Int. J. Appl. Math. Comput. Sci., 2015, Vol. 25, No. 2, 403–414
DOI: 10.1515/amcs-2015-0031

amcs

PROBABILITY TIMED AUTOMATA FOR INVESTIGATING COMMUNICATION PROCESSES

HENRYK PIECH [a,*], GRZEGORZ GRODZKI [a]

[a] Institute of Computer and Information Sciences
Częstochowa University of Technology, ul. Dąbrowskiego 73, 42-201 Częstochowa, Poland
e-mail: {henryk.piech,grzegorz.grodzki}@icis.pcz.pl

Exploitation characteristics behaves as a decreasing valors factor (DVF) which can be connected with degradation processes. It is a structure that consists of independent attributes which represent situations generally connected with a given exploitation factor. The multi-attribute structure contains attributes directly and indirectly referring to the main factor. Attribute states, by definition, can only maintain or decrease their values. Such situations are met in security, reliability, exploitation, fatigues and many complex one-directed or irreversible processes. The main goal refers to protocol security analysis during the realization of the communication run that specifies the assessment of the level of current and oncoming threats connected with cryptography authentication. In the communication run, the operations of different protocols mutually interleave. Our concept is based on the algorithm of attributes correction during exploitation process realization (Blanchet *et al.*, 2008). The monitoring and correcting procedures make it possible to provide forecast information about possible threats on the basis of the structure of the current attribute values.

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

1. Introduction

Probabilistic timed automata (PTA) have various forms depending on the feasibility of state achievements (Luu *et al.*, 2012; Lanotte *et al.*, 2010). These specific conditions control the modeling process, but in general probability and time are the main parameters which influence a new state transition (Luu *et al.*, 2012). The character of influences can have a directed or an undirected form. The undirected actions are realized by ingredient components of automata node modifications. This strategy is realized in our proposal concerning the automata node structure (Lanotte *et al.*, 2010). In this case, we consider a complex node structure that contains chosen attributes describing investigated phenomena.

An additional assumption regards the irreversible character of attribute (or security, reliability, etc.) exploitation. A complex irreversible process can be a threat in accordance with several aspects. For example, communication protocol security can be analyzed according to the authentication of users, nonce freshness, the time of a message and key exploitation, etc. Such an approach supplies both detailed and global information about authentication, confidentiality and jurisdiction over users and messages. The probabilities of state transitions are defined on the basis of a protocol operation structure and a system of rules (Burrows *et al.*, 1990; Dechesne and Wang, 2010) activating particular security attributes. The analysis usually starts from a single communication protocol referring to two or three users as well as one message supplemented by its confirmation and additional user parameters, i.e., nonces (Burrows *et al.*, 1990; Sun *et al.*, 2013). This situation may be extended by a set of messages, a group of users, and auxiliary parameters. In this case, we will consider several levels of estimation of communication security (individually to interleaved protocols in a network run). However, the main approach remains the same (Basin *et al.*, 2011). Timed attributes will be corrected by a specific coefficient dependent on a predefined lifetime parameter (Basagiannis *et al.*, 2010; Xiong *et al.*, 2012; Lindell and Pinkas, 2009).

The probability measurable space is convenient in terms of attribute presentation. The essential task consists in the elaboration of a simple form of an attribute structure and its modification during the realization of an operation.

*Corresponding author

In order to describe the communication process, other authors sometimes use binary decision diagrams (BDDs) (Gosti *et al*., 2007), and in the case of a security problem they exploit Petri nets (Gu and Dong, 2005). The system works on-line as an auditing and warning tool. This gives us rich information about the detailed and general communication state of security. We also propose the models of security changing based on PTA for various structures of communication runs. The originality of our approach consists in the exploitation of communication logics (BAN, Horae, PCL) and the elaboration strategy of defining the security characteristics in detailed and global aspects according to the analysis of protocol runs. In past concepts, such a combination of mathematical and heuristic methods has not been observed. In previous algorithms, only reliability channel statistic characteristics were analyzed (Kwiatkowska *et al*., 2004).

2. Organization of irreversible process investigation

We decided to present a proposal for an automaton with respect to irreversible process modeling of communication security changes. It starts with a communication run of interleaving protocols. The state transition is connected inside the protocol reading operation and the automaton node change (Lindell and Pinkas, 2009; Ciobâcă *et al*., 2012).

Definition 1. The *state (node) transition* $tr(i)$ is defined to be the quintuple (i, Ac, At, F, BF), where i is the number of an automaton (PTA) node adequate to the number of operations in a run, Ac is the set of actions in a current operation, At is the set of node attributes, $F : Ac \rightarrow At$ is the set of transition functions, BF is the set of low and up bounds of feasible levels of security attributes (data for constraints and node codification).

Transition functions are a complex of nested functions. Firstly, we describe the character of functions. They will be logic security rules *fr* defined by Burrows *et al*. (1990) as well as McIver and Morgan (2011), heuristic rules *fh* concerning multi-times that use the same type of communication parameters (e.g., nonce) (Ciobâcă *et al*., 2012), the time of an activation attribute *ft*, the number and character of user influences *fu*. Accordingly, generally, we can use the following notation: $F : fu(ft(fh(fr(Ac_i, At_{i-1}))))$. A set of actions should be structured by particular action codes. Describing operations as a sequence of activated actions in each operation is very convenient. Actions are arguments of the transition function F.

Definition 2. The *operation* consists of actions presented in the form of the octuple $(Se, Re, U, CM, CN, K, D, Sc)$, where: Se is the code of a sender, Re is the code of a receiver, U is the set of users, CM is the code of a message, CN is the code of a nonce, K is the set of keys shared by sender and receiver, D is the degree of encryption, Sc is the code of a secret.

Obviously, a set of actions can be expanded and the operation structure will then be rebuilt. Look at arguments of logic rules represented by the function *fr*:

- $P| \equiv X$: P believes X,
- $Q \triangleright X$: Q sees X (Q has received X),
- $P \triangleleft X$: P said X (P has sent X),
- $P| \Rightarrow X$: P controls X, (P asserts that X is right, P has jurisdiction over X),
- $\rightarrow^K P$: P has K as its public key,
- $P \leftrightarrow^K Q$: P and Q share K as a public key,
- $\#(X)$: message X is fresh,
- X_K: message X is encrypted by the key K,
- $\langle X \rangle_Y$: message X with an attached secret Y.

The transformation $Tri : Ac_i \rightarrow Cr_i \rightarrow At_i$ of the current i-th operation actions into the security attribute activation (connected with the possibility of their correction) is realized in the form of rules by means of the table of equivalent correction coefficients, where Cr_i is the set of attribute correction coefficients.

