On a concept of scalable security: PKI-based model with supporting cryptographic modules

Bogdan Księżopolski[1], Zbigniew Kotulski[2]
[1]Institute of Physics, M. Curie-Skłodowska University,
Pl. M. Curie-Skłodowskiej 1, 20-031 Lublin, Poland
e-mail:bogdan@kft.umcs.lublin.pl
[2] Institute of Fundamental Technological Research of PAS
ul. Świętokrzyska 21, 00-049 Warsaw, Poland
and Institute of Telecommunications of WUT
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
e-mail:zkotulsk@ippt.gov.pl

Public services called „e-everything" (e-government, e-banking, e-commerce, etc.) meet many different barriers that reduce their efficient applicability. One of them is requirement of assurance of the information security when it is transmitted, transformed, and stored in the electronic service. It is possible to provide an appropriate level of security applying the present-day information technology. However, the level of the protection of information is often much higher than it is necessary to meet potential threats. Since the level of security strongly affects the performance of whole system, the excessive protection decreases the system's reliability and availability and, as a result, its global security. In this paper we present a model of scalable security for digital information transmission systems (being usually the crucial part of e-service). In our model the basic element of the security is the Public Key Infrastructure (PKI) enriched by specific cryptographic modules.

1. Introduction

Advanced teleinformatic technologies, nowadays provide a wide range of possibilities of development of industry or the institutions of public services. The big stress is put on the development of well-available information services called "e-everything" like e-government, e-money, e-banking. These mentioned processes are fulfilled mainly by electronic way, thanks to which one can increase their availability, cutting down the expenses at the same time.

Implementation of these services is connected with the proper level of information security sent between the parts of protocols [12,14,16]. Among teleinformatic technologies and cryptographic modules there are the ones, which protect different information security services e.g. : confidentiality, integrity, non-repudiation, and anonymity of data. The important problem seems to be the establishing of the level of information security fulfilled services in a given protocol. Every use of any Internet service is connected with information exchange, which in the case of successful attack, causes different threats to the whole process. This problem can be solved by estimating the security levels for each phase of the protocol

[1]. Such attitude seems to be only a partial solution, because thanks to a given service one can send information of different level of threats. A common practice is to use exaggerated means of information security which decreases efficiency, system availability, introduces redundancy. Another effect of exaggeration of security mechanisms is increasing the system complexity, which later influences implementation of a given project in practice.

The solution of this case seems to be the introduction of scalable security model, which can change security level depending on particular conditions of a given case. In the article mechanism, which can modify the level of information security for each phase of protocol, is presented. Parameters, which influence modification of the security level, are: the risk of successful attack, probability of successful attack and independence of security elements. The used security elements, which take care of the protection of information, are based mainly on PKI services and cryptographic modules.

2.Security services

In practice realization of the electronic processes is connected with fulfillment of many law a formal standards. While projecting the systems we can take care of different security services [1,2]. Among them we can enumerate: Confidentiality of data, integrity of data, anonymity parts of protocol, non-repudiation of sender and receiver, authorization, secure storage, management of privileges, public trust. Every security service has got its own characteristics (Table 1).

Group of services	**Name of services**	**Characteristics**
Integrity	*integrity of data*	Guarding against improper information modification or destruction
Non-repudiation	*non-repudiation of action*	Non-repudiation of sending the message
	non-repudiation of sender	Non-repudiation of sender identity
	non-repudiation of receiver	Non-repudiation of receiver identity
Confidentiality	*Confidentiality of data*	Preserving authorized restriction on information access and disclosure
Authorization	*Authorization of parts of protocol*	Correct authorization of parts of protocol is needful for taking part in protocol
Privileges	*management of privileges*	function in protocol are depend on the permission level
Anonymity	*Network anonymity*	Anonymity of message sender (with network anonymity)
	Anonymity of data	Anonymity of message sender (without network anonymity)
Availability	*Availability of services*	Ensuring timely and reliable access to and use of information

Public trust	*Trust between parts of protocol*	Possibility of public verification of action in protocol between parts of protocol
	TTP trust	Possibility of public verification of action in protocol with TTP usage
Secure storage	*Secure storage of data*	Confidential and permanent storage of information

Table 1: Characteristics of security services.

