The Modeling Algorithm of Communication Run Operations in a Network

Henryk Piech[1], Grzegorz Grodzki[1], and Aleksandra Ptak[2]

[1] Czestochowa University of Technology, Dabrowskiego 73, 42202 Czestochowa, Poland
{henryk.piech,grzegorz.grodzki}@icis.pcz.pl
[2] Czestochowa University of Technology, Armii Krajowej 19, 42218 Czestochowa, Poland
olaptak@zim.pcz.pl

Abstract. Communication processes have to be observed because there are possibilities that a different kind of threats will occur in the processes of exchanging information in a network. These threats are connected with: the possibility of decryption, losing jurisdiction, believing in and freshness of information, message interception by intruders, etc. We also consider the run of the communication protocol operation. Security attributes have been introduced to analyze the chosen aspects of security, which are proposed by Burrows, Abadi, Needham [4] and others. They have created the system of rules that defines interrelated parts of communication operations with security aspects. In this research we continued the analysis of security in the direction of building the model of auditing and dynamic modification of chosen factors (adequate to the security aspect) with the possibility to form a prognosis.

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

1 Introduction

Information is sent in the form of a message according to protocol systems which should guarantee: encryption safety, sufficient belief level, protection against intruders, and the freshness of information elements [1, 2]. Usually, we may use many mutually interleaved protocols in networks [3, 4]. Obviously, information refers to a different group of users (usually, they are grouped in a pair). Therefore, security analysis will be referred to those groups and they will be the basis of the creation of the so called main security factor [1]. Another main factor can take into account the set of messages, the public key, secret, nonce, etc. Among security attributes one may include the following: the degree of encryption, key and secret sharing, believing in sender or receiver, believing in the honesty of the user, and the freshness level of a message or nonce [1]. M. Kwiatkowska presents security attributes in the figure of probability parameters [5, 6]. This form is smart and very convenient. Therefore, the present proposition is additionally based on the transformation possibility of time attributes into probability characteristics. Apart from rule and time influences we also regard the intruder threat. The influence on security attributes is realized with the help of correction coefficients which also have a probabilistic form, according to the admitted approach. The above-mentioned rules deal on the basis of conditions that are

W. Zamojski et al. (eds.), *Theory and Engineering of Complex Systems and Dependability*,
Advances in Intelligent Systems and Computing 365, DOI: 10.1007/978-3-319-19216-1_32

actions which really appear in communication operations. The division of protocols into operations and operations into actions can be found in many works [1, 2]. In the proposed model we also exploit the so called tokens, which have binary character. A token directly appoints the secure or threat attribute level depending on the relation to a given security threshold. This type of approach improves the assessment reaction on security state changing and helps in the estimation of probability distribution to the next stages and thereby to one of the forms (presented in the following section) of prognosis creation. The proposed model of communication run investigation can be easily realized with probabilistic timed automata (PTA) [7 - 9] and colored Petri net [11, 12], which is especially effective in the parallel strategy variant.

2 The Procedure Concerning the Influences of Protocol Actions

At first, the main security factor(s) is (are) declared for the current action. Action usually influences one or several attributes. The first appearance of the action associated with the security factor leads to the activation of the new automata, node and the equivalent calculation node. This node contains a specific set of attributes. Let us consider the sequence of actions contained in the sample protocol. The structure of the protocol is presented in the example of ASF Handshake [1] (as one of possible main factors):

1. $A \to B: \{N_a\}_{K(a,b)}$,
2. $B \to A: \{N_a, N_b\}_{K(a,b)}$,
3. $A \to B: \{N_b\}_{K(a,b)}$,
4. $B \to A: \left\{A \leftrightarrow^{K'(a,b)} B, N_b'\right\}_{K(a,b)}$,

where:

N_a, N_b, N_b' - are nonces.