3. Data presentation in a formal logic description

The initial, intermediate and final results will be presented in a binary or probabilistic form so that we can define classes of their representation. Let us start with actions and attributes. Actions are part of protocol operations whereas attributes are secure characteristics. Hence, the class of actions is a set of their binary representation according to the protocol operation content. Attributes have the same means as actions and can be represented in both the binary and probabilistic forms, depending only on the modeling stage (Lindell and Pinkas, 2009). Thus, in the first case, we can denote by R the class of the set of A ($Ac \vee At$) actions or attributes:

$$R = \{A\}^{2m} = \{A_0, A_1, \ldots, A_m\}^{2m},$$

where m is the maximal range of the action (attribute) number.

Different binary combination subsets of A can be defined: $A = \{\cup A_i = \{*\}, i = 0, 1, \ldots, m\}$, where$\{*\}$ signifies $\{0 \vee (0, 1]\}$.

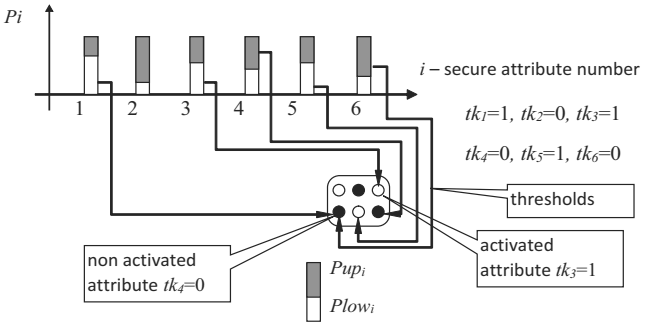


Fig. 1. Attribute binarization represented by tokens tk.

We use a simplified binary notation: $\{tk_i = 0\} \Rightarrow \{tk_i = " - "\} \Rightarrow \{tk_i = " "\}$, and $Tk = \{tk_1, -, tk_3, -, -\} = \{tk_1, tk_3\}$, which is illustrated in Fig. 1.

The main problem consists in the elaboration of a method of changing values of the probability of security attributes: $prob(A_j)$. The fourth type of attribute influences will be regarded in the secure communication analysis:

- logic rules based on BAN formalism (Burrows *et al*., 1990),
- heuristic (experience) rules,
- the lifetime of communication attributes,
- the number and character of users.

All these factors are exploited with respect to the creation of correction coefficients in order to modify the probability of the secure attribute in every protocol operation. Two kinds of corrections will be used depending on the type of attribute and factor:

- transformation on the basis of the previous value (adequate to the previous state) of attribute probability,
- exchange of the previous attribute probability for a new correcting coefficient.

The characteristics are as follows:

- user honesty,
- assertion about belief,
- belief about message freshness,
- assertion about attestation,
- assertion about shared keys, secrets
- belief that receiver has jurisdiction over message, etc.

The state structure proposal is presented in Fig. 2.

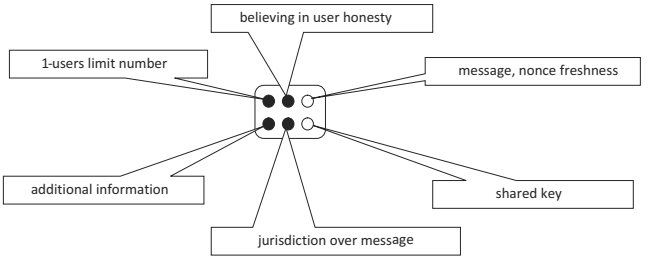


Fig. 2. Structure of the security state: black—attribute activation, white—attribute has loose activity.

4. Type of security attribute modification during the realization of a communication run

In practice, the type of influences is represented by two forms of algorithm attribute corrections:

(i) $mc = 0, 1$: correction by multiplication by a given updating coefficient MCC when the influence of logic and heuristic rules may be observed, $mc = 1$: the activation of this attribute correction form, $mc = 0$: the rejection of this correction form.

(ii) $ec = 0, 1$: correction by exchanging to the current level (represented by the current coefficient value of ECC) when the influences of lifetime or users (intruders) may be observed.

This situation can be illustrated as in Fig. 2. Therefore, it is possible to simultaneously use two forms of correction for a single attribute. For that reason, if $ec = 1$, then the attribute value has not increased:

$$\begin{aligned} at_{t=k+1}(i) &\xrightarrow{mc=0,ec=0} at_{t=k}(i), \\ at_{t=k+1}(i) &\xrightarrow{mc=1,ec=0} at_{t=k}(i) \cdot MCC, \\ at_{t=k+1}(i) &\xrightarrow{mc=0,ec=1} ECC, \\ at_{t=k+1}(i) &\xrightarrow{mc=1,ec=1} \min\{at_{t=k}(i) \cdot MCC, ECC\}. \end{aligned}$$

The experiments have proven that heuristic rule influences (for example, with the multi usage of the same nonce) are more effective in specific cases when correction is realized in the following way:

$$at_{t=k+1}(i) \xrightarrow{mhc=1,ehc=0} at_{t=k}(i) \cdot (1 - MCC),$$

or

$$at_{t=k+1}(i) \xrightarrow{mhc=1,ehc=0} at_{t=k}(i) \cdot (1 - at_{t=k}(i)).$$

The actual value of ECC with a lifetime influence will be counted using the formula

$$ECC = 1 - e^{t_j - lt_i},$$

where t_i is the time of attribute activation, lt_i is the attribute lifetime.

In reality, the time activity is transformed into the probability attribute value, according to a given attribute lifetime (Fig. 3).

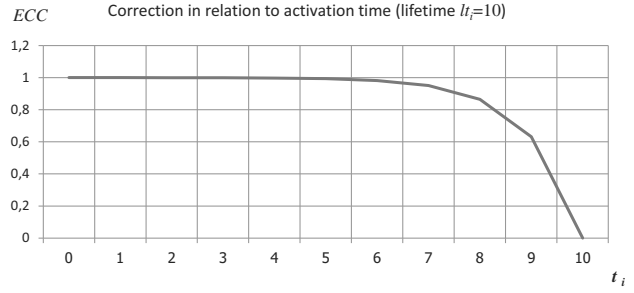


Fig. 3. Action time influence on an attribute.

The actual value of ECC, with additional user (intruder) influence, will be counted by the formula: $ECC =$ if$(nus < nht)$ then $ECC = 1$ else $ECC = e^{nht-nus}$ where nus is the number of users (in the environment of the main security factor), nht is the number of honest users (in the environment of the main security factor).