3. Security elements

The system conditions that are described by the security services, can be fulfilled with many different security elements. To achieve this goal we can use different mechanisms [3,4,5]. In the article we will focus on two groups of solutions, services based on PKI [1,3] and independent cryptographic modules [4].

Security elements based on PKI:

- *Registration* In order for a user to join the PKI environments ones must register with certifying TTP. The main function is to establish the reliable unique binding between a user and his public key Function (public key / secret key).
- *Digital Signatures:* Thanks to digital signature the message authentication, message integrity and non-repudiation can be fulfilled.
- *Encryption:* Encryption is a basic service providing the cryptographic functions for protection of message confidentiality In a computer network
- *Time-stamping:* Time-stamping is described as the process of attaching data and time to a document in order to prove that it existed at a particular moment of time.
- *Non-repudiation:* This mechanism involves the generation, accumulation, retrieval and interpretation of evidence that a particular party processed a particular data item.
- *Key management:* The service deal primarily with the handing of cryptographic keys in a proper, efficient, scaleable and secure way [6].
- *Certificate management:* A digital certificate is an electronic token ensuring the binding between an entity and its public key. The functions supporting this service include generation, distribution, storage, retrieval, and revocation of digital certificates.
- *Information repository:* This service maintains the collection of data critical for the operation of the TTP system [7].
- *Directory services:* In order to interact, a member of a PKI must hale access to information about other PKI members.
- *Camouflaging communication:* Camouflaging communication not only provides data confidentiality, but also hides the very fact of communication.
- *Authorization:* PKI user who possesses a resource may grant the right to another PKI user to access this resource. TTPs should ensure the granting of rights, including the ability to access specific information or resources.

- *Audit:* In order to ensure that certain operational, procedural, legal, qualitative and several other requirements are complied with, so that is enhanced, an auditing service is required.
- *TTP to TTP interoperability:* Interoperability services are concerned with the issues necessary for establishing a network of TTPs, using simultaneously different TTP can do verification of parts of protocol, which ensure the authenticity of TTP usage.
- *Notary:* Using TTP can do public verification of parts of protocol.

Cryptographic modules:

- *SSS:* Secured secret sharing scheme can be used in case when the encrypted message by particular public key can be decrypted only with cooperation of define number of parts [4].
- *PKG:* The module which generate strong cryptographic keys, the e.g. PKG, based on biometric method [10]. This technique generate personalized cryptographic keys from the face biometric, which offers the inextricably link to its owner.
- *Anonymizer:* The mechanism, which protects anonymity of parts of protocol, the example of it could be Crowd. This is scalable system based on world-wide-web services, which assure anonymity of message sender inside network communication [13].
- *AA:* The user identification scheme that also can simultaneously achieve key exchange requirement while preserving the user anonymity [15].
- *Individual numbers:* Generated individual numbers by parts of protocol can improve user anonymity [9].

4. The conception of scalable security

The realization of electronic process is dependent of proper level of security. During the projecting of mentioned process the security mechanisms are established. They are usually overestimated according to real risk. One can notice that there are differences connected with information sent in the same electronic process. They concern different threats, which in the case of successful attack will affect the parts of protocol. In case of small threat there is a great possibility of decreasing redundant means of information security, which in reality will improve efficiency, system availability and as a consequence it will increase its security.

Conditions

Secure electronic processes are based on cryptographic protocol. By their means one can introduce many security services, which enable the process. The cryptographic protocols realize security services by means of different security elements: e.g. PKI services and cryptographic modules. The usage of these security elements is strictly defined by cryptographic protocols. As a result of that, any modification of their content is forbidden otherwise it will ruin all protocols.

Te solution of that problem is creating different protocols realizing the same service but on different level of security[1]. By using a precise service one can choose a

[1]For simplicity, when we will change the unimportant element from a protocol point of view but important as far as security, we will call it a new protocol

protocol in accord with security requirements. Some security elements are worth configuring before the process of using the services but not on a dynamic process. Using some unchangeable security elements whose change is critical for given processes causes it.

