IADIS International Journal on Computer Science and Information Systems
Vol. 8, No. 2, pp. 132-146
ISSN: 1646-3692

INTELLIGENT SYSTEM OF COMMUNICATION SECURITY INVESTIGATION

Henryk Piech. *Czestochowa University of Technology, Dabrowskiego 69,Poland*

ABSTRACT

In the presented investigation, communication security aspects are analyzed in an audited protocol operation run. The run is represented by mutually interleaved protocols. Each protocol consists of communication operations and these operations are decomposed into action sets. An action can play the role of an argument in logic rules described in works [3]. Rules help to extract security attributes which will be corrected after every operation. The action recognition is part of the proposed method and the adequate algorithm refers to protocols, keys, messages and users. The correction part refers to jurisdiction over message, believing in user honesty, key sharing, information freshness, the degree of encryption, etc. The investigation analysis and equivalent procedures are not complex and may be realized dynamically during the auditing process.

KEYWORDS

Security investigation, probabilistic timed automata, auditing

1. INTRODUCTION

The application of the presented investigation is associated with intelligent systems servicing information exchange in networks [12, 13]. Security problems consist in observing security aspects (factors) on different hierarchy levels [5] during the communication operation run realization [7]. Among them there could be: account protocols, keys, users, etc. [10]. One of auxiliary but important security aspects regards the possibility of threaten situation prognosis [6, 8]. The proposed approach is connected with the decomposition of the communication run into detailed (even atomic) actions [9]. These fundamental elements dynamically and systematically influence security factors. The run decomposition prepared in this way permits to exploit communication logic rules presented in [4]. Obtained results, being a set of conclusions, are instantly used for security attribute corrections. Another problem, omitted in this research presentation, concerns the experienced strategy of the estimation correction coefficients referring to a different kind of attributes. The same matter is connected with timed attributes, for which the lifetime parameter has to be evaluated. Attributes will be corrected by

heuristic rule described in [1], basing on the user number. The presented materials consider questions connected with action recognition and attribute modification. The full strategy assumes the possibility of realizing the auditing process on the parallel conversion structure [14]. For algorithm creation it is convenient to prepare a model in the form of probability timed automata [2, 3] or as a Petri net structure. Colored or time Petri nets [11] are the most adequate to simulate security state transitions.

2. ACTIONS AND ATTRIBUTE GRAMMARS

For the presentation of the same example of rules the set of predicates of communication BAN logic [4] should be defined:

$Q \leftrightarrow^{K} R$ -users Q and R communicate via the shared key *K*,
$\rightarrow^{K} Q$ - user Q has *K* as its public key,
$Q \Leftrightarrow^{Y} R$ - users Q and R share X as a secret,
$\{X\}_K$ - the message X encrypted by the key *K*,
$\{X\}_K{}^{Q}$ - the message X encrypted by the key *K* by the user Q,
$<X>_Y$ - the message X with a secret Y attached,
$Q \mid\equiv X$ - user *Q* believes the message X,
$Q \rhd X$ - user *Q* sees the message *X*,
$Q \lhd X$ - user *Q* sends the message *X* once,
$Q \mid\Rightarrow X$ - user Q has jurisdiction over X,
$\#(X)$ - the message is fresh.

Let us try to define the set of actions and attributes. For this aim, the rules based on BAN logic will be exploited:

1. Authentication rule - type I:

if $(A \mid\equiv ((A \leftrightarrow^{K} B), A \rhd \{X\}_K)$ then $(A \mid\equiv (B \lhd X)$

The rule can be interpreted as follows: if A and B shared the key K and A sees the message *X*, then A believes that this message is from B.

2. Authentication rule type II:

if $(A \mid\equiv (\rightarrow^{K} B), A \rhd \{X\}_{K^{-1}}{}^{R}, R \neq A)$ then $A \mid\equiv (B \lhd X)$

The rule can be interpreted as follows: if *A* asserts that *B* has the key *K* and *A* sees the message *X* encrypted by K^{-1}, then A believes that X is sent by B.

3. Authentication rule type III:

if $(A \mid\equiv (B \leftrightarrow^{Y} A), A \rhd \{X\}_Y)$ then $(A \mid\equiv (B \lhd X)$

The rule can be interpreted as follows: if *A* and *B* shared the secret *Y* and this secret is attached to the message X then *A* believes that *X* is sent by *B*.

1. Nonce rule

if $(A \mid\equiv \#(X), A \mid\equiv (B \lhd X))$ then $A \mid\equiv (B \mid\equiv X)$

The rule 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 $(A \mid\equiv (B \mid\Rightarrow X), A \mid\equiv (B \mid\equiv X)$ then $A \mid\equiv X$

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 $(A \mid\equiv (A \leftrightarrow^{K} B),\ A \rhd \{X\}_{K}^{R},\ R \neq A)$ then $A \rhd X$.

