the threat zones. It is assumed that each message can be described by a set of security attributes. These attributes are involved with assertion, believing the sending, receiving messages, encryption, decryption by keys, nonce generation, attaching secrets, etc. Additionally, we should regard the number of users and their character (honest, intruder) [1],[2]. These considerations are based on time influences on chosen security attributes, strictly on their level (value).

The general system structure includes modules of security element definitions, input protocol operation descriptions, input table of dependence rules, lifetime checkers, and user set correctors. Each protocol operation includes strategic elements which will be chosen according to communication security, and hence, they will be named security elements. It should be said that security elements have independent or partly involved character. So, in relation to communication security, the mechanism gives us result information which consists of different aspects of security. One of the assumptions, which helps us to create and exploit probabilistic timed automata (PTA), is a finite set of the security state. The security element values are difficult to estimate. In this variant, according to known applications, we use their probability evaluation (security elements - tokens). Then, we introduce activation bounds (low and up) for each token. If the probability of the security attribute P_i (value of elements) is between $Plow_i$ and Pup_i then the adequate token is activated $tk_i = 1$, otherwise the token loses its activation $tk_i = 0$ (Fig. 10). This strategy allows us to define the finite set of states; their number will be equal 2^{le}, where le - the number of tokens (security elements). The best security situation is described by all activated tokens

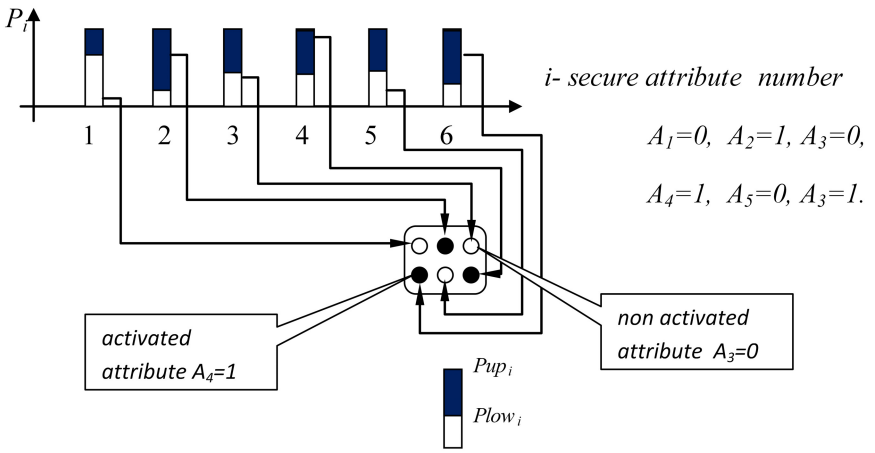

Fig. 10. Token binarization

$\forall_{i=1,2,\ldots,le}\ tk_i = 1$. The states can be changed during the realization of a protocol operation, but it is not necessary. The reasons for the non changing state are as follows:

- stable security attribute activation probability level $\forall_{i=1,2,\ldots,le}\ P_i^{(j+1)} = P_i^{(j)}$, where j - the number of the protocol operation,
- stable token activation $\forall_{i=1,2,\ldots,le}\{tk_i^{(j+1)} = tk_i^{(j)}\ |\exists_{k\in\{1,2,\ldots,le\}} P_k^{(j+1)} \neq P_k^{(j)}\}$.

The protocol operation description is a complex process because the following information should be provided:

- the scale of message encryption,
- the source and target of message transmition in pre-assumption,
- sharing keys (secrets) of users,
- belief (or assertion) parameters about security elements (attributes),
- lifetime parameter referring to a message, key, secret, nonce,
- the set of active users (honest and intruders).

It is assumed that in one operation only one message and one nonce will be sent, and that a set of keys and secrets will be exploited to respect standard rules (convention). However, also in this case, we have to use the multi-dimension table (due to the set character of the identical parameters). This situation can be illustrated by the tree structure depicted in Fig. 11. One part of operation