A new session key $K'(a,b)$ for A and B is generated when the starting session key is $K(a,b)$. Initial conditions are adequate actions and may be presented as follows:

- both users agree on the starting key: A believes $A \leftrightarrow^{K(a,b)} B$, B believes $A \leftrightarrow^{K(a,b)} B$,
- A defers to $B's$ authority on session keys, A believes B controls $\forall K : A \leftrightarrow KB$,
- B generates the new session key, B believes $A \leftrightarrow^{K(a,b)} B$, - nonces are fresh: A believes N_a is fresh, B believes that N_b is fresh, B believes that N_b' is fresh.

The same parameters can play the role of the main security factor; for example: protocol, user (or a pair of users), key, message service, etc. The type of influences is practically regarded in two algorithm forms concerning attribute corrections:

— $mc = \{0,1\}$ - correction by multiplication by a given updating coefficient MCC in the case of influence pertaining to logic and heuristic rules, $mc = 1$ - the activation of this form regarding attribute correction, $mc = 0$ - the rejection of this form of correction.

— $ec = \{0,1\}$ - correction by exchanging to the current level (represented by the current coefficient value of ECC) in the case of influences relating to the lifetime or user (intruder). Therefore, it is possible to simultaneously use two forms of correction for a single attribute. So, when $ec = 1$ then the attribute value does not have to increase:

$$at_{t=k+1}(i) \xrightarrow{mc=0,ec=0} at_{t=k}(i),$$
$$at_{t=k+1}(i) \xrightarrow{mc=1,ec=0} at_{t=k}(i) * MCC,$$
$$at_{t=k+1}(i) \xrightarrow{mc=0,ec=1} ECC,$$
$$at_{t=k+1}(i) \min\{at_{t=k}(i) * MCC, ECC\}.$$

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

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

or

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

where:

$mhc(i) = \{0,1\}$ - heuristic rule influence activation,

$ehc(i) = \{0,1\}$ - dishonest user influence activation.

The actual value of ECC, in the case of the lifetime type of influence, will be counted by the formula:

$$ECC = 1 - e^{t(j) - lt(i)} \tag{1}$$

where:

$t(j)$ - the time of attribute activation,

$lt(i)$ - the attribute lifetime.

In reality, the time activity is transformed into a probability attribute value according to a given attribute lifetime. The actual value of ECC, in the case of an additional user (intruder) type of influence, will be counted by the formula:

$$\begin{aligned} &ECC = if\ (nus < nht)\ then\ ECC = 1, \\ &else\ ECC = enht - nus \end{aligned} \tag{2}$$

where:

nus - the number of users (in the environment of the main security factor),

nht - 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. 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 - the number of main factors: $k = 1,2,...,nf$;

nna - the length of the multi-run; the number of all actions in the network: $j = 1,2,...,nna$;

nat - the number of attributes (it is assumed that structures of security nodes for all main factors are the same: $i = 1,2,...,nat$. Obviously, it is not necessary; in such case we will use $i(k) = 1,...,nat(k)$,

$fat(k,i)$ - the matrix of the attribute structure of main factors,

$mhc(i)$ - heuristic rule influence activation,

$ehc(i)$ - dishonest user influence activation,

mcc - the correcting coefficient value,

mhc - the correcting coefficient value,

$t(i)$ - the time of the *i*-th attribute activation,

$lt(i)$ - the lifetime of the *i*-th attribute $ec(i) = 1$,

nus - the number of users,

nht - the number of honest users,

$w(i)$ - the weight of the attribute according to communication security.

After the realization of the procedure of describing variable input, we will execute the security assessment algorithm. The stages of this algorithm are as follows:

1. action input,
2. the recognition of the attribute corrected by the action,
3. the recognition of the type of correction,
4. correction realization,
5. go to the 3-rd point until the last attribute,
6. the recognition of the main factor activated by the action,
7. token structure creation for the main factor,
8. security state estimation for the main factor,
9. go to the 6-th point until the last factor,
10. auxiliary analysis (the creation of the prognosis with respect to the threaten state the distribution of probabilities of transitions to the next stages),
11. go to the 1-st point until the last action.

