Communication Security Prognosis Realized as the Parallel Dynamic Auditing Intelligent System

Henryk Piech, Grzegorz Grodzki, and Piotr Borowik

Czestochowa University of Technology,
Dabrowskiego 73, 42201 Czestochowa, Poland
h.piech@adm.pcz.czest.pl, ggrodzki@icis.pcz.pl, piotrborowikii@gmail.com

Abstract. Communication operations are realized according to cryptography protocols in a typical network. Such communication among users takes place on the basis of public keys, secrets, supplies, encrypted messages and nonces. The investigation of the communication run gives security information about forthcoming threats. The main goal of the research consists in the elaboration of a useful and simple (in the sense of complexity) prognosis algorithm adapted to an auditing form of investigation. The results of the prognosis presented in the time parameter about impending threats are dynamically changed, operation by operation. Therefore, users can prepare a strategy of avoiding the closest (in the sense of time) or the most dangerous threat. The proposed approach is based on probability counting rules that guarantee fast realization at the cost of accuracy. The large scale of parallelization possibilities is also worth noticing. This follows from the independent module structure of fundamental security elements which are associated with dynamically designated counting threads.

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

1 Introduction

The auditing investigation program is treated as an intelligent system [1], [2] for dynamic generation of warnings about possible forthcoming threats. There are many approaches to elaborate convenient and useful information about the security of communication protocols [3]. Usually, proposed solutions refer to a single protocol [4] and independently give short deterministic or approximate information about the security of a protocol during the moment of realization [5]. Therefore, they have static character but it is obvious that network communication, in practice, consists of the operations of interleaving protocols. Consequently, the full process is realized on time [6], [7]. As a result our proposition concerning the algorithm tries to regard the practice situation. Such convention does not decrease the number of possibilities regarded in the exploitation of timed automata TA, probability timed automata or Petri nets [8],[9],[10]. On the contrary, in this

G. Jezic et al. (eds.), *Agent and Multi-Agent Systems: Technologies and Applications*,
Advances in Intelligent Systems and Computing 296,
DOI: 10.1007/978-3-319-07650-8_15, © Springer International Publishing Switzerland 2014

case possibilities that appear pertain to the creation of a complex structure of automaton nodes consisting of a set of security attributes. The auxiliary task consists in finding the connections among actions from protocol operations and predefined security attributes. To resolve this problem, we exploit the system of communication logic rules [11]. Actions in these rules are treated as a condition and security attributes are treated as conclusions. Therefore, the set of actions leads to choosing attributes, which will be corrected. Then the next problem appears: how to define methods and evaluate parameters for correction procedures. This problem has been resolved on the basis of experiments. We use two methods of attribute correction [12] based on the attribute value exchange (for time and the number of user attribute parameters) and on attribute multiplication by the correction coefficient (for probabilistic attribute parameters).

2 The Structure of Security Auditing Intelligent Systems in Reference to Parallel Realization Predispositions

Let us not forget the communication security investigation algorithm based on the probability timed automaton (PTA) model [13], [14] or on an adequate colored Petri net (CPN) structure [10]. The general configuration is presented in Fig.1. This algorithm characterized a multi stage structure and a simple conversion in particular stages. Up till now, almost all stages are described in detail and implemented. The largest complexity characterizes the procedure that combines the stages of rule activation and attribute correction. The complexity of this procedure is equal to $O(n^5)$. These parameters are only established from nested conversion cycles. As it is depicted in the algorithm block scheme, the possibility concerning the designation of a parallel thread appears only after the recognition of actions [15]. On the other hand, the observation of network communication may be divided into parts (segments). This creates added possibilities regarding parallelization. Generally, let us not forget that threads described and defined security situations with reference to keys, messages, users, secrets, etc. (the so called main security factors). Simplified theoretical analysis permits us to estimate the acceleration of calculations due to the parallel realization that is presented in the following way:

$$Accel = \frac{lo \cdot la\,(2lr \cdot lat + 2lm + lm \cdot lpr)}{lo\,(lat + lr \cdot lat + 2lat + lpr)} = \frac{la\,(2lr \cdot lat + 2lm + lm \cdot lpr)}{lat + lr \cdot lat + 2lat + lpr} \qquad (1)$$

where:
lo - the number of operations in a run,
la - the number of actions,
lat - the number of security attributes,
lr - the number of communication rules,
lm - the number of security modules (the main security factors),
lpr - he number of generated types of prognosis.