The rule can be interpreted as follows: if A and B shared the key K and A sees the message *X*, encrypted by the shared symmetric key, and the encryption was done by another *A* user then *A* sees *X*.

7. Vision rule - type II

if $(A \mid\equiv (\rightarrow^{K} A),\ A \rhd \{X\}_{K}^{R},\ R \neq A)$ then $A \rhd X$.

The rule can be interpreted as follows: if *A* asserts that he has the key *K* and *A* sees the message *X* encrypted by *K*, and the encryption was done by another *A* user then *A* sees *X*.

8. Vision rule - type III

if $(A \mid\equiv (\rightarrow^{K} A),\ A \rhd \{X\}_{K}^{-1\,R},\ R \neq A)$ then $A \rhd X$.

The rule can be interpreted as follows: if *A* asserts that he has the key *K* and *A* sees the message *X* encrypted by *K*, and the encryption was done by another *A* then *A* sees *X*.

9. 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.

10. The continuation of the jurisdiction rule

if $(A \mid\equiv (\forall x_1 \forall x_2 \ldots \forall x_n), (B \mid\Rightarrow X))$ then $(A \mid\equiv [a_1, a_2, \ldots, a_n / x_1, x_2, \ldots, x_n], (B \Rightarrow X))$

The rule can be interpreted as follows: if *A* believes in *B's* jurisdiction over all message components *X:* $x_1, x_2, \ldots, x_n$, then *A* believes that each component can be exchanged by the other element $a_1, a_2, \ldots, a_n$ with the continuation of *B's* jurisdiction over the new structure of *X*.

11. Secret transitivity rule

if $\{X\}\ A \rightarrow B$ then $Y\ \{X \wedge B \rhd Y\}$.

The rule can be interpreted as follows: if *A* sees *X* from *B* then the message secret consists of *X* and the secret that *B* sees secret *Y*.

12. General transitivity rule (Horae Logic [4])

if $(X \mid\!- X',\ \{X'\} s \{Y'\},\ Y' \mid\!- Y)$ then $\{X\} s \{Y\}$.

The problem consists in definition and recognition actions. Rule conditions should be directly connected with actions. Rule conclusion should be connected with attributes. The transformation of the run operation into action is the first stage of action recognition.

Each operation is divided into actions which are adequate to the function that they played. 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 $S=\{s(1), s(2),\ldots, s(ls)\}$,
- The receiver is represented by one user or a set of users $R=\{r(1), r(2),\ldots, r(lr)\}$,
- Sender and receiver create the group of users which can be limited for the excluding possibilities of intruder activity.

The idea of the decomposition of the operation into actions may be presented on the basis of a simplified example. By selecting a single operation, it is possible to describe it as a set of actions, obviously; for example: operation $A \rightarrow B$: $\{N_a\}_{K(a,b)}$ (from ASF Handshake protocol) consists of actions:

$A \leftrightarrow^{K(a,b)} B$ - adequate description: {*A, B,* $K_{(a,b)}$ *, *, * , Shared key, **},

$\rightarrow^{K} A$ - adequate description: {*A, *,* $K_{(a,b)}$ *, *, * , has key, **},

$\#(N_a)$ - adequate description: {*A, *, * , *,* N_a *, nonce is fresh, **},

where

{*}* – irrelevant parameter in the described action.

The practical implementation of this problem is more complex because a subset of action elements has to be regarded. There are subsets of users, keys, messages which should be presented in action identification.

Description example in fig.1. For action identification and recognition a system of their coding can be exploited. In the convenient approach there is the weight system of coding, e.g. binary, decimal, etc. Binary system is much more extended but it helps to descript particular actions more precisely.

pos	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	...
N^0	*s1*	*s2*	...		*r1*	*r2*	...		K_1	K_2	...		m_1	m_2	...	
1	1	0			1	0			1	0			0	0		
2	1	0			0	0			1	0			0	0		
3	1	0			0	0			0	0			1	0		
lim	2				2				1				2			

Figure 1. The decomposition of action elements into unified subcomponents, N^0 – the number of an operation, *lim* -the limit number of subcomponents in a given type of an action element

Generally, a simple coding system is proposed:

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

where

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

The mutual cooperation among recognizing and attribute correction procedures is illustrated in fig.2.

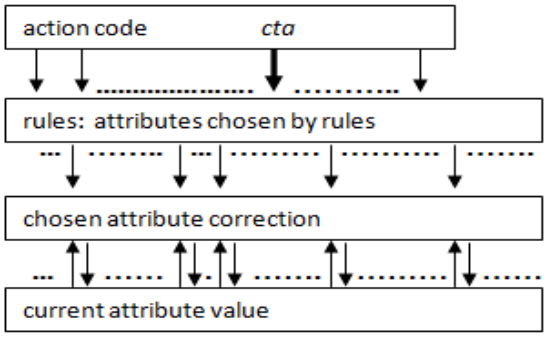

Figure 2. Information flow in a result of action (*cta*) activity. Corrected attributes will become current attributes for the next action

Chosen sets of attributes create security modules that concentrate around main factors like communication protocols, keys, messages service, and users. The idea of attribute allocation is presented in fig.3.