Parameters of scalable security

Security level of electronic process can depend on different factors. Security can be modified by means of their proper choice. In a presented conception of scalable security, protection of information is a correlation, which is a function of three parameters:

$$F_S = \sum_{i}^{a} \sum_{j}^{b} \sum_{x}^{c} (L_{ij}^{x})[\omega_{ij}^{x}(1 - P_{ij}^{x})](\frac{\omega_{ij}^{x} L_{ij}^{x}}{\omega_{ij}^{x}})^{Z}$$ [2]

Among parameters one can single out:

1. *Protection level:* L_{ij}^{x};
2. *Risk of attack on a given service:* $[\omega_{ij}^{x}(1 - P_{ij}^{x})]$;
3. *Dependence of security elements:* $(\frac{\omega_{ij}^{x} L_{ij}^{x}}{\omega_{ij}^{x}})^{Z}$;

Every of presented parameters are counted for all subprotocols which a given cryptographic protocol consists of and all the steps of these subprotocols.

The first parameter is a definition of protection level for a given cryptographic service in a given step of subprotocol. This is a sum of chosen security elements, which guarantee security of a given service.

The Second parameter shows a risk of attack on a given security service. This is a multiplication of average losses made by successful attack and probability of attack on a given security service.

The third describes independence of security elements used to gain a proper protection level. The security elements are tied, not using some protection of information mechanisms in a beginning subprotocols greatly influences other subprotocols. The level of convergence can also be changeable; it depends on e.g. a number of subprotocols, security level.

The security level of electronic processes mainly depends on the used elements of protection of information required by security services. In the presented article, security elements are based on PKI services and cryptographic modules. In Table 2,

---

[2] s – security level, which is realized by a given version of protocol;
i – a number of subprotocols in a given protocol;
j – a number of steps of parameters in a given subprotocol;
x – a concrete security service;
ω_{ij}^{x} - weight describing a average cost of loses after successful attack for a given service;
$\omega \in (0,1)$
L_{ij}^{x} - value of security elements for a given service; $L \in (0,1)$
P_{ij}^{x} - probability of attack on a given service; $P \in (0,1)$
Z - convergence of security elements. $Z \in (1,25)$

dependences of security services and security mechanisms are presented. Every security service can be realized by different security mechanisms. Security level of a given protocol will depend, among other things, on them. For every security elements their contribution to global protection of services is defined L_{ij}^{x}. Individual contribution is defined in percentage.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Integrity of data (I)	Digital Signatures L_I1=50%	Key management L_I2=10%	Certificate management L_I3=10%	Directory services L_I4=5%	TTP to TTP interoperability L_I5=15%	PKG L_I6=10%			
Non-repudiation of action (NRM)	Digital Signatures L_NRM=30%	Time-stamping L_NRM=15%	Key management L_NRM3=10%	Certificate management L_NRM4= 10%	Audit L_NRM5=5%	Non-repudiation PKI L_NRM6=10%	Directory services L_NRM7=5%	Information repository L_NRM8=5%	PKG L_NRM9= 10%
Non-repudiation of sender (NRS)	Digital Signatures L_NRS1=30%	Time-stamping L_NRS2=15%	Key management L_NRS3=10%	Certificate management L_NRS4=10%	Audit L_NRS5= 5%	Non-repudiation PKI L_NRS6=10%	Directory services L_NRS7=5%	Information repository L_NRS8= 5%	PKG L_NRS9= 10%
Non-repudiation of receiver (NRR)	Digital Signatures L_NRR1=30%	Time-stamping L_NRR2= 15%	Key management L_NRR3= 10%	Certificate management L_NRR4= 10%	Audit L_NRR5= 5%	Non-repudiation PKI L_NRR6= 10%	Directory services L_NRR7= 5%	Information repository L_NRR8= 5%	PKG L_NRR9= 10%
Confidentiality of data (C)	Encryption L_C1=50%	Key management L_C2=10%	Certificate management L_C3=10%	SSS L_C4=15%	Directory services L_C5=5%	PKG L_C6=10%			
Authorization of parts of protocol (Au)	Registration L_Au1=20%	Digital Signatures L_Au2=20%	Key management L_Au3=10%	Certificate management L_Au4=10%	TTP to TTP interoperability L_Au5=10%	Directory services L_Au6=5%	Authorization PKI L_Au7=10%	AA L_Au8=10%	
Management of privileges (MP)	Registration L_MP1=50%	Authorization PKI L_MP2=50%							
Network anonymity (AN)	Crowds L_AA1=100%								
Anonymity of data (AM)	Individual numbers L_AM1=100%								
Trust between parts of protocol (PTA)	Time-stamping L_PTA1=30%	Information repository L_PTA2=30%	Audit L_PTA3=20%	TTP to TTP interoperability L_PTA4=20%					
TTP trust (PTT)	Time-stamping L_PTT1=30%	Information repository L_PTT2=20%	Audit L_PTT3=10%	TTP to TTP interoperability L_PTT4=10%	Notary L_PTT5=30%				
Secure storage of data (SS)	Encryption L_SS1=30%	Time-stamping L_SS2=10%	Key management L_SS3=10%	Certificate management L_SS4=10%	Non-repudiation PKI L_SS5=10%	Information repository L_SS6=15%	Directory services L_SS7=5%	Audit L_SS8=5%	PKG L_SS9=5%

