Journal of Information Technology Impact

Vol. 2, No. 3, pp. 117-123, 2001

Trust Models for Access Control

Vasilios Katos [1]
William Alyea [2]
Cambridge Technology Partners
The Netherlands

Abstract

This paper using the concept of a trust model aims at providing practical value to distributed security. It describes a method to derive authentication and authorisation requirements from the reduction of trust inherently associated with the number of participating entities placed in a communication path between two parties. It demonstrates how a traditional approach to understanding the trust relationships between interacting entities can be applied directly to the definition and design of secure e-business architectures. By applying these principles, the business and social impact of a security architecture which optimizes the provided level of security, performance of the architecture and the investment associated with that security architecture can be realized.

Keywords: Trust models, authentication, authorisation.

Introduction – Background

In real life, society is inherently relying on trust. A customer, for example, trusts his money to a bank. However, in many cases contracts maybe based on a simple handshake.

In the electronic world, trust is not so easily realised or delivered. People do not have the opportunity to experience the comfort provided by a handshake or eye contact. In the electronic world trust is delivered (i.e. implemented) by security mechanisms, which can be cryptographic, physical, contractual, and so forth. It should be highlighted that although complete security is not feasible, trust is essential in order to design a system. In other words, trust is specified on a design level, whereas security must reflect trust to the implementation level.

Trust models are typically associated with public key infrastructures, where the distribution and possession of certain information influences the behaviour between communicating parties. More particularly, asymmetric encryption provides the means for linking trust with the certainty that public information is bound to an identity (Chadwick *et al*., 1997). Trust models are mainly focusing on these binding processes between the public key and the identity of the underlying party. Two well known trust models are the Pretty Good Privacy (PGP) Trust Model which is often referred to as a "web of trust" (Zimmerman, 1995), and the hierarchical Public Key Infrastructure (PKI) with a root Certificate Authority (ITU, 1989).

Identification, authentication and authorisation have become key aspects of distributed systems security today because they are inherently linked to confidentiality and non-repudiation. Twenty years ago, the problem of the proof of identity was more manageable, and confidentiality was the popular security service. Today's highly interactive e-commerce domain has resulted in a rearrangement of the priorities of security services. However, one can argue that confidentiality is again the ultimate security goal, and in order to be able to provide this service the problems that must be solved are more related to identification, authentication, authorisation and integrity, rather than developing a strong cipher.

This paper presents a method for identifying the locations of the authentication and authorisation mechanisms in a e-business architecture. The method focuses on *where* these access control mechanisms must be placed, rather than *how* they will be implemented, since the latter depends mainly on the type of system being deployed, the type of industry the system is serving, as well as the organisation's attitude towards security. The key parameter for identifying the locations is trust. Briefly stated, the level of trust between communicating parties may permit a certain party to utilise the authentication and authorisation performed by another party, or force additional authentication and authorisation. The concept of the trust model is used to manage the trust relationships. The paper is organised as follows. Two views of trust models are described in the next section, namely the deterministic and the probabilistic view. Then, the method of deriving access control is described by using a probabilistic trust model of a representative e-business architecture example. The paper concludes by presenting the business and social impact of the method.

Deterministic vs. Probabilistic View

The most basic of all trust models is the trust/no-trust model. Simply stated, an entity is either trusted to perform a given action or it is not. In this model, there is no "level of trust" associated with transactions beyond the binary definition of trust. As an example, the Customer-to-Bank scenario is used to explain the trust relations between the parties which perform a transaction. In real life, when a customer visits a branch office to perform a transaction with the Bank, the logo of the Bank displayed is adequate for fulfilling the customer's requirements for identification. The physical location of the branch is also an important factor for meeting the trust requirements of the customer. If the branch is not located in an office space, but instead it is a desk with a clerk sitting in the middle of a street, the customer will not trust the branch office. In the case of a Bank, the lack of such trust could prohibit the customer for completing business.

Depending on the type of transaction, there are different trust requirements placed upon each party. If the customer needs to make a deposit, his identity is not necessarily required, but the customer needs to positively identify the branch. Once this identification is performed, the user trusts the clerk sitting inside the physically protected area. The actual identity of the clerk is not necessarily known to the customer, but there exists transitional trust from the Bank to its representative clerk. It can be seen that trust is directly related to the identification of the Bank entity, not the clerk entity.

On the other hand, if the customer requests a withdrawal, he needs to identify himself to the clerk. Again the latter is the representative of the Bank. Although the customer can be

successfully identified, he is not permitted to walk up to the safe and withdraw the cash. Instead, the clerk is trusted to perform the customer's request. It can be seen that the central trusted entity for the transactions is the clerk, whose identity is known to the Bank and is trusted to access the Bank's safe to execute the withdrawal.

In the scenarios above the existence of trust sets the required milestones in the processes. The deterministic view of the trust model dictates that in every step in the process, an entity will be either trusted or not trusted. If the entity is not trusted, the action cannot be performed. The customer for example is not trusted to access the safe and withdraw money. If the entity is trusted, permission for performing the action is granted.

The probabilistic view of the trust model relates trust with the risk of breaking this trust. A level of trust is assigned to an entity, which relates to the risk of the trust being broken. More precisely, the higher the risk of broken trust, the lower the trust level. In risk classification theory, risk is inherently associated to the probability of an event being successful. In the security domain, such an event is an attack attempt - malicious or accidental – to the confidentiality, integrity and availability of the data of an information system. Once risk classification is performed, the risk levels are mapped to trust levels.