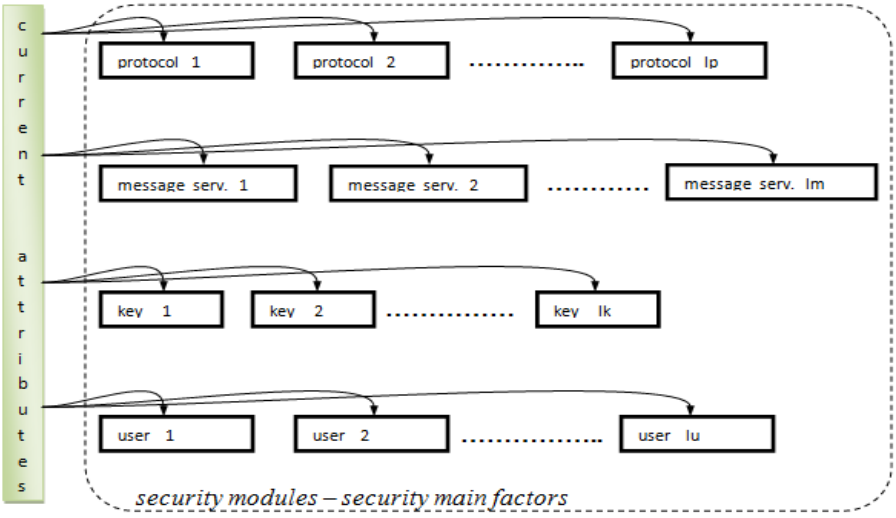

Figure 3. Security modules are built on the basis of chosen attribute sets. Each module (bold frame) consists of a different set of attributes.

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 developing approach several structures of the security main factor (security modules) are proposed (fig.4,5,6,7).

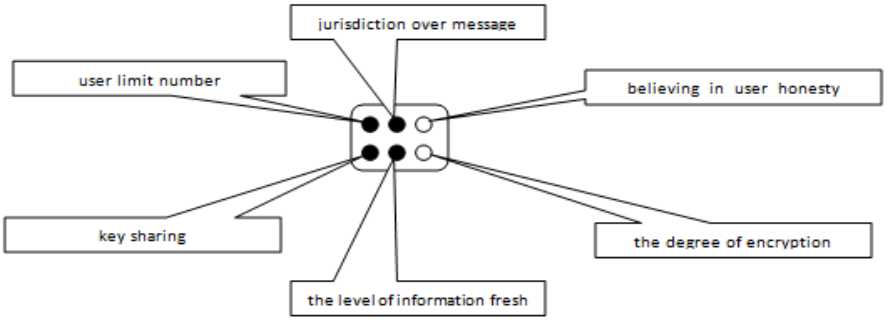

Figure 4. The structure of the protocol security module, black – attribute activation, white – attribute has loosed activity

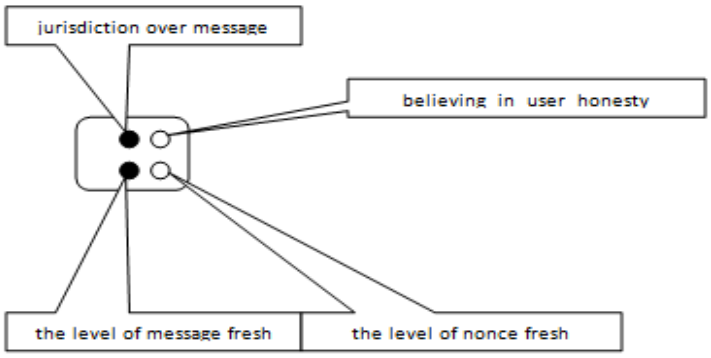

Figure 5. The structure of the message security module

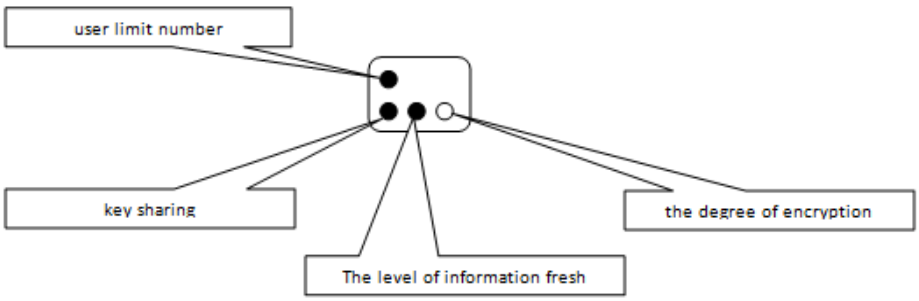

Figure 6. The structure of the key security module

The action influences on attributes are conveniently presented and realized with the usage of equivalent tables. These tables 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. For the description of such situation two handshake operations will be considered and described in fig. 8,9 (column descriptions are adequate to action and attribute definitions).