Table 2: Security dependencies describing possible security services and security elements realizing them.

Security dependencies of security elements (Table 2) are only an example. It can be created in a free way using different security mechanisms. The value of the parameter *L* is constant for particular security dependence; during creating protocols of different level of protection this parameter is not modified.

Parameter, which is set up for every step of subprotocols is weight for particular services ω_{ij}^{x}. These weights can be changeable in particular processes, because losses of successful attack can be different in depending on concrete, transported information.

5. Usage of scalable security: e-auction.

Conception of scalable security can be realized to different type of cryptographic protocol [8,9]. In the article we present an example, which implements conception of scalable security for electronic auction, which is based on proper cryptographic protocol [9].

Model

Analyze protocol of e-auction consists of four subprotocols: *certification, notification of auction, notification of offer as well as choice of offer.* In protocol take part *N* bidders (O_1, ... ,O_N), third trustworthy person that is *GAP* (main auction agency) as well as firm, which wants to announce the auction.

The first step of protocol is verification by GAP, the participants taking part in e-auction, that is the bidders O_N as well as firm F which wants to announce the auction (the *subprotocol of certification*). The next step is notification to GAP the auction by verified firm F. GAP publishes the conditions of notified auction, giving all requirements notified by F (*the subprotocol of notification of auction*). In the next step, person wanting to take part in auction, after earlier verification, sends his offer to GAP (*the subprotocol of notification of offer*). The last subprotocol is executed after elapsing of time for notification of offers, then firm F as well as bidders O_N, send their parts of secret (needed to read offers) to GAP. After decoding them, they will be sent to firm F, where victorious offer will be chosen. In the same subprotocol, the firm F sends information about the victorious offer to GAP, then it will be published to (be generally known) public message (*the subprotocol of choice of offer*).

The communication between participants of protocol is safe. We achieve it thanks to using public key cryptography, where every participant of protocol possesses his private key (SK) as well as public key (PK). Those practical keys are not solid, their validity ends with the validity of registration number, which is achieved in subprotocol of certification

Chosen protocol

In the article we will present usage of scalable security for subprotocol of notification of electronic auction whose description we show below (Figure 1).

The protocol can be notified by any person, which got earlier in subprotocol of certification suitable authorizations. Such a person, indicated as *F,* should possess the registration number NR_F, his time stamp T_{NR_F}, private key SK_F as well as conditions of notified auction WP_F . F generates with the help of the generator of random numbers (KG), his individual number NF.

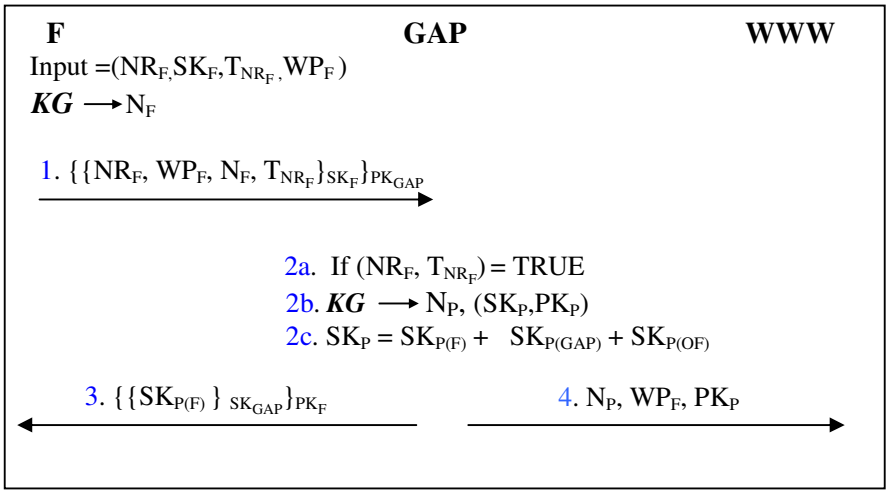

Figure 1: Graf to the auction notification subprotocol.