In reality, the time activity is transformed into the probability attribute value, according to a given number of honest users (Fig. 4).

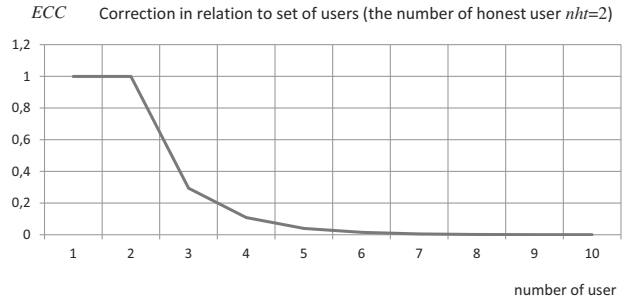


Fig. 4. Action user influence on an attribute.

Let us introduce the set of describing input variables: $c(j,k), cc(j,i), mc(i), mhc(i), ec(i), ehc(i)$, and index limits: nf is the number of main factors $k = 1, 2, \ldots, nf$, nna is the length of a multirun; the number of all actions in a network: $j = 1, 2, \ldots, nna$, nat is the number of attributes (it is assumed that structures of security nodes for all main factors are the same: $i = 1, 2, \ldots, nat$. Obviously, this is not necessary; in such a case, we will use $i(k) = 1, \ldots, nat(k)$, $fat(k,i)$ is the matrix of the attribute structure of main factors, $mhc(i) = \{0,1\}$ is the activation of the heuristic rule influence, $ehc(i) = \{0,1\}$ is the activation of the dishonest user influence, mcc is the correcting coefficient value, mhc is the correcting coefficient value, $t(i)$ is the time of i-th attribute activation, $lt(i)$ is the lifetime of i-th attribute $(ec(i) = 1)$, nus is the number of users, nht is the number of honest users, $w(i)$ is the weight of an attribute according to communication security.

The rate of attribute correction may be adjusted by the scaling parameter alpha: $ECC = 1 - e^{\alpha(t_j - lt_i)}$, (see Fig. 5), $ECC = e^{\alpha'(nht-nus)}$.

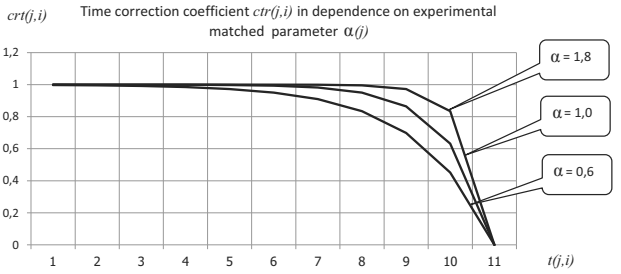


Fig. 5. Practical method of time security attribute correction.

5. Probability timed automata as the communication security investigation model

We propose to use PTA (or convert them to colored Petri nets) as the main tools for the investigation of communication security according to selected main factors, such as protocols, users, keys, messages, etc. The nodes presented in Fig. 2 will be the fundamental part of PTA (and Petri nets). Introduce the definition of the security state which will correspond to the automaton node.

Definition 3. The quadruple (At, Th, Tk, na), where At is a security attribute set, Th is the vector of the low level of feasible attribute values (thresholds), Tk stands for security tokens, na is the number of attributes, is a *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 threshold attribute activation (acceptation),
3. $Tk = \{tk_1, tk_2, tk_n\} \in \{0,1\}^n$: the binary vector of attribute activation: if $at_i \geq th_i$, then $tk_i = 1$; otherwise, $tk_i = 0$.

The global structure of these automata is presented in Fig. 6.

If any attribute is decreased to an unacceptable level, then there is no possibility to improve its value and security features cannot be increased. To present the time parameter with an intrinsic characteristic (according to the security aspect), we propose the following definition.

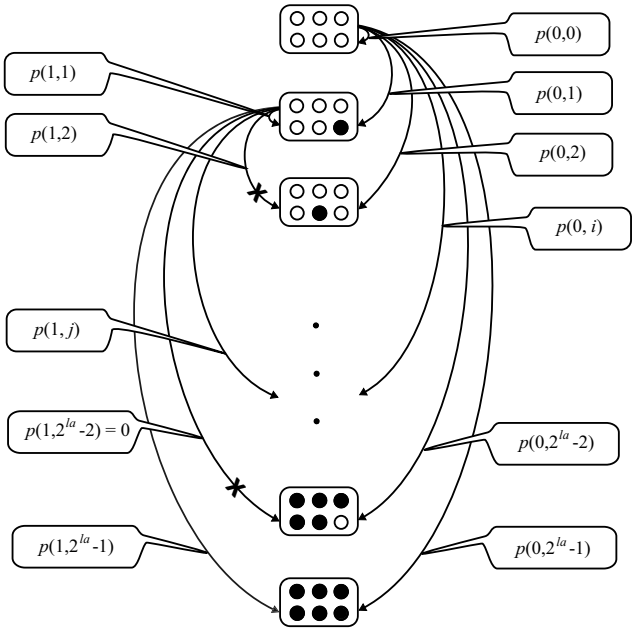


Fig. 6. Scheme of a probability timed automaton for the investigation of communication security, where $p(i, j)$ is the probability of state changing from state i to j, $j \geq i$.

Definition 4. The *probabilistic timed automaton PTA* is the sextuple 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, where $\sum_c \in \sum$ are declared as being currently possible; the function inv: $L \rightarrow CC(X)$ 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 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 (i) making a *discrete transition* or (ii) letting *time pass*. In Case (i), a discrete transition can be made according to any probabilistic edge $(l, g, \sigma, p*) \in p$ with an *enabled* source location l; 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 $p \cdot (X, l'')$. In Case (ii), the option of letting time pass is available only if the invariant condition $inv(l)$ is satisfied while time elapses and when an enabled probabilistic edge with a current step does not exist. Note that a timed one (Luu *et al.*, 2012) is a probabilistic timed automaton for which every probabilistic edge (l, g, σ, p^*) is such that $p^* = \mu(X, l'')$ (the distribution point assigning probability 1 to (X, l'') for some $(X, l'') \in 2^X \times L$).

6. Exploitation of communication logic rules for security attribute modification

The security structure of a state (node) is created according to a chosen attribute that plays the role of the main criterion. By using communication logic rules (Burrows *et al.*, 1990) and exploiting a protocol operation action as arguments, it is possible to define attributes that will be corrected. It helps to describe the decision part of the assessment table (Fig. 7).