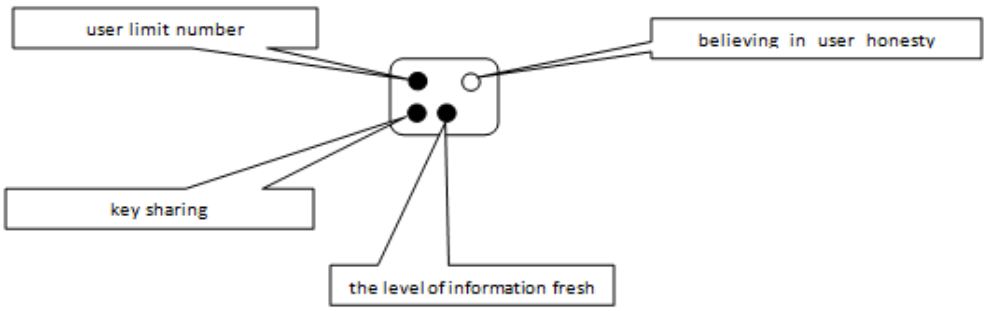

Figure 7. The structure of the user security module

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, *, * , *,* N_a *, 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, *, * , *,* N_b *, nonce is fresh, **}.

The connection between tables (through action numbers) permits the realization of appointed security attribute corrections. The set of attributes is chosen on the basis of rules as well as time and heuristic functions [9]. For the attribute correction the correction coefficients *CC(at(i))* will be used, previously predetermined for each attribute (fig. 10).

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

Figure 8. 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*	*Ks1*	*Ks2*	*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

Figure 9. The appointment of attribute corrections according to BAN rules, where *JMi* – jurisdiction over *i*-th message, *Bhi* –believing in *i*-th user honesty, *NFi* – the freshness of *i*-th nonce, *Dei* – the degree (over one) of *i*-th message encryption, *Ksi* – *i*-th key sharing , *Uli* – the exceeding limit number of users seeing *i*-th message in an encrypted form

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

Figure 10. 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 blockading the correction of the adequate *Ks* attribute (obviously, only in the case when *De>0*), where *den* – the degree of encryption

By continuing the description of the example the initial values of attributes are given. 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) the following levels of attributes are observed (fig.11).

attribute values after 2 communication operations											
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

Figure 11. The states of security attributes after two example operations

Attributes *Nf* and *Ks* are treated as timed attributes. Hence, the following formula is used for their correction: $at^{(k)}(i)=1-e^{k-lf(at(i))}$, where *k* – operation number [10].

By coming to the structure of tokens (binary structure) and the established threshold for all attributes on the same level equal to 0,7, it is possible and interesting to depict a security situation regarding different security modules (main factors) (fig.12).