Step1:
In the first step, F sends to GAP, signed digitally (SK_F) as well as coded (PK_{GAP}) following information: his registration number (NR_F), his time stamp (T_{NRF}), the conditions of auction (WP_F) as well as his individual number (N_F).
Step2:
The main auction agency (GAP) verifies the registration number F (NR_F) as well as validity of his gauge of time. After positive authorization GAP generates the individual number of auction (N_P) as well as a few keys for concrete auction (SK_P,PK_P). The private key of auction (SK_P) is divided by use of the threshold scheme of dividing secret. Secret is divided into three parts, designed for F($SK_{P(F)}$), for GAP ($SK_{P(GAP)}$) as well as bidders in auction ($SK_{P(OF)}$). Each part is necessary to reproduce private key (SK_P).
Step3:
GAP sends digitally signed (SK_{GAP}) as well as coded (PK_F)) - the part of secret designed for F ($SK_{P(F)}$).
Step4:
GAP publishes, for example on WWW site, the number of auction (N_P), conditions of it (WP_F) as well as its public key (PK_P).

Results

The first step, which we should make, is defining weights, which describe risk „ω_{ij}^{x}"for particular security services in the steps of subprotocol. In described case defined weights are constant for a given process. If any security service is not required in a given step, the weight of described risk is equal to zero. Below we present the values of weights for a given subprotocol (Table 3):

	Step 1	Step 2	Step 3	Step 4
ω^{I}	**0,5**	**0,4**	**0,3**	**0,3**
ω^{C}	**0,7**	**0,7**	**0,5**	**0**
ω^{NRS}	**0,3**	**0**	**0,3**	**0,3**
ω^{Au}	**0**	**0,7**	**0**	**0**
ω^{SS}	**0**	**0,3**	**0**	**0**
ω^{MP}	**0**	**0,3**	**0**	**0**

Table 3: The values of weights for a given subprotocol

During the second step, we define security elements, which realize chosen security elements (Table 4). This element is changeable for every version of described subprotocols. In the article we will describe three versions of subprotocol, the first, basic ("A"), and others, with larger number of security elements ("B") and smaller number of security elements.

	A						***B***						***C***					
	L^{I}	L^{C}	L^{NRS}	L^{Au}	L^{SS}	L^{MP}	L^{I}	L^{C}	L^{NRS}	L^{Au}	L^{SS}	L^{MP}	L^{I}	L^{C}	L^{NRS}	L^{Au}	L^{SS}	L^{MP}
Step 1	0,8	0,7	0,65	0	0	0	0,95	0,9	0,8	0	0	0	0,5	0,5	0,45	0	0	0
Step 2	0,35	0,85	0	0,95	0,65	0,5	0,5	0,9	0	1	1	1	0,3	0,35	0	0,5	0,45	0,5
Step 3	0,8	0,7	0,5	0	0	0	0,95	0,85	0,6	0	0	0	0,5	0,5	0,3	0	0	0
Step 4	0,5	0	0,4	0	0	0	0,8	0	0,55	0	0	0	0,5	0	0,3	0	0	0

Table 4: Security elements for a given subprotocol.

During the third step, we set up probability of attack on a particular services in described steps of protocol. (Table 5) Those values are constant for a given process.

	Step1	Step2	Step3	Step4
P^{I}	0,8	0,3	0,3	0,7
P^{C}	0,7	0,9	0,8	0
P^{NRS}	0,4	0	0,2	0,6
P^{Au}	0	0,5	0	0
P^{SS}	0	0,3	0	0
P^{MP}	0	0,5	0	0

Table 5: The values of probability in a given subprotocol.

The last parameter is a parameter of function convergence whose characteristics are shown in Figure 2.

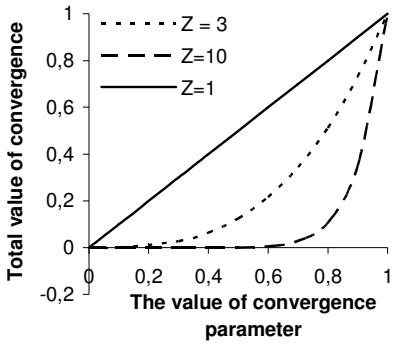

Figure 2: Characteristic of convergence parameter.