The general presentation of the algorithm in the form of a pseudo-code is very convenient:

```
Procedure SECURITY ASSESSMENT
for j ← 1 to nna do
{j - the number of the action}
begin
  for i ← 1 to nat do
  {i - the number of the attribute}
  if cc(j,i) = 1 then
  begin
  if mc(i) = 1 then atm(i) ← at(i)*mcc;
  if ec(i) = 1 then ate(i) ← (1-exp(t(i)-lt(i));
  if mhc(i) = 1 then ahm(i) ← at(i)*(1-mhc);
  if ehc(i) = 1 then if (nus > nht) then
 ahu(i) ← exp(nht-nus)else ahu(i) ← at(i);
 at(i) ← min(atm(i), ate(i), ahm(i), ahu(i));
  if at(i) < th(i) then tk(i) = 0 else tk(i) = 1;
  end;
  for k ← 1 to nnf do
  {k - the number of the main security factor}
  begin
  GFS(k) ← 0; St(j) ← 1;
  for i ← 1 to nat do
  if fat(k,i)=1 then
  begin
  GFS(k) ← GFS(k) + w(i) * at(i);
  {GFS(k)-the security level of k-th factor}
  St(j) ← St(j) + 2(i-1) * tk(i);
  {St(j) - the code of the new state}
  end
  end
end.
```

The procedure concerning the distribution of the creation probability to the next states and the prognosis procedure will be described in the following sections.

3 The Procedure of Security Distribution Analysis

The fundamental axiom is based on the impossibilities of increasing the level of communication security during the realization of a run. Therefore, the security states, which are possible to achieve can be constrained in the following way:

$$\left\{St(j): at(i,j) \le at(i,j-1), i = 1,2,...,\text{nat}\right\}$$

or

$$\{St(j): tk(i,j) \le tk(i,j-1), i = 1,2,...,\text{nat}\} \tag{3}$$

where

j - the number of the current action

In general, the probability of achieving the threaten state $tk(i,j)=0$ by a given attribute is defined as follows:

$$prob(tk(i,j)=0) = \begin{cases} th(i)/at(i,j-1) \text{ if } at(i,j-1) < th(i). \\ 1 \qquad\qquad otherwise. \end{cases} \tag{4}$$

Let us define the description of the threaten zone.

Definition 1. A tuple (At', Tk'', Th) is a threaten zone description, where At' - the regarded attributed set (their names), Tk'' - given token boundary values (in the threaten zone all tokens are equal to or less than Tk'', Th - the set of given thresholds for all regarded attributes.

Definition 2. The scale of coming closer to the threaten zone CTZ is measured with respect to the average of distances between given (Th is used as a characteristic of the threaten zone) and current attributes. In practice, it is expressed in the following way:

$$CTaZ(k) = 1/nat(k) \sum_{\substack{i=1, \\ at(i)>th(i), \\ tk(i)\ge\sim tk''(i), \\ fat(k,i)=1}}^{nat} (at(i)-th(i)) \text{ - the attribute closeness}$$

for the *k*-th factor;

or

$$CTZt(k) = 1/nat(k) \sum_{\substack{i=1, \\ tk(i)>tk''(i), \\ fat(k,i)=1}}^{nat} (tk(i)-tk''(i)) \text{ - the token closeness}$$

for the *k*-th factor;

where

$\{*\}$ - added conditions referring to the index.

Axiom 1. The number of new possible states achieved from the state $ST(j)$ is equal to:

$$nnst = 2^v, \text{ where } v = \sum_{i=1; tk(i)\ge tk''(k,i)=1}^{nat} (tk(i)-tk''(i)),$$

where $tk_j(i)$ - token values in the investigated *j*-th state (it infers from the impossibility to achieve the security level better than the current situation).

Obviously, only attributes with the token equal to 1 will be taken into account in the analysis of the distribution probabilities concerning the transition to the next state. Attribute transition probability is estimated on the basis of the current attribute value and a given security threshold (threaten zone is placed under it) (fig.1).