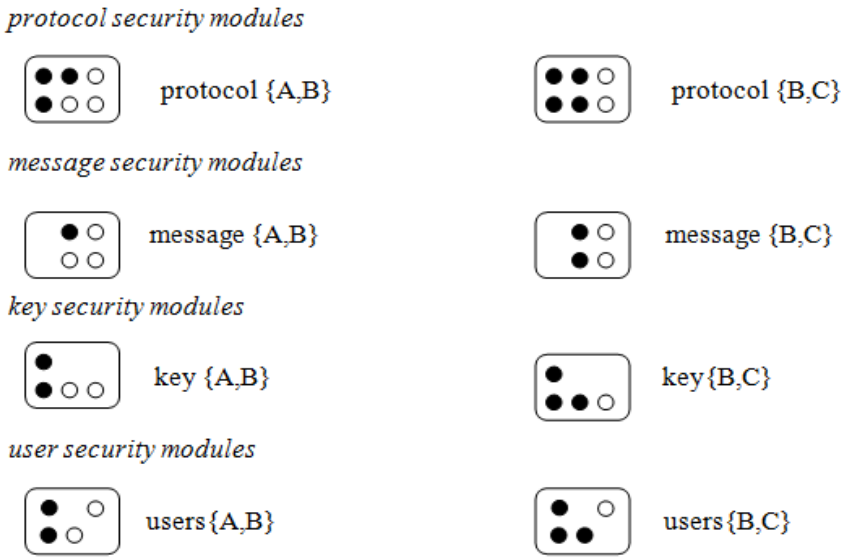

Figure 12. The states of security modules, where *A,B,C* users

By treating all attributes with the same validity it is possible to estimate the level of security for all modules. This problem can be realized by the multiplication of specified component attribute probability values:
SL(protocol{A,B})=0,875*0,64*0,632*1*0,865*1=0,306
SL(protocol{B,C})=0,875*0,64*0,865*1*0,95*1=0,46
SL(message{A,B})=0,875*0,64*0,632=0,354
SL(message{B,C})=0,875*0,64*0,862=0,483
SL(key{A,B})=0,632*1*0,865*1=0,547
SL(key{B,C})= 0,865*1*0,95*1=0,821
SL(users{A,B})=0,64*0,632*0,865*1=0,350
SL(users{B,C})=0,64*0,865*0,955*1=0,404

Similarly, tokens variant can be calculated on the basis of the estimation percent of participation active tokens in the full token set for a given security module (the main security factor). For the results estimated above the security spectrum can be graphically presented.

3. FORMALISMS IN COMMUNICATION SECURITY DESCRIPTION

Communication run is the main object which shall be studied. It has a complex and interleaved structure (combined and interleaved protocol operations) [3].

Definition 2. A tuple (*Sp*, *So*, *Ss*, *nr*), where *Sp* – the set of protocols, *So* – the set of operations, *Ss*- operation sequence (the order in a run), *nr*- the number of operations
is a communication run with the detailed structure:

1. $So_j = \{ac_v, v=1,2,...,na_j\}$ – the operation consists of actions ac_v, na_j – the number of an action in *j-th* operation,
2. $ac_v = \{S, R, K, M, N, Ch, Ad\}$ – an action may contain information about the sender (*S*), receiver (*R*), message(*M*), the character of dealing(*Ch*), additional information - e.g. secrets, etc.(*Ad*).

Definition 3. A tuple **(***At, Th, Tk, na*), where *At* – security attribute set, *Th* – the vector of the low level of feasible attribute values (thresholds), *Tk* – security tokens, *na* – the number of attributes, is the communication security state described as follows:

1. $At = \{at_1, at_2, ..., at_n\} \in [0,1]^n$ – the vector of attribute activation probabilities,
2. $Th = \{th_1, th_2, ..., th_n\} \in [0,1]^n$ – the vector of a threshold attribute activation,
3. $Tk = \{tk_1, tk_2, ..., tk_n\} \in \{0,1\}^n$ – the binary vector of an attribute activation:

if $at_i \geq th_i$ then $tk_i=1$ otherwise $tk_i=0$.

The attributes should be adequate to actions because actions influence them in a direct or indirect way. Therefore, attributes may express assertion about the user honesty belief, belief about message fresh, assertion about attestation, assertion about the shared key, belief that the receiver has jurisdiction over a message, etc.

As a consequence, the action influence on the secure attribute will be considered.

Definition 4. A tuple (*At, Ac, PF, So*), where $PF: Ac \times At \rightarrow [0,1]$ – attribute modification probability matrix *MAPM*, is an action influence for a given operation range *So*. Two types of influence direction structures are possible:

1. $F: ac \rightarrow At$ – modification functions realized from the side of a single action over particular attributes: $f_{i,k}$(ac(k),*at(i)), i=1,2,...,n, k=const* (an action number).
2. $G: Ac \rightarrow at$ – modification functions realized in accordance with one attribute from the set of actions: $f_{i,k}(ac(k)$,at(i)*), k=1,2,...,la, i = const.*

Generally, the following types of influences are usually regarded:

- logic rules based on BAN formalism [7],
- heuristic (experience) rules (regarding neuralgic situations; for example: nonce, message, secret repeating),
- the lifetime of communication attributes,
- the number and character of users (honest or intruder).

The security communication state is described by the set of attributes, whereas its code is determined by the set of adequate tokens.

Definition 5. The security communication state *SCS* is defined by a tuple (*At, t(j), Tk*), where *t(j)* is the moment of *j*-th operation realization (the run current of the clock reading).

$SCS_p = \{at_i, i=1,2,...n\}$,

$SCS_t = \{tk_i, i=1,2,...n\}$,

The code of the current communication state is defined as follows:

$code\_st(t(j)) = CSCS(j) = \sum_{i=1}^{n} 2^{i-1} tk(i)$, where attributes are ascending and ordered according to their validation.