In practice, the acceleration parameter is approximately equal to $0,51 \cdot Accel$ on average. Example 1 illustrates the change of the acceleration parameter for the following data: lo = 30 (irrelevant parameter due to (1)), la=8, lat =[4,8] lr =[8,12], lm =3, lpr =2 (Fig.2). Example 2 illustrates the change of the acceleration parameter for the following data: lo = 30 (irrelevant parameter due to (1)), la=[8,10], lat =[4,8] lr =10, lm =3, lpr =2 (Fig.3) Example 3 illustrates the change of the acceleration parameter for the following data: lo = 30 (irrelevant parameter due to (1)), la=8, lat =[4,8], lr =10, lm =[2,4], lpr =2 (Fig.4).

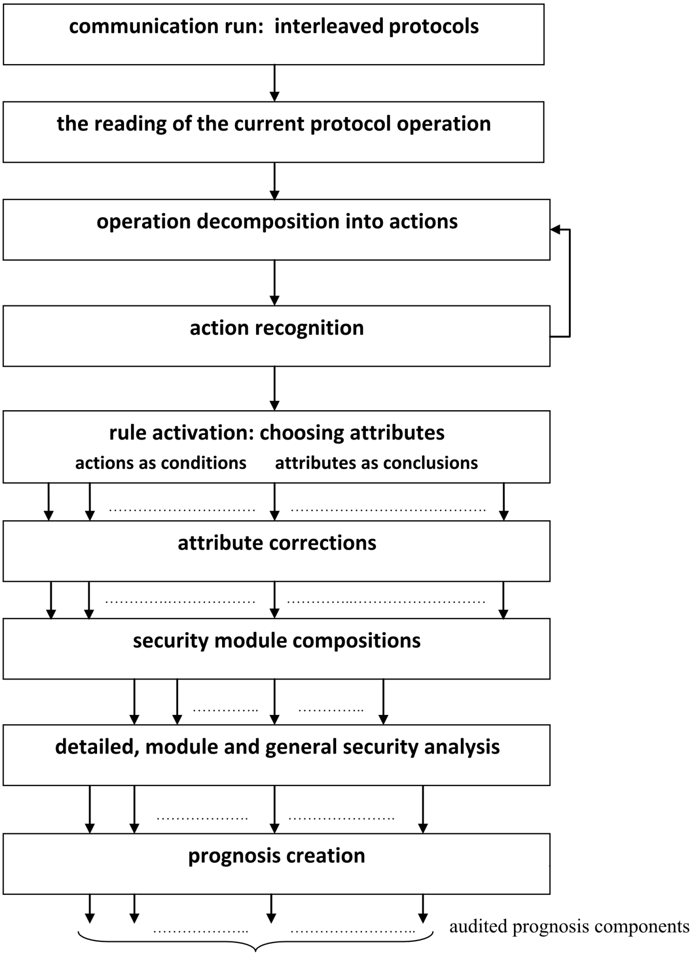
communication run: interleaved protocols
the reading of the current protocol operation
operation decomposition into actions
action recognition
rule activation: choosing attributes
actions as conditions
attributes as conclusions
attribute corrections
security module compositions
detailed, module and general security analysis
prognosis creation
audited prognosis components

Fig. 1. The block scheme of the security investigation algorithm

The nonlinear and linear dependencies reflect the simple form of conversion and the easiness of parallelization regarding calculation that generally consists in thread decomposition. Each thread may be connected with: protocols, messages, actions, rules, users and will be subjected to parallel execution. These elements

