Trust Management in E-business Systems - From Taxonomy to Trust Engine Architecture

DENIS TRČEK   GORAZD KANDUS
Department of digital communications and networks
Institut ”Jožef Stefan”
Jamova 39, 1001 Ljubljana
SLOVENIA
denis.trcek@ijs.si   gorazd.kandus@ijs.si   http://epos.ijs.si

Abstract: Trust is becoming an increasingly important topic in security of e-business systems. Trust turned out to be essential for further penetration of e-business technologies, especially for agents based technologies. Therefore a proper taxonomy is needed and trust has to be formalized in order to enable development of trust engine for such applications. The main objective of the paper is thus how to practically deal with trust in e-business environment, from taxonomy to trust engine architecture. The approach is based on facts learned from e-business systems security.

Keywords: e-business security, agents technologies, trust management.

1 Introduction

The importance of security was growing with the penetration of computer communications during the last decades. Trust is closely related to security in distributed systems. This has been recognized already in mid-eighties, but the relationship between trust and security, including the definition of trust with its formalization, is still a topic of research. In Webster’ s dictionary, trust is defined as an assumed reliance on some person or thing. It is a confident dependence on the character, ability, strength, or truth of someone or something. Furthermore, trust is a charge or duty imposed in faith or confidence or as a condition of a relationship. Finally, trust means placing a confidence (in an entity).

In standardization area trust became an issue almost twenty years ago [1]. Few years later formal methods for analysis of cryptographic protocols were developed. Trust played an important role there as well. For example, BAN logic [2] that was the most successful formalism in the field, significantly depended on trust. One of its basic definitions states: ”$P \mid\!\Rightarrow X$: *P has jurisdiction over X* - The principal P is an authority on X and should be trusted on this matter. In the second half of the nineties some specialized trust management solutions appeared:

- W3C standardized a platform for content selection or PICS [3]. PICS defines formats and distribution of labels that are meta-data for description of Web documents.
- AT&T developed PolicyMaker [4] that binds access rights to an owner of a public key, whose identity is bound to this key through a certificate.
- IBM recognized trust to be central to e-business so it developed Java based Trust Establishment Module and appropriate language [5]. The solution is similar to PolicyMaker. It supports role-based access based on X.509 [6] certificates, where it additionally provides negative rules for preventing access.

Recently, an extensive survey on trust has been published by Grandison and Sloman [7]. This survey defines trust informally, states its properties, analyzes existing solutions for trust management and lists the most common fields of application of trust management. Authors provide their definition of trust as ”the firm belief in the competence of an entity to act dependably, securely, and reliably within a specified context”. The main motivation of authors to study trust is to be able to model it for use in automated systems, therefore they do not consider the social concept of trust. However, it is not clear how the authors of the above definition intended to implement trust supporting applications.

It is common to all above approaches to consider trust as a property of a system. Moreover, trust management is frequently misinterpreted as ”who is allowed to do what under what circumstances”, which actually denotes security policy [8]. So at the beginning of nineties Denning [9] analyzed the concept of trust and came to conclusion that trust is not a property of an entity or a system, but it is an assessment. This assessment is based on experiences, it is shared through network of people interactions and is continually remade each time the system is used. And this will be the basis for our approach.

The paper is structured as follows. In the second section, the selection of a proper framework for trust formalization with relation to distributed security is given. In the third section a taxonomy of trust is given[1], while in the fourth section trust is formalized and trust engine architecture is given. There is a conclusion with directions for future work in the last section.

2 Selection of a Proper Framework for Trust

Security in distributed systems can be analyzed from three points of view. The first one are cryptographic primitives. The second one are cryptographic protocols (interactions) and the third one are programs (code) as such. Starting with crypto-primitives it can be observed that there are no formal proofs about the bottom-line time-computational complexity of cryptographic primitives. Put another way, it is not known whether more efficient algorithms for attacks than those currently known exist or not (see e.g. [10, 11] and [12, 13]). Similar holds true for crypto-protocols. It is not possible to state a bullet-proof evidence that a particular protocol is bug-free. Formal techniques like BAN are only strengthening such belief, but they can not provide a complete assurance about it (as a result, wise engineering practices have to be followed [14]). The third point of view is related to implementation - security is also a matter of general software correctness [15], where things are getting worse on account of several reasons. Wireless nomadic computing requires handling of unpredictable application environment, and support for applications to be aware of environment / context is needed. Next, instead of client server paradigm a more efficient, peer-to-peer networking, is emerging. But peer-to-peer networking inherently brings more entropy [16]. On top of this, objects are becoming mobile and intelligent and their interaction is at their own will.