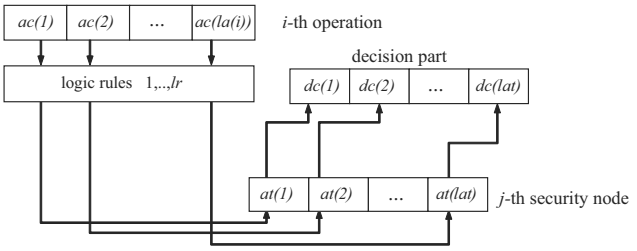


Fig. 7. Effect of the exploitation of communication rules, where lr is the number of rules.

Let us introduce the notion of a general effect that pertains to the activation of communication logic rules.

Definition 5. The *general effect of rule activation* is the quintuple $GERA = \{Pr, Dc, SR, Mac, Mat\}$, where Pr is the profile vector, Dc is the decision vector, SR is the the set of communication logic rules, Mac is the action mark vector (action engaging), Mat is the attribute mark vector (the proposal of security attribute correction), such that

(i)

$$Pr\,(j) = \bigcup_{k=1}^{lac} \{ac\,(k, j) : mac\,(k, j) = 1\},$$

where k is the action number, j is the code of the current communication operation, $ac(k, j)$ is the k-th action of the j-th operation, $mac(k, j) = 1$ if the k-th action includes the j-th operation, otherwise $mac(k, j) = 0$, the mark of an action, lac is the number of all possible actions;

(ii) if

$$\exists\,(r \in SR)\,\{\forall s = 1, \ldots, lcon\,(r) \exists k = 1, \ldots, lac : (con\,(s, r) = ac\,(k, j)) \wedge (mac\,(k, j) = 1)\},$$

then

$$\forall p = 1, \ldots, lcon\,(r)\ \exists i = 1, \ldots, lat : at\,(i) = conc\,(p, r) \wedge (mat\,(i, j) = 1),$$

where r is the rule code, s is the number of a condition in a rule, $con(s, r)$ is the s-th condition in the r-th rule, $lcon(r)$ is the number of conditions in the r-th rule, p

is the number of conclusions in a rule, $conc(p,r)$ is the p-th conclusion in r-th rule, $lconc(r)$ is the number of conclusions in the r-th rule, i is the attribute number, lat stands for the number of attributes, $mat(i,j) \in \{0,1\}$ is the proposal for the i-th security attribute correction after the j-th operation. This notation means that if a rule exists, in which all conditions are fulfilled by current operation actions $((con(s,r) = ac(k,j)) \wedge (mac(k,j) = 1))$, then all attributes appointed by these rule conclusions should be corrected $(mat(i,j) = 1)$;

(iii)

$$Dc\,(j) = \bigcup_{r=1}^{SR} \bigcup_{i=1}^{lat} \{at\,(i) : mat\,(i,j) = 1\}.$$

The effect of one communication logic rule activation will be obviously defined in the same way if this rule is extracted from the rule set: $R = SR$. By studying rules proposed in the literature (Burrows *et al.*, 1990), it is possible to group them into sets with reference to authentication, nonce, jurisdiction, vision, freshness and transitivity (Fig. 8). These groups can be treated as kinds of rules. The rule activity RA takes place when all given rule conditions are fulfilled:

$$\forall s = 1,\ldots,lcon\,(r) \exists k = 1,\ldots,lac : \\ (con\ \ (s,r) = ac\,(k,j)) \wedge (mac\,(k,j) = 1)\,.$$

The set of operation rules $SAR(j)$, activated by the j-th one, is defined as follows:

$$SAR\,(j) = \bigcup_{\substack{R \in SR \\ r \in cSR}} \{R : RA\,(r,j) = 1\}$$

where r is the code of rule R.

The security module SAM consists of chosen security attributes. Firstly, in order to realize this choosing process, we have to define the main factor, such as the user, the key, the message, the secret, etc. The decision about the i-th security attribute correction after the j-th operation is defined as follows: If

$$\exists r \in SR : RA\,(r,j) = 1 \exists p \in \{1,\ldots,lat\,(r)\} : \\ (at\,(i) = conc\,(p,r)) \wedge (at\,(i) \in SAM)$$

then

$$mat\,(i,j) = 1.$$

7. Searching for a useful form of communication forecast

First of all, we approve the decision about the sensibility of the preparation concerning longtime and short time forecasts. This prognosis refers to the main security

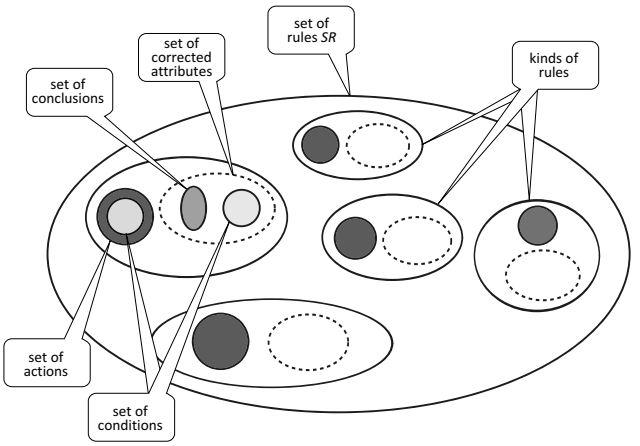


Fig. 8. Structure of communication logic rules.

factors (analyzed as security modules). It may have a general and detailed character. Its creation is based on the current attribute values $at(i)$, threshold attribute levels $th(i)$ (minimal accepted attribute value), the probability of attribute corrections and the attribute structure of a given security module $sm(k) = atp(1,k), atp(2,k), \ldots, atp(lat,k)$, where $atp(i,k) = 0,1$ stands for the binary participation index referring to the i-th attribute in the k-th security module structure.

It is useful to introduce the following types of prognosis:

- detailed, referring to security attributes,
- module, referring to security modules,
- general, referring to all or chosen sets of security modules.

Another prognosis classification refers to the way of probability estimation of attribute corrections. In this case, we propose the following classification structure:

- with intruders,
- without intruders.

In both of these estimation variants, we may use a different approach (Fig. 9):

- according to past communication operations,
- according to predicted future operations regarding the structure of the communication protocol.

The full analysis of the prognosis is realized on the basis of probability and binary variables.

The analyzing system will define and predict the threat zone on the basis of $pp(i)(pf(i))$. This zone is expressed in time or probability according to single attributes, single modules or a set of modules (Fig. 10).

A more detailed presentation requires formal definitions and security prognosis grammar fixing.