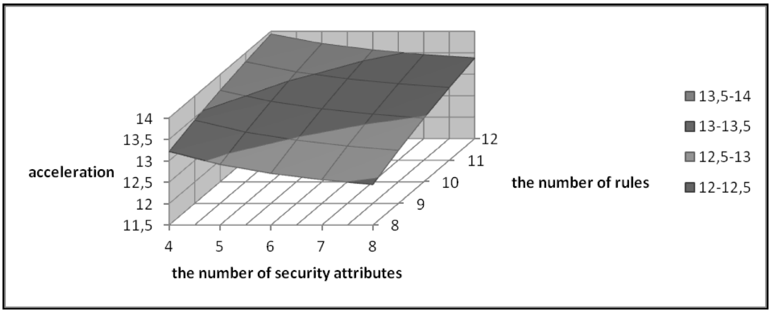

Fig. 2. Acceleration dependence deriving from the number of security attributes and the number of rules (example 1)

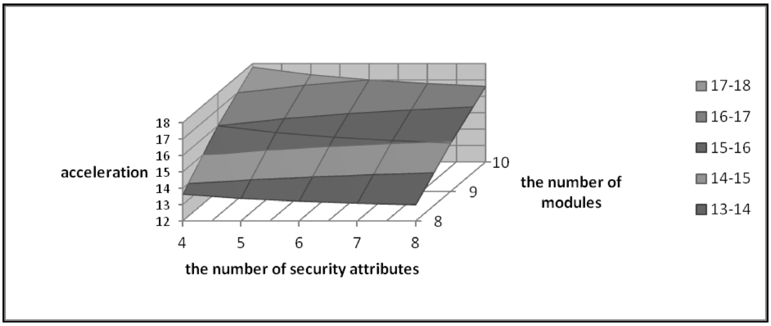

Fig. 3. Acceleration dependence deriving from the number of security attributes and the number of modules (example 2)

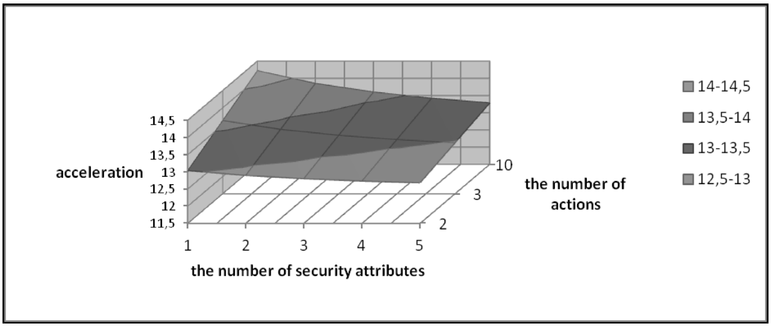

Fig. 4. Acceleration dependence deriving from the number of security attributes and the number of actions (example 3)

create security modules. Parallelization possibilities still appear after operation reading. In the designated thread, we may recognize one kind of action, and we should also exploit several threads if *la* actions are searched up. When the full set of actions recognizes the next group of threads it is activated for choosing security attributes, which will be corrected on the basis of rules. In this case we exploit lr threads. Attribute corrections should be sequentially realized due to a specific action depending on the character of attribute influence. The calculation parallelization possibilities reappear after attribute modifications. It is connected with creation security modules. Only then, we may start the prognosis analysis.

3 Searching for the Useful Communication Prognosis Form

Firstly, we approve the decision concerning the sensibility of longtime and short-time prognosis preparation. This prognosis refers to main security factors (analyzed as security modules). The prognosis may have a general and a detailed character. The prognosis creation is based on current attribute values $at(i)$, threshold attribute levels $th(i)$ (the minimal accepted attribute value), the probability of attribute corrections and the attribute structure of a given security module $sm(k) = atp(1,k), atp(2,k), ..., atp(lat,k)$, where $atp(i,k) = 0, 1$ - 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 regarding 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.5):

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

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

The analyzing system will define and predicate the threat zone (expressed in time or probability) on the basis of $pp(i)$ $(pf(i))$, according to single attributes, single modules or the set of modules (Fig.6).