Thus when making threats analysis, a rational attitude towards a potential breach is as follows. A distributed communication system is treated as a generic set, consisting of atomic elements, which are crypto-primitives, crypto-protocols and software units. In many cases it is not possible to exactly determine the probability of a failure for each of those elements. Based on experiences it is however possible to have some belief that addresses a subset of that generic set and assigns certain beliefs to these subsets. Such attitude makes sense also when a successful breach occurs. It is often impossible to identify the very atomic state that ”went wrong”. But based on experiences it is possible to have an opinion how likely it is for a group of atomic states that one of them went wrong.

An approach that takes this phenomenon into account, and can be used for formalization of trust, is Shafer’ s theory of evidence [17]. Recently this theory has been used as a basis for Jøsang’ s subjective algebra [18, 19], which is directly related to trust. Its main contribution is preservation of a mathematically sound basis, while introducing logical operations for trust. Subjective algebra contains not only equivalents to traditional logical operators (conjunction, disjunction and negation), but it also introduces new ones like recommendation and consensus. These operations manipulate operands that express imperfect knowledge of subjects. Thus trust ω (also called an opinion) is modeled with a triplet

[1] Taxonomy is a system for naming and organizing things into groups which share similar qualities.

(b, d, u), where b stands for belief, d for disbelief and u for uncertainty. Each of those elements gets its continuous values from a closed interval $[0, 1]$, such that $b + d + u = 1$. For example, an opinion of agent A about object O can be expressed as $\omega_O^A = (0.4, 0.4, 0.2)$. Jøsang' s algebra is mainly concerned with expressed trust and laws of its propagation in social interactions. It is trying to find justifications for its logical operations and it has to be proved with practical experiments how realistically it reflects the phenomenon of trust.

The basis for further derivations in this paper will be Shaffer's theory. The main problem with this theory is an assumption that agents are able to rationally assign proper values to (b, d, u). This is not very often the case, so additional improvements are needed for trust management. Moreover, rational values of this triplet present only a small part of the whole trust phenomenon.

3 Trust Taxonomy

It is important to note that trust as a manifestation of reasoning and judgment processes is a notion that inherently belongs to the field of psychology [20]. This doesn't necessarily mean that trust is incomputable, but the following elements have to be taken into account:

- **Irrationality**. It should not be assumed that each agent is able to rationally assign values to d, b, u. It is obvious that this is not the case in many situations, for example, when an entity is under pressure, tired, etc.

- **Context dependence**. Agent' s trust is a function of a context (environment). The first level of context dependence deals with agent' s trust by exclusion of social interactions. The second level of context dependence includes social interactions.

- **Temporal dynamics**. Agent' s relation towards object / subject being trusted is a dynamic relation and it changes with time.

- **Action binding**. An opinion can serve as a potential (a basis) for agent' s deeds.

- **Feed-back dependence.** Trust is not a product of a completely independent mind. Being forced to adopt a certain kind of behavior, agent may change opinion about the very same kind of behavior.

- **Trust differentiation.** Trust evolves into various forms. The reasons are bad communication capabilities of an entity, expressing trust, bad perceiving capabilities of a targeting entity, and trust being mediated intentionally modified.

The basic properties of trust should take irrationality, temporal dynamics, context dependence, action binding, feed-back dependence and trust differentiation into account. Besides, taxonomy of trust has to include also the following:

- Trust should be divided into minor trust (denoted by $\underline{\omega}$), that is expressed, communicated, and major trust, that is personal, intimate trust (denoted by $\overline{\omega}$). Major trust is further divided into rational trust, denoted by $\overline{\omega}_r$, and actual trust, denoted by $\overline{\omega}_a$.

- Trust should be modeled in a way that encompasses improperly assigned values. For this purpose a so called *reason-lapse* function ζ is introduced, which appropriately assigns values to b, d and u, satisfying the condition that $b + d + u = 1$.

4 Formalization of Trust and Trust Engine Architecture

A model, based on these requirements, will give a tangible ground for implementation, experimental research and judgment about computational trust.