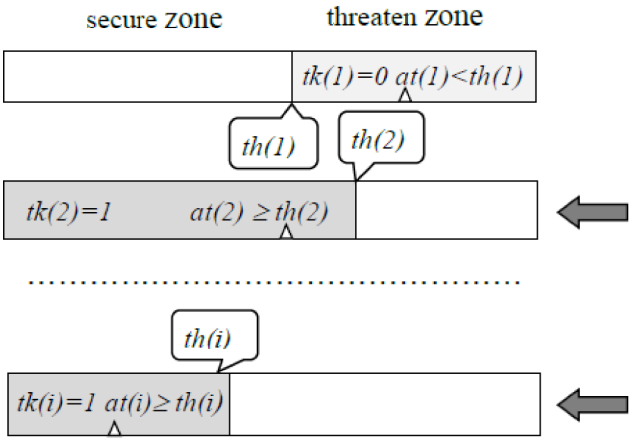

Fig. 1. The selection of relevant attributes according with: {tk(i)=1}

Collar 1. The probability of the achieved concrete possible state is defined as follows:

$$prob\left(St\left(j\right)St\left(j+1\right)=St\left(g\right)\right)=$$

$$\frac{prcg\left(k,j,g\right)prtg\left(k,j,g\right)}{\sum_{tk^{j+1}(1)=0}^{1}\sum_{tk^{j+1}(2)=0}^{1}\cdots\sum_{tk^{j+1}(nat)=0}^{1}prc\left(k,j\right)prt\left(k,j\right)},$$

where g - a given state from the v feasible state,

ingredient formula elements are defined as follows:

$$prcg\left(k,j,g\right)=\prod_{\substack{i=1;\\ tk^{j}(i)=1;\\ fat(k,i)=1;\\ tk^{j+1}(i)=0;\\ tk^{j+1}(i)\in St^{(g)}}}^{nat}th\left(i\right)/at\left(i\right),$$

$$prtg\left(k,j,g\right)=\prod_{\substack{i=1;\\ tk^{j}(i)=1;\\ fat(k,i)=1;\\ tk^{j+1}(i)=1;\\ tk^{j+1}(i)\in St^{(g)}}}^{nat}\left(at\left(i\right)/th\left(i\right)\right)/at\left(i\right),$$

$$prc\left(k,j,g\right)=\prod_{\substack{i=1;\\ tk^{j}(i)=1;\\ fat(k,i)=1;\\ tk^{j+1}(i)=0;}}^{nat}th\left(i\right)/at\left(i\right),$$

$$prt\left(k,j,g\right)=\prod_{\substack{i=1;\\ tk^{j}(i)=1;\\ fat(k,i)=1;\\ tk^{j+1}(i)=1;}}^{nat}\left(at\left(i\right)/th\left(i\right)\right)/at\left(i\right).$$

The result presented in the collar is based on attribute changing probability $th(i) = at(i)$ for the token (adequate for the attribute) transition from 1 to 0, and $(at(i) - th(i))/at(i)$ for staying on level 1 (transition from 1 to 1). The consideration of the transition probability refers to a given main factor k. Thus, only attributes fulfilling the condition: $\{fat(k,i) = 1\}$ would be regarded. The given next state g, for which the probability transition is defined, has to be described by the set of tokens fulfilling the condition: $\{tk^{j+1}(i) \in St^{(g)}\}$. The denominator represents the full probability space for all feasible states. It is a permutation that refers only to relevant attributes, i.e. those which can change their token value $\{tk^{j}(i) = 1\}$ because only tokens defined the state code. Therefore, only this kind of situation can influence the changing state: $\{tk^{j+1}(i) = 0\}$ or $\{tk^{j+1}(i) = 1\}$. Let us pay attention to two theoretical cases: token transition from 0 to 1 and from 0 to 0. The first case is impossible (axiom 1), the second is realized with probability equal to 1 (multiplication by 1 does not contribute any changes). By calculating the dominator value, we may estimate the distribution of probabilities. This task is realized by the following procedure.

```
Procedure TRANSITION PROBABILITY ANALYSIS (j, k)
u := 1; s := 0;
for tk(j+1, 1) := 0 to 1 do
{tk(j+1, i) - token value in (j+1) - th state (next
state)}
  for tk(j+1, 2) := 0 to 1 do
  ……..
  for tk(j+1, nat) := 0 to 1 do
  begin
    m(u) := 1;
    for i := 1 to nat do
    begin
    if (tk(j, i) = 1) and (tk(j+1, i) = 0) and
    (fat(k, j) = 1)
    then m(u) := m(u) * th(i) / at(i);
    if (tk(j, i) = 1) and (tk(j+1, i)=1) and
    (fat(k, j) = 1)
    then m(u) := m(u) * (at(i) - th(i)) / at(i)
{m(u)- denominator (7) component of probability of tran-
sition to u - th feasible state, u - state code}
    end;
  s := s + m(u); {value of denominator (7 )}
  p(j, u) = m(u) / s;
  {p(j, u) = transition probability from state j-th
  to u-th}
  u := u + 1;
  end.
```