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

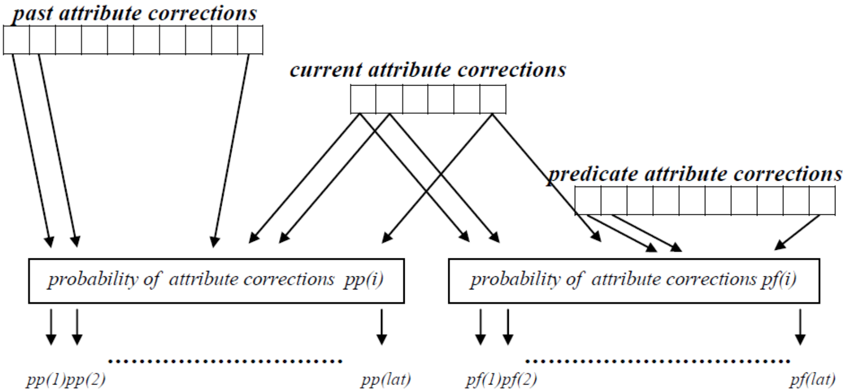
past attribute corrections
current attribute corrections
predicate attribute corrections
probability of attribute corrections pp(i)
probability of attribute corrections pf(i)
pp(1)pp(2)
pp(lat)
pf(1)pf(2)
pf(lat)

Fig. 5. The dependence diagram about attribute correction connected with probability estimation procedures

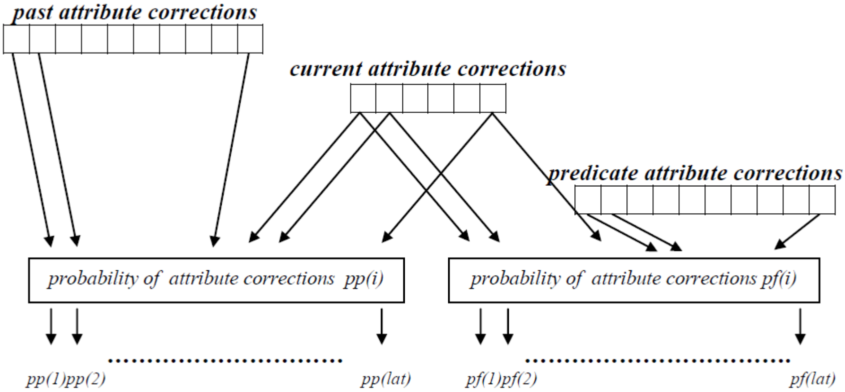
past attribute corrections
current attribute corrections
predicate attribute corrections
probability of attribute corrections pp(i)
probability of attribute corrections pf(i)
pp(1)pp(2)
pp(lat)
pf(1)pf(2)
pf(lat)

Fig. 6. Diagram regarding a different type of threat zone creation

4 Security Prognosis Formalisms and Evaluation System

Let us start from atomic security elements, i.e. security attributes. The security zone and the threat zone are defined as follows:

Definition 1. *Security attribute zone is the difference between the current attribute value and the threshold attribute value $sz(i) = at(i) - th(i)$. The threat attribute zone is the difference between the threshold attribute value and the current attribute value $fz(i) = th(i) - at(i)$.*

The next definition is connected with the so called tokens, which appoint the attribute security state.

Definition 2. *Token delivers information about the location of the current attribute value in relation to the security threshold (attribute state): $tk(i) = 1$, if $at(i) \geq th(i)$, $tk(i) = 0$, otherwise.*

Definition 3. *Security module state is defined by the model set of the attribute state and the security token state: $ams(k) = pat(1,k) \cdot pat(2,k) \cdot ... \cdot pat(lat,k)$, where:*

$p(i,k)$- probability security $factor$ referring to the k-th model,
$pat(i,k) = at(i)$ if $atp(i,k) = 1$,
$pat(i,k) = 1$ if $atp(i,k) = 0$,

and

$tms(k) = btk(1,k) \cdot btk(2,k) \cdot ... \cdot btk(lat,k)$,

where:

$btk(1,k)$ - binary security factor referring to the k-th model,
$btk(i,k) = tk(i)$ if $atp(i,k) = 1$,
$btk(i,k) = 1$ if $atp(i,k) = 0$.

Definition 4. *Security i-th attribute (k-th module) approbation is connected with fulfilling the condition tk(i)=1 (tms(k)=1).*

Definition 5. *Attribute security spectrum is defined by the sets of $at(1), at(2), ..., at(lat)$ and $tk(1), tk(2), ..., tk(lat)$.*
K-th module security spectrum is defined by the sets of $at(1), at(2), ..., at(lat)$, $tk(1), tk(2), ..., tk(lat)$ and $atp(1,k), atp(2,k), ..., atp(lat)$.