Definition 4.1 *Let T denote a set of time values t and let Δ denote a set of deeds δ. Then the set of contexts Γ is defined as $\Gamma = T \times \Delta$.*

Definition 4.2 *Let $\overline{\omega}_r$ denote agent' s major rational opinion with b, d and u being his / her belief, disbelief and uncertainty, such that $b + d + u = 1$ and $b, d, u \in [0, 1]$. Then rational opinion is defined as $\overline{\omega}_r = (b(t, \delta), d(t, \delta), u(t, \delta))$.*

Definition 4.3 *Let ζ denote a reason-lapse function that operates on $\overline{\omega}_r$, i.e. the values of b, d and u, by preserving $b+d+u=1$ and $b,d,u \in [0,1]$. Then agent' s actual opinion is defined as $\overline{\omega}_a = \zeta(\overline{\omega}_r)$.*

Definition 4.4 *Let $\overline{\omega}_a$ denote actual opinion of subject S as defined above and let δ denote a deed of subject S. Then a relationship betwen the set of opinions and set of deeds is defined with action-binding function η, such that $\delta = \eta(\overline{\omega}_a)$.*

Definition 4.5 *Let $\overline{\omega}_a$ denote actual opinion of subject S as defined above. Then expressed opinion $\underline{\omega}_a$ is given by a function ϑ, so that $\underline{\omega}_a = \vartheta(\overline{\omega}_a)$.*

Summing up, irrationality is modeled by reason-lapse function ζ, action binding by function η, feedback dependence by function φ, trust differentiation by functions ζ, ϑ, and context dependence (including temporal dynamics) by function φ. A note on action-binding function. One could comment that it should operate on actual opinion. However, this contradicts the nature of reason-lapse function, because this function presents an ignorance as such, including environment.

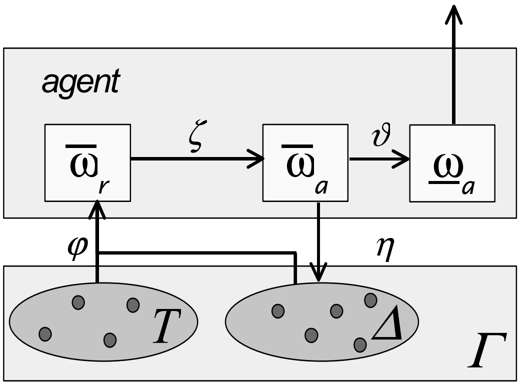

Figure 1: A model of trust for agents environment.

Figure 1 serves as the basis for implementation of trust in a computerized environment, where the situation should be analyzed from two points of view. The first one covers agents in our possession, while the second one covers foreign agents.

It is evident that rational trust $\overline{\omega}_r$ and functions φ, ζ can be in principle modeled for agents in our possession, while this does not hold true for foreign agents But for practical implementations in computer networks, the nature of functions η, ϑ and opininons $\overline{\omega}_a, \underline{\omega}_a$ is far more important:

- Action-binding function and major actual opinion can be obtained by observation of deeds. An action is an evident fact that is influenced ba opinions. It is also a kind of an aggregate of history of opinions, affected by contexts and it can be used to judge on opinions.

- For modeling trust differentiation it is interesting to note that minor actual opinion may vary significantly from major actual opinion. But security of agents is still very limited and there is no way to prevent code peeping [2]. In this case there would be no need to model ϑ, as $\overline{\omega}_a = \underline{\omega}_a$.

It is obvious that an implementation of trust engine requires significant resources not only in terms of space, but also processing capabilities, especially when considering extensive contexts. Trust implementation requires recording of agent' s history. This consequently requires establishment of appropriate databases that are used for calculation of trust by use of appropriate AI techniques, e.g. datamining. Thus a tiny trust engine that would be suitable for implementation within agent itself would have significant limitations.

Because of these facts trust engine should consist of server engine at a remote location that is contacted by a mobile agent over the network. Intensive and realistic treatment of trust requires trust engine to be at a remote location, running as server, while the actual trust of an agent at a certain moment is communicated to the agent. Besides, to make trust even more accurate and realistic, a trust engine server can be upgraded with a front-agent that is capable to communicate with various databases over the internet [21].