In a described subprotocol the value of the parameter Z = 3 was chosen.
The last step checking the security levels of particular version of subprotocols is counting function F; the results are presented in Table 6.

	Step1	*Step2*	*Step3*	*Step4*	*Total*
A	0,123512	0,372681	0,125026	0,008697	**0,629915813**
B	0,29296	0,773427	0,254351	0,047845	**1,368582313**
C	0,026756	0,04318	0,021319	0,006597	**0,097851875**

Table 6: The values of security levels for particular steps and whole subprotocol.

Conclusions

After analyzing the results we can assume that we obtained three versions of described subprotocol, every with different level of protection. Basic level of subprotocol ("A") is much higher from the level with minimal security elements ("C"). We can assume, that only in case with transporting unimportant data in a given process is worth using. The version with the highest security level ("B"), points to essential protection of subprotocol. That is why it is worth using this version for processes where critical information for parts of protocol takes part.
Setting up different security levels for every subprotocol in the whole protocol helps us to change particular versions of subprotocol, creating freely scalable, as far as security level, the final protocol. Such a possibility can be useful in case of modifying the security levels in particular phases of subprotocol [17], which can decrease system performance as a result its security.

7. Bibliography.

[1] C. Lambrinoudakis, S. Gritzalis, F. Dridi, G. Pernul, „Security requirements for e-government services: a methodological approach for developing a common PKI-based security policy", Elsevier: Computer Communication 26 (2003) 1873-1883.

[2] NIST, "Volume I: Guide for Mapping Types of Information and Information Systems to Security Categories", March 2004.

[3] A. Patel, P. Gladychev, S. Katsikas, S. Gritzalis, D. Lekkas, KEYSTONE project, "Support for Legal Framework and Anonymity in the KEYSTONE Public Key Infrastructure Architecture".

[4] K. Kulesza, Z. Kotulski, On Automatic Secret Generation and Sharing for Karin-Greene - Hellman Scheme, in: J. Sołdek, L. Drobiazgiewicz, [ed.], Artificial Intelligence and Security in Computing Systems, Kluwer 2003, pp. 281-292. ISBN: 1-4020-7396-8.

[5] J. Groves, "Security for Application Service Providers", Network Security, Volume: 2001, Issue 1, January 1, 2001, p.6-9.

[6] ISO/IEC 11770-3: Key management-Part 3: Mechanisms using asymmetric techniques. 1999-11-01.

[7] ETSI TS 102 042: Policy requirements for certification authorities issuing public key certificates. 2002.

[8] L. Barlow; „A Discussion of Cryptographic Protocols for Electronic Voting", 2003.

[9] B. Księżopolski, Z. Kotulski; "Cryptographic protocol for electronic auctions with extended requirements"; Annales UMCS Informatica v.2, 2004, pp. 391-400.

[10] A. Teoh, D.Ngo, A.Goh; "Personalised cryptographic key generation based on Face Hashing"; Elsevier, Computer & Security (2004) 23, pp. 606-614.

[11] G. Saez; "Generation of key predistribution schemes using secret sharing schemes"; Elsevier , Discrete Applied Mathematics 128 (2003), pp. 239-249.

[12] J. Groves; "Security Application Service Providers"; Elsevier , Network Security, Volume: 2001, Issue 1, pp.6-9.

[13] M. Reiter, A. Rubin; "Crowds: Anonymity for Web Transaction"; ACM Transaction on Information and System Security, Vol. 1, No. 1, November 1998, pp. 66-92.

[14] M. Merabti, Q. Shi. R. Oppliger; "Advanced security techniques for network protection"; Elsevier , Computer Communications 23 (2000) pp.151-1583.

[15] W. Tzong-Sun, H. Chien-Lung; "Efficient user identification scheme with key distribution preserving anonymity for distributed computer networks"; Elsevier, Computer & Security (2004) 23. pp. 120-125.

[16] M.A. Patton, A. Josang; "Technologies for Trust in Electronic Commerce"; Kluwer Academic Publishers, Electronic Commerce Research, 4, pp. 9-21 (2004).

[17] S. Moitr, S. Konda; "An empirical investigation of network attacks on computer system" Elsevier, Computer & Security (2004) 23, pp. 43-51.