Now, the protocol structure will be considered according to attribute engaging. The communication protocol consists of operations which activate actions and those lead to chosen (by rules) attribute corrections. Therefore, it is sensible to define the number of given attribute corrections in the investigated protocol.

The protocol structure indicates the set of corrected attributes. The approach of these attributes cannot be ordered in our prognosis. The distribution of attribute corrections is the only relevant element. For each standard protocol, the number of attribute corrections should be predefined, e.g. in the following form: for protocol Pi distribution of attribute corrections:
$dpa(i) = nat(1,i), nat(2,i), ..., nat(lat,i)$,

At the beginning of the prognosis analysis, a problem appears consisting in small quantity of data. Generally, the situation, which takes place after the part concerning the communication run reading, can be depicted as in Fig.7.

Both the set of corrected and the set of not corrected attributes (black rings and white rings in Fig.7) can be exploited for the prognosis creation:

$$nca\,(i) = \sum_{j=1}^{k} cat\,(i, c\,(j)) \tag{2}$$

$$nnca\,(i) = \sum_{j=1}^{k} nat\,(i, c\,(j)) - cat\,(i, c\,(j)) \tag{3}$$

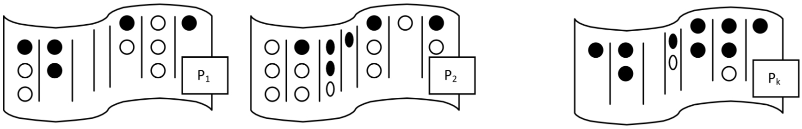

Fig. 7. The situation after the start of k protocol reading and the set of attribute corrections (black rings). For different protocols we define different distributions $dpa(i) \neq dpa(j)$.

where:
i - attribute number (code),
j - activated protocol number,
$c(j)$ - the code of the j-th activated protocol,
k - the current number of activated protocols,
$nca(i)$ - the current number of corrected attributes in all recognized protocols,
$cat(i,j)$ - the current number of corrected attributes in the j-th protocol,
$nnca(i)$ - the current number of not corrected attributes.

Obviously, the kind of protocols, which appear in the next part of the communication run, is unknown. Therefore, only activated protocols will be regarded. We propose two variants of prognosis, according to data exploitation:

- *ex post*, where the probability of attribute corrections $pexp(i)$ is calculated on the basis of an action that takes place after reading up the run operation:

$$pexp\,(1) = \frac{nca\,(1)}{\sum_{j=1}^{k} nat\,(i, c\,(j))} \tag{4}$$

- *ex ante*, where the probability of attribute corrections $pexa(i)$ is calculated on the basis of an action in the future realization of protocols activated till now:

$$pexa\,(1) = \frac{nnca\,(1)}{\sum_{j=1}^{k} nat\,(i, c\,(j))} \tag{5}$$

The result of prognosis analysis consists of:

- $ta(i,o)$ - the predicated time of the i-th attribute life (with its level above a given threshold $th(i)$) counted from the current o-th operation reading,

- $tm(j,o)$ - the predicated time of the j-th module life (with all included in module attributes above given security thresholds, e.g. the protocol module. It is also counted from the current o-th operation reading.

The main drawback of the prognosis implementation consists in the approximated character of base data:

- $pexp(i)$ and $pexa(i)$, i=1, 2,..., lat,
- $sz(i)$,

- $po(i)$ - the probability of the i-th attribute correction after a single operation.

The last parameter contributes especially strong simplification. On the other hand, such approach essentially decreases the complexity of the prognosis algorithm. The $pexp(i)$ and $pexa(i)$ refer to the action and $po(i)$ refers to the operation, therefore, the conjunction of these elements will be represented by multiplication: $pexp(i) \cdot po(i)$ or $pexa(i) \cdot po(i)$. Finally, the time prognosis parameter, which refers to attributes, can be defined in the following way:

- *ex post* prognosis variant:

$$ta'\left(i,o\right) = \left[\frac{sz\left(i\right)}{ccr\left(i\right) \cdot pexp\left(i\right) \cdot po\left(i\right)}\right] \left[the\, steps\, of\, operating\, reading\right] \tag{6}$$

- *ex ante* prognosis variant:

$$ta''\left(i,o\right) = \left[\frac{sz\left(i\right)}{ccr\left(i\right) \cdot pexa\left(i\right) \cdot po\left(i\right)}\right] \tag{7}$$

where:
$[*]$ - round function,
$ccr(i)$ the given correction coefficient for the i-th attribute.