Table 1. Matrix transformation MCC: empty fields conventionally contain irrelevant values, i.e., 1 or 0.

Attribute description		Action description—characteristics										
code	attributes	server	sender	receiver	intruder	key	message	nonce	character	freshness	jurisdiction	secret
1	users	1	1	1	1				1			
2	believing in honesty	1	1	1		0	1	1	1			
3	freshness		1	1	0	1	1	1	1			
4	shared key		1	1	0	1	1	1	1			
5	jurisdiction	1	1	1		1	1	1	1		0	
6	additional information		1	1		1	1	1	1			1

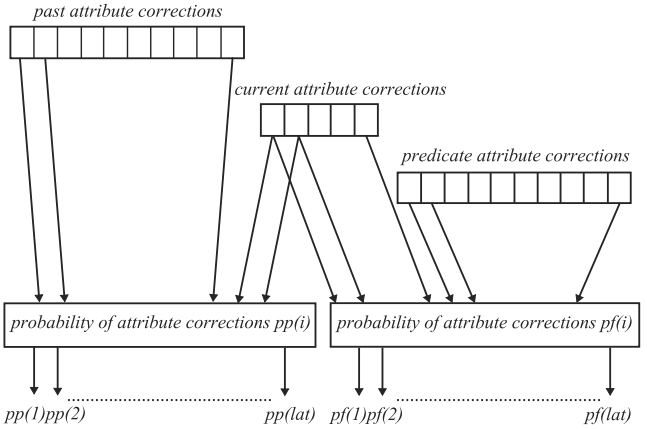


Fig. 9. Dependency diagram of estimation procedures pertaining to the attribute correction probability.

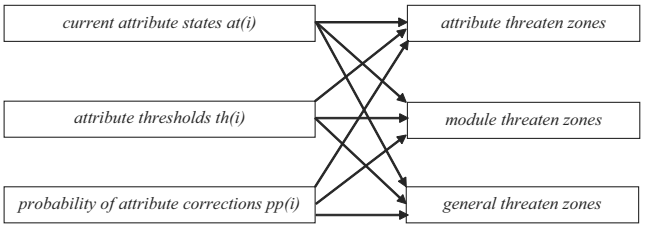


Fig. 10. Diagram depicting different types of threat zone creation.

8. Description of the implementation approach

We propose a system that activates security attribute correction on the basis of rules introduced by Barrows, Abadi, Needham and their communication logic BAN. It will be used as one of the procedures that exploits the structure (in the simplest variant matrix structure) for the conversion of protocol actions into attribute modification: $Ac \subseteq Op(i) \subseteq Pr(cp) \sim At(cp)$, where Ac is the set of actions participating in the i-th operation, $Op(i)$ is the i-th protocol (in a communication run) operation, $Pr(cp)$ is the code of investigated protocol determinates according to its type and users, $At(cp)$ is the set of attributes representing the protocol cp, '$\sim$' is the activation of an attribute correction process.

Generally, one may use the system of BAN rules but in the implementation that infers mechanisms contained in this system it can be reduced to logic reducts, which excluded the redundancy of logic transformation. For example, the system of a logic reduct can be presented in a matrix form $MAA[m \times n]$, where m is the number of attributes, n is the number of action characteristics (Table 1).

How can we exploit this matrix? Firstly, the protocol profile (PP) should be prepared according to action characteristics (Table 2). The matrix of the protocol profile consists of vectors of characteristics that are adequate to protocol operations. Secondly, simple logic formulas are used for checking the consistency of characteristics between MCC and PP:

1. If $(pp(server) + pp(sender) + pp(receiver)) = 1$ then $pp(intruder) = 1$; it is interpreted in the following way: if the second honesty belief does not appear among users, then intruder interference is possible.
2. If $pp(intruder) = 1$ then $at(users) \xrightarrow{ec=1} at'(users)$.
3. If $pp(key) = 0$ then $at(believing\ in\ honesty) \xrightarrow{mc=1} at'(believing\ in\ honesty)$.
4. If
$$\left(\sum_{t=1}^{Tc} pp^{(t)}\,(message) > 1\right)$$
or
$$\left(\sum_{t=1}^{Tc} pp^{(t)}\,(nonce) > 1\right),$$
then
$$at(believing\ in\ honesty) \xrightarrow{mc=1} at'(believing\ in\ honesty),$$
where t is the number of protocol operations, Tc stands for currently realized protocol operations, irrelevant profile parameter (equals to $(1 \vee 0)$—empty fields in matrix PP) do not contribute to the sum; it is interpreted in the following way: if the same message or the same nonce appears for the second time, then the belief in honesty is reduced.

5. If
$$\left(\sum_{t=1}^{Tc} pp^{(t)}\,(message) > 1\right)$$
or
$$\left(\sum_{t=1}^{Tc} pp^{(t)}\,(nonce) > 1\right),$$
then $at(freshness) \xrightarrow{ec=1} at'(freshness)$; it is interpreted in the following way: if the same message or the same nonce appears for the second time, then freshness is decreased.
6. If $((pp(message) = 1$ or $pp(nonce) = 1)$ and $(pp(key) = 0)$ or $pp(intruder) = 1)$ then $at(sharedkey) \xrightarrow{mc=1} at'(sharedkey)$; it is interpreted in the following way: if a message or nonce appear and a key is not shared or an intruder activates, then the shared key attribute looses part of its value.
7. If $(pp(key) = 1$ and $(pp(message) = 1$ or $pp(nonce) = 1))$ and $(pp(server) + pp(sender) + pp(receiver) = 1))$ or $pp(key) = 0$ then $at(jurisdiction) \xrightarrow{mc=1} at'(jurisdiction)$; it is interpreted in the following way: if a key is shared and information (message or nonce) is exchanged behind the server, then the jurisdiction level of a sender and receiver over this information will be decreased.
8. If $pp(intruder) = 1$ and $pp(secret) = 0$, then $at(additionalinf.) \xrightarrow{mc=1} at'(additionalinf.)$; it is interpreted in the following way: if an intruder activates and a secret is not attached, then the confidence level is decreased.

These rules were created on the basis of a probabilistic and intuitive approach. Therefore, attribute corrections can be also inferred from the levels of suspicion, believing in honesty and the right character of communication actions. Obviously, the set of rules may be enriched by new proposals for additional reasons; for example, the appearance of a new form of communication attacks. Moreover, we may also change numbers of action characteristics and security attributes.