The algorithm return values of the probability concerning the transition to all feasible next states:

$$prob\left(St\left(j\right) \rightarrow St\left(j+1\right)\right) = m\left(u\right)/s,\ \ u = 1,2,....,v. \tag{6}$$

The short time prognosis is defined by the maximum transition probability and the long term prognosis can be determined on the basis of trends or the distribution of the different types of operations in the run (this problem will not be explored here).

4 Conclusions

The simple form of procedure algorithms does not guarantee a low level of complexity but simultaneously the limited number of attributes (usually less than 10) permits us to audit long sequences of operations in a run (in the online investigation). On the other hand, the experiences pertaining to known inter- leaved protocols (Kerberos, Andrew RPC, Needham-Shredder, CCITT X.509 etc.) show and approve a short (<30 state changing) process of achieving a threaten zone in main security factors. Usually, these factors consist of several (5 - 8) security attributes. It gives the possibility to create the prognosis and give warning about different kinds of threats.

References

1. Burrows, M., Abadi, M., Needham, R.: A Logic of Authentication. In: Harper, R. (ed.) Logics and Languages for Security, pp. 15–819 (2007)
2. Zhang, F., Bu, L., Wang, L., Zhao, J., Li, X.: Numerical Analysis of WSN Protocol Using Probabilistic Timed Automata. In: 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems (ICCPS), p. 237 (2012)
3. Lynch, N.: Timed and Probabilistic I/O Automata. In: 28th Annual IEEE/ACM Symposium on Logic in Computer Science (LICS), p. 12 (2013)
4. Chen, T., Han, T., Katoen, J.: Time-Abstracting Bisimulation for Probabilistic Timed Automata. In: 2nd IFIP/IEEE International Symposium on Theoretical Aspects of Software Engineering, TASE 2008, pp. 177–184 (2008)
5. 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)
6. 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)
7. Huang, Y.-S., Chiang, H.-S., Jeng, M.D.: Fault measure of discrete event systems using probabilistic timed automata. In: 2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 1218–1223 (2011)
8. Wu, J., Wang, J., Rong, M., Zhang, G., Zhu, J.: Counterexample generation and representation in model checking for probabilistic timed automata. In: 2011 6th International Conference on Computer Science & Education (ICCSE), pp. 1136–1141 (2011)

9. Thorat, S.S., Markande, S.D.: Reinvented Fuzzy logic Secure Media Access Control Protocol (FSMAC) to improve lifespan of Wireless Sensor Networks. In: 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), pp. 344–349 (2014)
10. Liu, K., Ye, J.Y., Wang, Y.: The Security Analysis on Otway-Rees Protocol Based on BAN Logic. In: 2012 Fourth International Conference on Computational and Information Sciences (ICCIS), pp. 341–344 (2012)
11. Zhang, H., Liu, F., Yang, M., Li, W.: Simulation of colored time Petri nets. In: 2013 IEEE International Conference on Information and Automation (ICIA), pp. 637–642 (2013)
12. Li-Li, W., Xiao-jing, M., Yang, N.: Modeling and verification of Colored Petri Net in stop and wait protocol. In: 2010 International Conference on Computer Design and Applications (ICCDA), vol. 5, pp. V5-24–V5-28 (2010)