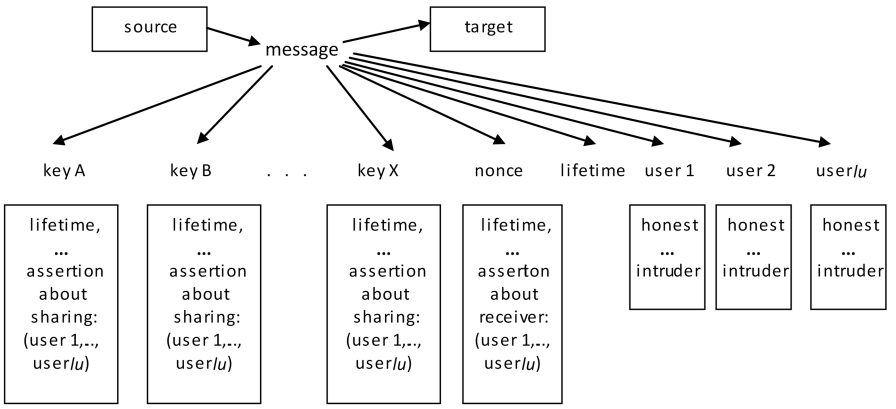

Fig. 11. The tree illustration of protocol operation description

parameter description has strictly deterministic character (real or integer) and the other part has probabilistic values. Time parameters are included in the first and belief of assert estimators in the second group. The assertion about honest or dishonest user behavior is obviously included in the assertion characteristic. If user honesty is certain then initial probability of this attribute is equal to 1: $P_i^{(0)} = 1$. The lifetime parameter is obviously constant but the time of adequate element activation increases according to real process realization (protocol operations). It is necessary to define and input the presented set of data for every message (message rules). They are the basis for using the chosen set of rules that change the level of attributes (security elements and next tokens). In order to present an identical example of message rules, we should define the set of communication logic predicates.

5 Conclusions

In the proposed approach it is possible to realize communication operation auditing and dynamically estimate the full spectrum of security aspects. The investigation is based on correction security attributes regarding rules, lifetimes and heuristic. Generally, the proposed algorithm is simple, but the preparation of dealing with the subject, which consists in the creation of security module structures and correction coefficient evaluation on the basis of experiences, can be more absorbing for communication security analytics. These structures and parameters should respect concrete situations and regard network information transfer and possible communication threats connected with different protocol realizations.

References

1. Burrows, M., Abadi, M., Needham, R.: A Logic of Authentication. Robert Harper., Logics and Languages for Security, pp. 15–819 (2007)
2. Focardi, R., Gorrieri, R.: A Classification of Security Properties. Journal of Computer Security 3, 5–33 (1995)
3. Beauquier, D.: On Probabilistic Timed Automata. Theoretical Computer Science 292, 65–84 (2003)
4. Gray III, J.W.: Toward a Mathematical Foundation for Information Flow Security. Journal of Computer Security 1, 255–294 (1992)
5. Evans, N., Schneider, S.: Analysing Time Dependent Security Properties in CSP Using PVS. In: Cuppens, F., Deswarte, Y., Gollmann, D., Waidner, M. (eds.) ESORICS 2000. LNCS, vol. 1895, pp. 222–237. Springer, Heidelberg (2000)
6. Kwiatkowska, M., Norman, G., Segala, R., Sproston, J.: Automatic Verification of Real-time Systems with Discrete Probability Distribution. Theoretical Computer Science 282, 101–150 (2002)
7. Kwiatkowska, M., Norman, R., Sproston, J.: Symbolic Model Checking of Probabilistic Timed Automata Using Backwards Reachability. Tech. rep. CSR-03-10, University of Birmingham (2003)
8. Alur, R., Dill, D.L.: A Theory of Timed Automata. Theoretical Computer Science 126, 183–235 (1994)
9. Szpyrka, M.: Fast and flexible modeling of real-time systems with RTCP- nets. Computer Science, 81–94 (2004)
10. Tadeusiewicz, R.: Introduction to Inteligent Systems. In: Wilamowski, B.M., Irvin, J.D. (eds.) Chapter No 1 in Book. The Industrial Electronic Handbook. CRC Press, Boca Raton (2011)
11. Tadeusiewicz, R.: Place and role of Intelligence Systems in Computer Science. Computer Methodsin Material Science 10(4), 193–206 (2010)
12. Alur, R., Courcoubetis, C., Dill, D.L.: Verifying Automata Specifications of Probabilistic Real- Time Systems. In: Huizing, C., de Bakker, J.W., Rozenberg, G., de Roever, W.-P. (eds.) REX 1991. LNCS, vol. 600, pp. 28–44. Springer, Heidelberg (1992)