The above-mentioned rules (IR) infer from BAN rules but are adapted to algorithm implementation. This approach permits us to choose an attribute for correction and define the way of its modification (mc, ec strategies). At the same time, we have a possibility to analyze the security of both the single protocol and the run of protocols (the interleaving parts of protocols). The second case refers to the dynamic analysis during the realization of a communication run. The information flow (rules 4 and 5) makes practical sense because the accumulated parameter values essentially decrease the security level of appointed attributes.

Table 3. Example of a communication run.

Protocol of Andrew RPC/1			
A→B:id(A).Na	B→A:<Na.K'AB>KAB		
Protocol of Woo and Lama/1			
I(A)→D:C	D→I(C):Nd	I(C)→D:Nd	D→I(S):<C.Nd>KDS
Protocol of Andrew RPC/2			
A→B:<Na>K'AB	B→A:<Na.K'AB>KAB		
Protocol of Nesset—supplemented			
B→E:<Nb>Ke	E→B:<Ne,KEB>KE^{-1}	B→E:<N'b>KEB	
Protocol of Woo and Lama/2			
I(S)→D:<C.Nd>KDS			

9. Algorithm and results of investigating communication security

The algorithm consists of the following stages:

1. Stencil input reading of the elements of the matrix MCC.
2. Reading and recognizing the current operation in a communication run.
3. The transformation of an operation run into a protocol profile vector (an adequate row in the matrix PP).
4. Switching on clocks and reading activity time parameters of attributes dependent on lifetime characteristics.
5. The accumulation of information connected with messages and nonces (due to the content of rules 4 and 5).
6. The exploitation of rules in order to select corrected attributes.
7. The realization of the correction procedure.
8. The output of the current security state, the values of security attributes for all protocols.
9. Optionally, the creation of a threat prognosis for protocols according to the given main security factors.
10. Threat prognosis output.
11. If the communication run continues, then go to Step 2.
12. Additional analysis, e.g., with respect to the comparison of protocol securities.

The description of a communication run contains interleaving parts of protocols. Such an example may be presented in the following form.

The realization of the run from Table 3 is described in accordance with parts of interleaving protocols: Andrew RPC: part 1 (2 operations), Woo and Lama—part 1 (4 operations), Andrew RPC—part 2 (2 operations), Nessett—part 1 (‘3 full operations), Woo and Lama—part 2 (1 operation). There are 4 protocols and 12 operations. For each protocol (treated as main security factors), we

Table 2. Description of protocol actions (i.e., matrices of the protocol profile PP).

Andrew Secure Handshake protocol											
	server	sender	receiver	intruder	key	message	nonce	character	freshness	jurisdiction	secret
A→B:A,<Na>KAB		1	1		1		1	1	1	1	
B→A:<Na+1,Nb>KAB		1	1		1		1	1	1	1	
A→B:A,<Nb+1>KAB		1	1		1		1	1	0	1	
B→A:<K'AB,N'b>KAB		1	1		1	1	1	1	1	1	

Needam Shroeder protocol											
	server	sender	receiver	intruder	key	message	nonce	character	freshness	jurisdiction	secret
A→B:A,B	1	1						1	1	1	
S→A:<KB,B>Ks(-1)	1		1		1	1		1	0	1	
A→B:<Na,A>KB		1	1		1	1	1	1	0	1	
B→S:B,A	1	1						1	0	1	
S→B:<KA,A>Ks(-1)	1		1		1	1		1	0	1	
B→A:<Na,Nb>KA		1	1		1		1	1	0	1	
A→B:<Nb>KB		1	1		1		1	1	0	1	1

Nesset protocol											
	server	sender	receiver	intruder	key	message	nonce	character	freshness	jurisdiction	secret
A→B:<Na,KAB>KA(-1)		1	1			1	1	1	1	0	
B→A:<Nb>KAB		1	1		1		1	1	1	1	

Nesset protocol—supplemented											
	server	sender	receiver	intruder	key	message	nonce	character	freshness	jurisdiction	secret
B→A:<Nb>KA		1	1		1		1	1	1	1	
A→B:<Na,KAB>KA(-1)		1	1			1	1	1	1	1	
B→A:<N'b>KAB		1	1		1		1	1	1	1	

Woo and Lama protocol											
	server	sender	receiver	intruder	key	message	nonce	character	freshness	jurisdiction	secret
A→B:id(A)		1	1					0	1	1	
B→A:Nb		1	1				1	0	1	0	
A→B:<Nb>KAS		1	1		1		1	1	0	1	
B→S:<id(A).<Nb>KAS>KBS	1	1		1	1	1	1	1	0	1	
S→B:<id(A).Nb>KBS	1		1	1	1	1	1	1	0	1	

Woo and Lama protocol—id(A) caught by intruder I											
	server	sender	receiver	intruder	key	message	nonce	character	freshness	jurisdiction	secret
I(A)→B:A			1	1					1	0	
B→I(A):Nb		1		1			1		1	0	
I(A)→B:Nb			1	1			1	1	0	0	
B→I(S):<A.Nb>KBS		1		1	1	1	1	1	0	0	
I(S)→B:<A.Nb>KBS			1	1	1	1	1	1	0	0	

Andrew RPC protocol											
	server	sender	receiver	intruder	key	message	nonce	character	freshness	jurisdiction	secret
A→B:id(A).Na		1	1				1		1	0	
B→A:<Na.K'AB>KAB		1	1		1	1	1	1	0	1	
A→B:<Na>K'AB		1	1		1		1	1	0	1	
B→A:Nb		1	1				1		1	0	

Attack on protocol RPC											
	server	sender	receiver	intruder	key	message	nonce	character	freshness	jurisdiction	secret
A→I(B):id(A).Na		1			1		1		1	0	
I(B)→A:id(B).Na			1		1		1		1	0	
A→I(B):<Na.K'AB>KAB		1			1	1	1	1	0	0	
I(B)→A:<Na.K'AB>KAB			1		1	1	1	1	0	0	
A→I(B):<Na>K'AB		1			1	1	1	1	0	0	
I(B)→A:Ni			1		1		1		1	0	
A→I(B):N'a		1			1		1		1	0	

Low protocol											
	server	sender	receiver	intruder	key	message	nonce	character	freshness	jurisdiction	secret
A→B:id(A).Na		1	1				1		1	0	
B→A:<Na.K'AB.id(B)>KAB		1	1		1	1	1	1	0	1	
A→B:<Na>K'AB		1	1		1		1	1	0	1	
B→A:Na		1	1				1		0	0	

determine security attributes for modification (Table 4). Before starting the main part of our investigation, we should prepare parameters connected with the correction process (correction coefficients, lifetime and alpha parameters) and security threshold values for all attributes (Table 5). An alpha parameter plays the role

Table 4. Description of a communication run (code description in accordance with Fig. 1).