[2] A promising method for prevention of this threat would be mobile cryptography, which is currently at a theoretical stage.

5 Conclusions

Trust is essential for e-commerce. For a wider penetration and better acceptance of agents based solutions it is desired to make trust somehow computable. In this paper trust was analyzed, its taxonomy was given, trust was formalized and implementation of trust engine was presented. It has been argued why theory of evidence is an appropriate basis for this purpose and it has been shown that the solution is complementary with subjective algebra.

One essential question to be addressed at the end is why fuzzy logic [22] has not been mentioned in this paper. It seems natural to think about treatment of trust with fuzzy logic, as trust is essentially a fuzzy term. When talking about uncertainty of natural language, fuzzy logic handles phenomena of qualitative expressions with a reference to a quantitative system, to an absolute framework. For example, sentence ”John is tall.” is analyzed with a reference to length expressed in meters, centimeters, etc. The problem with trust is that there does not exist such a framework and it has yet to be established.

References

[1] Department of Defense, *Trusted Computer System Evaluation Criteria*, DOD 5200.28-STD, December 1985.

[2] Burrows M., Abadi M., Needham R., *A Logic of Authentication*, ACM Transactions on Computer Systems, vol. 8, no. 1. pp. 18-36, February 1990.

[3] Miller J., Resnick P., Singer D. *PICS Rating Services and Rating Systems*, http://www.w3c.org/TR/REC-PICS-services.

[4] Blaze M., Feigenbaum J.,Lacy J., *The Role of Trust Management in Distributed Systems Security*, IEEE Conf. on Security and Privacy, Oakland 1996.

[5] Herzberg A. et al., *Access Control Meets Public Key Infrastructure*, IEEE Conf. on Security and Privacy, Oakland 2000.

[6] ITU-T, *The Directory - Public-key and attribute certificate frameworks*, Recommendation X.509), Geneva 2000.

[7] Grandison T., Sloman M., *A Survey of Trust in Internet Applications*, IEEE Communications Surveys, IEEE Society Press, 4.th Quarter 2000, pp. 2-13.

[8] Trček D., *Security policy conceptual modeling for networked information systems*, Computer Communications, No. 17, Vol. 23, Elsevier, November 2000, pp. 1716-1723.

[9] Denning D., *A new Paradigm for Trusted Systems*, Proc. of ACM SIGSAC New Security Paradigms Workshop, ACM, New York 1993, pp. 36-41.

[10] Ajtai M., Dwork C., *A public-key cryptosystem with worst-case / average-case equivalence*, Proc. of 29.th ACM STOC, 1997, pp. 284-293.

[11] Nguyen P., Stern J., *Cryptanalysis of the Ajtai-Dwork Cryptosystem*, Proc. of Crypto 98, LNCS, Springer Verlag, Heidelberg 1998, pp. 223-242.

[12] Rivest R., *The MD5 Message Digest Algorithm*, RFC 1321, IETF, April 1992.

[13] Dobbertin H., *The Status of MD5 After a Recent Attack*, CryptoBytes, RSA Labs, Vol. 2, No. 3, 1996, pp 1-6.

[14] Abadi M., Needham R., *Prudent Engineering Practice for Cryptographic Protocols*, SRC Research Report 125, Digital Corp., Palo Alto, 1994.

[15] Geihs K., *Middleware Challenges Ahead*, Computer, IEEE, June 2001, pp. 24-31.

[16] Parameswaran M. et al., *P2P Networking: An information Sharing Alternative*, IEEE Computer, July 2001, pp. 31-37.

[17] Shafer G., *A Mathematical Theory of Evidence*, Princeton University Press, Princeton 1976.

[18] Jøsang A., *A Logic for Uncertain Probabilities*, Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, No. 3, Vol 9, World Scientific Publishing Company, June 2001.

[19] Jøsang A., *The right type of trust for distributed systems*, Proc. of the New Security Paradigms Workshop, ACM, 1996.

[20] Piaget J., *Judgment and reasoning in the child*, Littlefield Adams, Totowa 1969.

[21] Gams M., *A Uniform Internet-Communicative Agent*, Electronic Commerce Research, No. 1, Vol. 1, Kluwer Academic Publishers, 2001, pp. 69-84.

[22] Zadeh L.A., *Fuzzy Sets*, Information and Control, No. 8, 1965, pp. 338-353.