State transition is realized on the basis of the previous state of attribute probability values and their current modifications in accordance with rules:
if $(@_{k=1,2,...,lac(1,j)}\ f_{1,k}(ac(k,j),\text{at}(1)) \geq th_1$ than $tk_1=1$ otherwise $tk_1=0$,
if $(@_{k=1,2,...,lac(2,j)}\ f_{2,k}(ac(k,j),\text{at}(2)) \geq th_2$ than $tk_2=1$ otherwise $tk_2=0$,
…….
if $(@_{k=1,2,...,lac(n,j)}\ f_{n,k}(ac(k,j),\text{at}(n)) \geq th_n$ than $tk_n=1$ otherwise $tk_n=0$,
where
$@f_{i,k}$ - nested modifications $f_{i,k}$ of the *i*-th attribute by all actions in the *j*-th operation;
$@f_{i,k} : \forall_{k=1,2,...,lac(i,j)}\ f_{i,k}(f_{i,k-1})$, $f_{i,0}=at\_init\ (i)$ – the initial attribute state according to the *j*-th operation,
lac(i,j) – the number of actions influencing the *i*-th attribute in the *j*-th operation.

The task of the commutation security investigation in a protocol run is complex, so it is important to build the hierarchic structure of data converting processes [1, 7]. Let us start from extracting the main parameters which are characterized by the slowest, or even zero level, changeability. It could be, for example: a given protocol, the service of a transmitted message, the activity of a concrete user, the exploitation of a given mutual shared key, etc. The assumption about the protocol structure guarantees the multi parameters of the run analysis character. The set of security attributes can be defined for each parameter thereof. It shall be named as security structure.

Definition 6. A set of chosen attributes $Sa(p)=\{at'_{1,p}, at'_{2,p}, ..., at'_{la,p}\}$, where *at'* is the attribute name, creates *p*-th parameter security structure if investigated actions influence its elements in the way determined by a given modification function $f^{(p)}_{i,k}(ac(k),at(i))$, $i=1,2,...,la(p)$, $k=1,2,...,lac(i)$:

1. $Sa(p) \subseteq \{Sa(1) \cup Sa(2) \cup \ldots \cup Sa(lp)\} = \{at'_1, at'_2, ..., at'_n\}$,
2. $Sa(p) \cap Sa(r)$ cannot be equal to zero, $p \neq r$,
3. $f^{(p)}_{i,k}$ cannot be equal to $f^{(r)}_{i,k}$, $p \neq r$,

where
lp – the number of chosen main parameters.
The sense of the modification attribute function may have, in general, two forms:

- modification by multiplication by correcting the coefficient CC,
- modification by exchanging to the current level (represented by the current value of CC).

The differences between $f^{(p)}_{i,k}$ and $f^{(r)}_{i,k}$ (5.3) are inferred from the possibility of $CC(p) \neq CC(r)$. There is a sequence of actions as the ingredients of an interleaved protocol part (the first row). In the second row one may observe attributes describing the security states of the kind of chosen main parameters: $\{Sa(1) \cup Sa(2) \cup \ldots \cup Sa(lp)\}$. The last row denotes the set of fundamental security factors (main components): protocols and keys. An action influencing attributes is realized by the sequenced nested modification function: $@f(^{p)}_{i,k}$ which regards the types of actions and attributes (i.e. their mutual relations) and the character of the main parameter.

The differences among main components appertaining to modification functions characterize:

1. differences among correcting coefficients for different main parameters,
2. the differences of the attribute structure containing for different main parameters.

4. PARALLEL MODELING OF SECURITY COMMUNICATION AUDITING

The communication auditing consists in the chronologic investigation data of run operations and current security analysis according to chosen main components and the indication (or even predicate) of the closest to threaten zone situation. Remarking that the secure structures of a different main factor are independent or complex dependent it is sensible to organize the systems of conversion threads, basing on separated chosen main factors (fig.13,14)

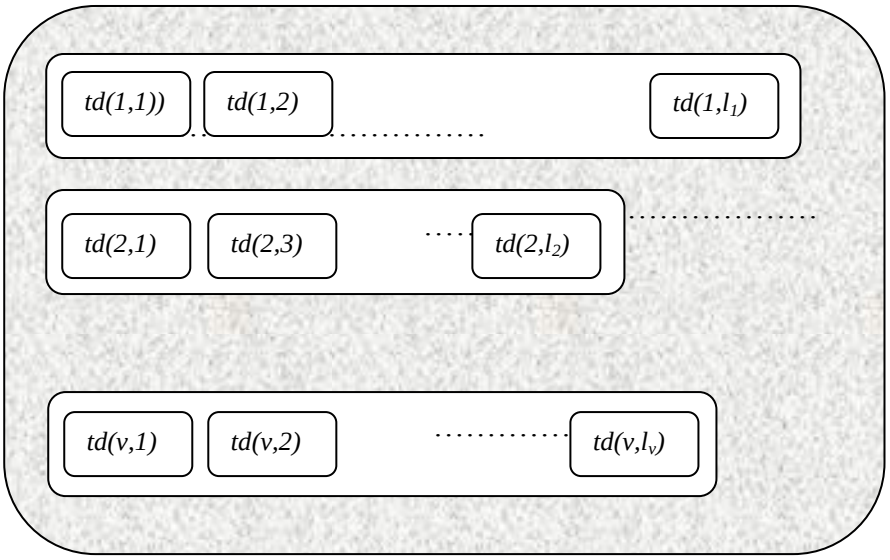

Figure 13. The structure of threads for communication secure investigation , where *td(i,j)* – the thread of *i*-th type and *j*-th structure, l_i – the number of *i*-type thread elements, *v*- the number of thread type (adequate to main security parameters).