N	operation	protocol users	codes of chosen attribute	type of modification
1	A→B:id(A).Na	A,B	2,4	mc=1
2	B→A:<Na.K'AB>KAB	A,B	2,3	mc=1
3	I(A)→D:C	C,D,I	1,2,4,5	mc=1, ec=1
4	D→I(C):Nd	C,D,I	1,2,4,5	mc=1, ec=1
5	I(C)→D:Nd	C,D,I	1,2,3,4,5	mc=1, ec=1
6	D→I(S):<C.Nd>KDS	C,D,I	1,2,3,4,5	ec=1, mc=1
7	A→B:<Na.K'AB>	A,B	2,3	mc=1, ec=1
8	B→A:<Na.K'AB>KAB	A,B	2,3,4,5	mc=1, ec=1
9	B→E:<Nb>KE	B,E	-	-
10	E→B:<Ne,KEB>KE^{-1}	B,E	4,5	mc=1, ec=1
11	B→E:<N'b>KEB	B,E	-	-
12	I(S)→D:<C.Nd>KDS	C,D,I	1,2,3,4,5	mc=1, ec=1

Table 5. Experimentally adapted exploitation parameters.

Additional parameters						
type of modification	users	believing in honesty	freshness	shared key	jurisdiction	additional information
correction	–	0,7	–	0,5	0,8	0,8
alpha	1	–	0,6	–	–	–
thresholds	0,6	0,6	0,6	0,6	0,6	0,6

of a scaling factor. All parameters are obtained as a result of experiments regarding the specific character of communication protocols subjected to the analysis.

The investigation is realized in real time according to the communication run process (Tables 3 and 4). The results of the investigation are presented in Fig. 11 and Tables 6 and 7.

The graphical result presentation is depicted in Table 7, in relation to particular protocols.

The modeling process of security level changes is connected with the determination of transition state probabilities and time parameters for activated attributes. The set of clocks is adequate to a set of attributes whose level depends on time. For example, there are attributes such as a shared key, the freshness of messages and nonces, etc. Therefore, the clock is switched on when rules indicate a given attribute for the first time in order to correct it. The evaluation of

Table 6. Results of attribute value correction during the realization of a run.

operation N	users	believing in honesty	freshness	shared key	jurisdiction	additional information
1	1	0,7	1	0,5	1	1
2	1	0,63	0,9918	0,5	1	1
3	0,3679	0,7	1	0,5	0,8	1
4	0,3679	0,63	1,0000	0,485	0,64	1
5	0,3679	0,567	0,9918	0,4705	0,512	1
6	0,3679	0,5103	0,9850	0,4563	0,4096	1
7	1	0,567	0,8347	0,5	1	1
8	1	0,5103	0,6988	0,5	1	1
9	1	1	1	1	1	1
10	1	1	1	0,97	0,5	1
11	1	1	1	0,97	0,5	1
12	0,3679	0,4593	0,4512	0,4426	0,32768	1

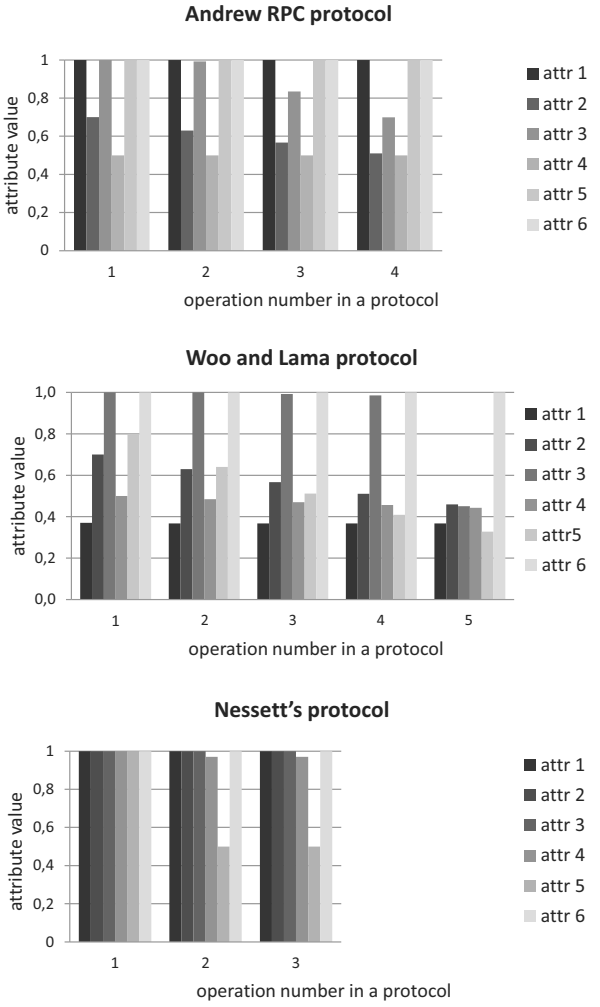


Fig. 11. Diagrams depicting the dependency attribute value during the realization of an operation sequence.

Table 7. Changes concerning the activation of security attributes for all protocols participating in a communication run.

Results (Andrew RPC protocol)						
type of modification	users	believing in honesty	freshness	shared key	jurisdiction	additional information
activation	1	1	1	0	1	1
Results (Woo and Lama protocol)						
type of modification	users	believing in honesty	freshness	shared key	jurisdiction	additional information
activation	0	0	0	0	0	1
Results (Nessett's protocol)						
type of modification	users	believing in honesty	freshness	shared key	jurisdiction	additional information
activation	1	1	1	1	0	1

probability concerning the state transition is a more complex problem. Theoretically, the transition probability $p(c, j)$ (where c is the code of a current state, j is

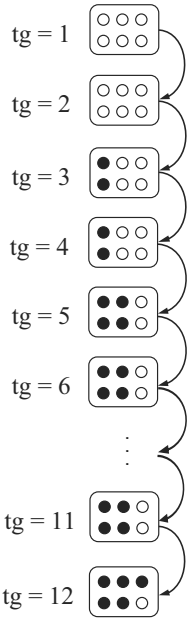


Fig. 12. Diagram of a state transition in PTA in accordance with the realization of protocol operations ($tg = 1, \ldots, 12$). The locations of security attributes are adequate to the node structure in Table 2.

the one of feasible states, i.e., reachable) depends on distances between the current value of an attribute ati and a given security threshold th_i. In other words, the analysis permits us to catch dependencies among the transition state probability and action distribution in protocols that are contained in an investigated communication run. This problem is presented in our project and articles associated with it. Here we present only the diagram of PTA functioning in accordance with the protocol operation sequence (Fig. 12). Our approach provides a large set of information about communication mutually involving security parameters (and characteristics). It infers from the condition and conclusions of communication rules being the main part of logic grammars. The functional communication model, based on tested and verified probability timed automata and colored Petri net conventions, was included in our and other (Kwiatkowska *et al*., 2004) algorithm strategies. As in both solutions, our algorithm similarly works in an on-line regime and generates characteristics and conclusions inferring from them in 1–9 sec.