The time prognosis parameter, which refers to modules, can be defined in the following way:

- *ex post* prognosis variant:

$$ta'\left(i,o\right) = \left[\begin{array}{cc} & \\ \min & \dfrac{sz\left(i\right)}{ccr\left(i\right) \cdot pexp\left(i\right) \cdot po\left(i\right)} \\ i = 1, ..., lat & \\ atp(i) = 1 & \end{array}\right] \tag{8}$$

or

- *ex ante* prognosis variant:

$$ta''\left(i,o\right) = \left[\begin{array}{cc} & \\ \min & \dfrac{sz\left(i\right)}{ccr\left(i\right) \cdot pexa\left(i\right) \cdot po\left(i\right)} \\ i = 1, ..., lat & \\ atp(i) = 1 & \end{array}\right] \tag{9}$$

Sometimes, we adapt the more complex data about protocols realized in the future and it is possible to simply regard this information in calculation parameters $pexa(i)$.

5 Conclusions

The result of security communication prognosis permits us to send warnings about the necessity of key or secret exchanging, the appearance of additional users (intruders), the necessity of refreshing the nonce, etc. By supplying time distances concerning the impending threats it is possible to choose the most dangerous and the closest one and make the adequate preventions. The proposed intelligent system works in real time and investigates the continuously changing situation in the auditing convection. We realize the simple prognosis analysis regarding 16 attribute corrections after a single communication operation by program implementation in part of a second (0.6-0.92 sec.). Generally, the reaction to the current communication situation is quick enough to stop protocol realization and to create a prevention strategy.

References

1. Tadeusiewicz, R.: Introduction to Inteligent Systems. In: Wilamowski, B.M., Irvin, J.D. (eds.) The Industrial Electronic Handbook, ch. 1, pp. 1-1 – 1-12. CRC Press, Boca Raton (2011)
2. Tadeusiewicz, R.: Place and role of Intelligence Systems in Computer Science. Computer Methodsin Material Science 10(4), 193–206, 13. Tadeusiewicz, R.: Place and role of Intelligence Systems in Computer Science. Computer Methodsin Material Science 10(4), 193–206 (2010)
3. 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)
4. 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)
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. Focardi, R., Gorrieri, R., Martinelli, F.: Information Flow Analysis in a Discrete -Time Process Algebra. In: Proc. of 13th CSFW, pp. 170–184. IEEE CS Press (2000)
7. Gray III, J.W.: Toward a Mathematical Foundation for Information Flow Security. Journal of Computer Security 1, 255–294 (1992)
8. 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)
9. Alur, R., Dill, D.L.: A Theory of Timed Automata. Theoretical Computer Science 126, 183–235 (1994)
10. Szpyrka, M.: Fast and exible modeling of real-time systems with RTCP- nets. Computer Science, 81–94 (2004)
11. Burrows, M., Abadi, M., Needham, R.: A Logic of Authentication. In: Harper, R. (ed.) Logics and Languages for Security, pp. 815–819 (2007), Di Pierro, A., Hankin, C., Wiklicky, H.: Approximate Non-Interference. Journal of Computer Security 12, 37–82 (2004)
12. Piech, H., Grodzki, G.: The system conception of investigation of the communication security level in networks. In: Abramowicz, W. (ed.) BIS Workshops 2013. LNBIP, vol. 160, pp. 148–159. Springer, Heidelberg (2013)

13. Beauquier, D.: On Probabilistic Timed Automata. Theoretical Computer Science 292, 65–84 (2003)
14. Focardi, R., Gorrieri, R.: A Classification of Security Properties. Journal of Computer Security 3, 5–33 (1995)
15. Tudruj, M., Masko, L.: Toward Massively Parallel Computation based on Dynamic Clasters with Communication on the Fly. In: IS on Parallel and Distributed Computing, Lille, France, pp. 155–162 (2005)