The following strategies of parallelization are considered:

a) creation threads according to the type of main secure components,
b) creation threads according to particular threads represented by the elementary part of a component,
c) creation threads according to attributes,
d) creation threads according to particular modification functions.

Counting accelerations in the abovementioned ways of parallelization have the following order of magnitude:

\

a) $acc_a = v$,

b) $acc_b = \sum_{u=1}^{v} l_u$,

c) $acc_c = \sum_{u=1}^{v} l_u la(u)$,

d) $acc_d = \sum_{u=1}^{v} l_u \sum_{i=1}^{la(u)} \sum_{j=1}^{lac(i,v)} 1 = \sum_{u=1}^{v} l_u \sum_{i=1}^{la(u)} lac(i,u)$.

The decreasing complexities in unified threads in the presented approaches of parallelization are equal to: $O(n^5)$, $O(n^4)$, $O(n^3)$, $O(n^2)$, respectively to a), b),c), and d). The smallest thread complexity infers from the structure of attribute modification probability matrix *MAPM* (definition 3). It is possible to assume that the number of main components can dynamically increase in the auditing process (due to more protocols, keys, users, etc.). These situations should be recognized by input procedures and the new threads have to be designated [12]. The data recognition consists in the defining type of new threads, which is connected with the determining sets of attributes and correction coefficients (fig.4).

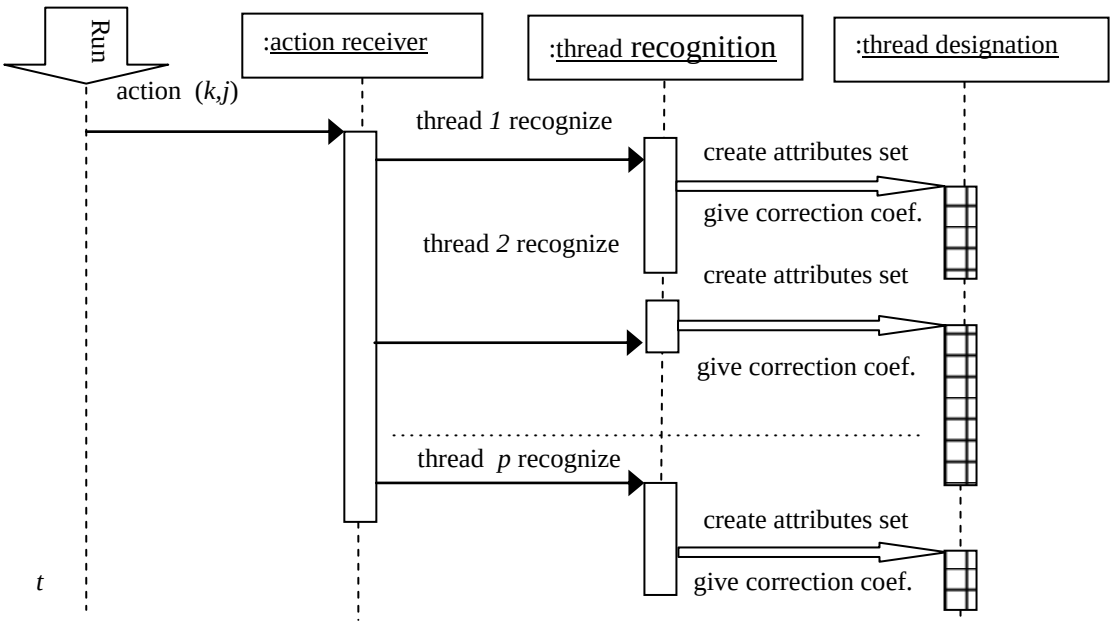

Figure 14. The auditing process diagram regarding the possibility of new thread designation.

The final decision about the communication security level or the threaten situation is represented, respectively, to main factors (components). The full information about it is calculated in the equivalent to chosen factor threads. The security assessment is considered in the detailed (according to particular security attributes) or aggregated form.