10. Conclusion

The proposed automaton has specific constraints according to the degradation (irreversible) character of the modeled phenomena. Such a kind of phenomena is very often encountered during the exploitation of nature and human dealings. The features of specific kinds of PTA permit us to dynamically investigate depreciation processes and to create forecasts about approaching threats (Pironti *et al*., 2012). The heuristic character of the algorithm requires the preparation of sets of correction parameters but, on the other hand, allows us to obtain a specialized form of implementation (it makes calculations faster). In this automaton we also converted timed characteristics to probability.

This type of approach leads to the creation of a uniform structure of system (based on PTA) organization. When comparing the proposed system with existing ones, modeled on the probabilistic timed strategy, it should be noted that our variant exploits the authentication logic and provides prognosis according to different security modules (protocols). Other systems only investigate statistic channel parameters or confirm threats *ex post*. The main advantage of the proposed algorithm and its implementation, in comparison with other solutions (Kwiatkowska *et al*., 2003), consists in exploiting a new kind of logic (authentication logic) with respect to security analysis, but not with reference to statistic and reliability analysis. The auditing strategy is realized with similar time parameter effectiveness. The system will be enriched by recommendation procedures, which will be realized in the case of oncoming threats.

References

Basagiannis, S., Katsaros, P. and Pombortsis, A. (2010). An intruder model with message inspection for model checking security protocols, *Computers and Security* **29**(1): 16–34.

Basin, D., Caleiro, C., Ramos, J. and Viganò, L. (2011). Distributed temporal logic for the analysis of security protocol models, *Theoretical Computer Science* **412**(31): 4007–4043.

Blanchet, B., Abadi, M. and Fournet, C. (2008). Automated verification of selected equivalences for security protocols, *Journal of Logic and Algebraic Programming* **75**(1): 3–51.

Burrows, M., Abadi, M. and Needham, R. (1990). A logic of authentication, *ACM Transactions on Computer Systems* **8**(1): 18–36, DOI: 10.1145/77648.77649.

Ciobâcă, S., Delaune, S. and Kremer, S. (2012). Computing knowledge in security protocols under convergent equational theories, *Journal of Automated Reasoning* **48**(2): 219–262.

Dechesne, F. and Wang, Y. (2010). To know or not to know: Epistemic approaches to security protocol verification, *Synthese* **177**(1): 51–76.

Gosti, W., Villa, T., Saldanha, A. and Sangiovanni-Vincentelli, A. (2007). FSM encoding for BDD representations, *International Journal of Applied Mathematics and Computer Science* **17**(1): 113–128, DOI: 10.2478/v10006-007-0011-6.

Gu, T. and Dong, R. (2005). A novel continuous model to approximate time Petri nets: Modelling and analysis, *In-*

ternational Journal of Applied Mathematics and Computer Science **15**(1): 141–150.

Kwiatkowska, M., Norman, G. and Parker, D. (2004). PRISM 2.0: A tool for probabilistic model checking, *1st International Conference on Quantitative Evaluation of Systems (QEST'04), Enschede, The Netherlands*, pp. 322–323.

Kwiatkowska, M., Norman, G., Parker, D. and Sproston, J. (2003). Performance analysis of probabilistic timed automata using digital clocks, *in* K. Larsen and P. Niebert (Eds.), *Formal Modeling and Analysis of Timed Systems (FORMATS'03)*, Lecture Notes in Computer Science, Vol. 2791, Springer-Verlag, Berlin/Heidelberg, pp. 105–120.

Lanotte, R., Maggiolo-Schettini, A. and Troina, A. (2010). Weak bisimulation for probabilistic timed automata, *Theoretical Computer Science* **411**(50): 4291–4322.

Lindell, Y. and Pinkas, B. (2009). A proof of security of Yao's protocol for two-party computation, *Journal of Cryptology* **22**(2): 161–188.

Luu, A., Sun, J., Liu, Y., Dong, J., Li, X. and Quan, T. (2012). SeVe: Automatic tool for verification of security protocols, *Frontiers of Computer Science* **6**(1): 57–75.

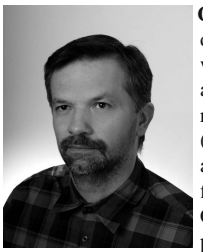
McIver, A. and Morgan, C. (2011). Compositional refinement in agent-based security protocols, *Formal Aspects of Computing* **23**(6): 711–737.

Pironti, A., Pozza, D. and Sisto, R. (2012). Formally based semi-automatic implementation of an open security protocol, *Journal of Systems and Software* **85**(4): 835–849.

Sun, H., Wen, Q., Zhang, H. and Jin, Z. (2013). A novel pairing-free certificateless authenticated key agreement protocol with provable security, *Frontiers of Computer Science* **7**(4): 544–557.

Xiong, L., Xiong, Y., Ma, J. and Wang, W. (2012). An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards, *Journal of Network and Computer Applications* **35**(2): 763–769.

Henryk Piech received an M.Sc.E.E. in informatics from St Petersburg Transport Institute in 1976, and his Ph.D. and habilitation from St Petersburg Electronic Institute in 1983 and 1990, respectively. He has developed stochastic processors and their applications. His research interests include optimization of organization strategy and computer conversion. He is also the author of about 200 publications in international journals and 12 monographs.



Grzegorz Grodzki received his M.Sc. and Ph.D. degrees in mechanic science from the Częstochowa University of Technology, Poland, in 1992 and 2000, respectively. At the time he conducted research on measurements with the use of LDA (laser Doppler anemometry), and digital analysis and processing of randomly sampled signals. He first worked at the Institute of Mathematics and Computer Science. Currently, he is an assistant professor at the Institute of Computer and Information Sciences, Częstochowa University of Technology. He is an author of over 15 papers in refereed journal and conference papers. His current interests include security informatics systems and computer networks, as well as optimization of organization strategies.

Received: 26 November 2013
Revised: 24 April 2014