Definition 7. A tuple ($At^{(p)}$, $Th^{(p)}$, $la^{(p)}$) , where $At^{(p)}$ - the vector of assigned to p-th security component attributes, $Th^{(p)}$ – the vector of threshold values, $la^{(p)}$ – the number of attributes investigated in the p-th component, is the basis for detailed and aggregated security estimation:

1) a tuple ($At^{(p)}$, $Tk^{(p)}$) is the detailed security estimation,
2) a tuple ($sa^{(p)}$, $st^{(p)}$), where $sa^{(p)}$ – the scale of security, estimated on the basis of attributes in the following way:

$$sa^{(p)} = \frac{1}{la^{(p)}} \sum_{i=1}^{la^{(p)}} ((at^{(p)}(i) - th^{(p)}(i)),$$

$st^{(p)}$ – the scale of security, estimated on the basis of tokens in the following way:

$$st^{(p)} = \frac{1}{la^{(p)}} \sum_{i=1}^{la^{(p)}} (at^{(p)}(i) \geq th^{(p)}(i)),$$ represents detailed security estimation,

is the aggregated security estimation.

The algorithm realized by unified threads can be presented as follows:

1. On the basis of the matrix *MAPM* particular attributes are extracted sequentially for each action with the help of double loops:
   - the outer loop defined the number of an action: *j*,
   - the inner nested loop defined the number of an attribute: *i*.
2. Chosen attributes are checked according to the type of modification on the basis of the following vectors :
   - $Tm(i) = (tm(i,j), j=1,\ldots,lac), tm(i,j)=\{0,1,2\}$, where 0- the code of modification by multiplication, 1 - modification by exchanging, 2 – the code of modification by both mentioned types.
   - $CC(i)= (cc(i,j), j=1,\ldots,lac), cc(i,j)=[0,1]$.
3. A chosen attribute modification procedure.
4. An adequate attribute token correction (definitions 3,4).
5. The security assessment in accordance with definition 7.
6. The auxiliary analysis taking into account the estimation of probability transition to next states and the level of convergence to the threaten zone.
7. The calculation of the code of a real new state (definition 3,4,5).
8. Optionally, the prognostic analysis.

Utilized and useless threads will be removed from the system or exchanged onto a new one after fulfilling specific conditions, such as: zero security token levels, exceeding the given lifetime of the main factor.

5. CONCLUSION

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 creation 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 threatens connected with different protocol realizations.
and possible applications.

REFERENCES

1. Alur R., Courcoubetis C., Dill D. L., 1992, *Verifying Automata Specifications of Probabilistic Real-Time Systems. Real-Time*, Theory in Practice, Springer LNCS, 600, pp 28-44.
2. Alur R., Dill D. L., 1994, *A Theory of Timed Automata*, Theoretical Computer Science, 126, pp 183-235.
3. Beauquier D., 2003, *On Probabilistic Timed Automata*, Theoretical Computer Science, 292, pp 65-84.
4. Burrows M., Abadi M., Needham R., 2004, *A Logic of Authentication, Robert Harpe Logics and Languages for Security*, Approximate Non-Interference Journal of Computer Security, 12, pp 37-82.
5. Evans N., Schneider S, 2000, *Analysing Time Dependent Security Properties in CSP Using PVS*, Proc. of Symp. on Research in Computer Security, Springer LNCS, 1895, pp 222-237.
6. Focardi R., Gorrieri R., 1995, *A Classification of Security Properties*, Journal of Computer Security, 3, pp 5-33, 1995.
7. Focardi R., Gorrieri R., Martinelli F., 2000, *Information Flow Analysis in a Discrete -Time Process*, Algebra. Proc. of 13th CSFW, IEEE CS Press, pp 170-184.
8. Gray III J. W.,1992, *Toward a Mathematical Foundation for Information Flow Security*, Journal of Computer Security, 1, pp 255-294.
9. Kwiatkowska M., Norman G., Segala R., Sproston J., 2002, *Automatic Verification of Real-time Systems with Discrete Probability Distribution*, Theoretical Computer Science, 282, pp 101-150.
10. Kwiatkowska M., Norman G., Sproston J.,2003, *Symbolic Model Checking of Probabilistic Timed Automata Using Backwards Reachability*. Tech. rep. CSR-03-10, University of Birmingham.
11. Szpyrka M., 2004, *Fast and exible modeling of real-time systems with RTCP- nets*, Computer Science, pp 81-94.
12. Tadeusiewicz R., 2011, *Introduction to Inteligent Systems*, chapter No 1 in book: Wilamowski B.M., Irvin J.D.(Eds.): The Industrial Electronic Handbook, CRC Press, Boca Raton, pp 1-12.
13. Tadeusiewicz R., 2010, Place and role of Intelligence Systems in Computer Science, Computer Methodsin Material Science, Vol.10, No 4, pp 193-206.
14. Tudruj M., Masko L. 2005, *Toward Massively Parallel Computation based on Dynamic Clasters with Communication on the Fly*, IS on Parallel and Distributed Computing, Lille, France, pp 155-162.