Against the above background it follows that the mapping function between risk and trust must be proportionally inverse. Although the actual form of the function depends on the organisation's culture regarding risk taking, the following three mapping function categories may be mentioned:

- For a **trust optimistic** organisation, where risk needs to increase significantly in order to have effective trust reduction, a representative function is of a concave form as shown in Figure 1(a).
- For a **trust pessimistic** organisation, where trust is high only when risk is maintained adequately low, a representative function is of a convex form as shown in Figure 1(b).
- For a **trust indifferent** organisation, a representative function would be linear, as shown in Figure 1(c).

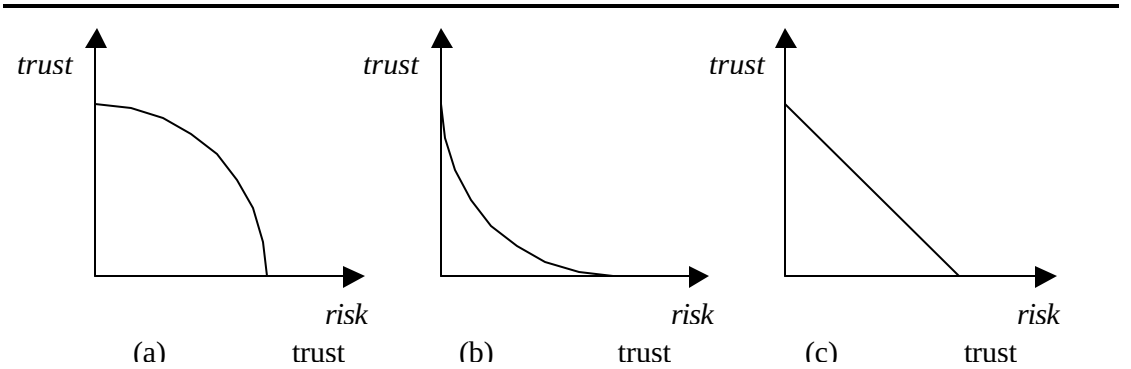

Figure 1. Three representative mapping functions

A method for deriving access control from the trust model

In this section an example e-business architecture is used to describe a new method for placing authorisation and authentication mechanisms based on the level of trust between the interacting entities.

A representative e-business architecture involves a user (U), which communicates with the web server (A), which communicates with an application server containing the business logic (B), which must communicate with a database to retrieve data (C). Firewall architectures suggest that the web server is situated on a demilitarised zone (DMZ), due to the direct session with an external entity (i.e. the user).

Trust directly relates to authorisation and authentication requirements as shown in Figure 2.

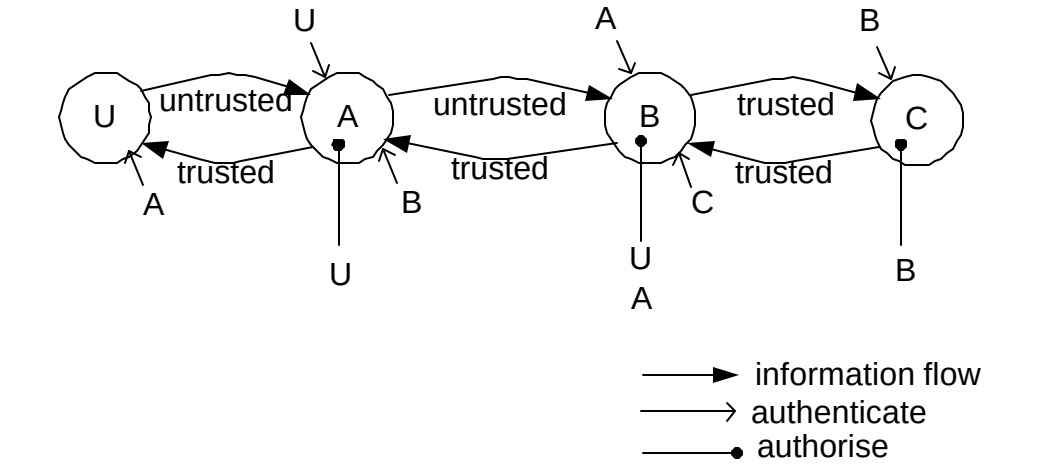

Figure 2. An example trust model.

- In the example in Figure 2, A does not trust U, B does not trust A, but C trusts B. It can be seen that the replies are trusted, since in the opposite case it would not make sense to contact the entity in the first place. The trust is assigned on the arrow which represents information flow from one entity to another. In a typical e-business architecture, the user initiates a transaction by sending a request to the system. This is implemented with an HTTP client on the user side sending a request to an HTTP server.

- Entities must be authenticated and authorised to perform a given action.

- The higher the level of required trust, the stronger the security mechanisms that must be in place. In some cases, trusting an entity is a mandate (for example, an authentication server).

- A user is not trusted. Therefore, authorisation must always be performed.

The methodology for the deterministic trust model is fairly straightforward; when an entity is not trusted, authorisation is performed on all entities which follow the untrusted path. Authentication is performed between interacting entities. In the example above, B does not trust A, therefore authorisation is performed for both U and A. On the other hand, C trusts B and authorisation is not required for entities other than B. That is, C trusts that B has adequately authenticated U, or trusts that U has been authenticated somewhere in the communication path and the authentication is valid. Therefore, there is no need to perform additional authentication of U, since this would increase the overhead.

In the case of the probablistic trust model, risk classification (NIST, 1992) is performed on each of the participating entities. The results are assigned on the entities as trust levels, and the trust/no trust condition is based on the accepted level of trust. Figure 3 presents a probablistic trust model with the trust defined in the normalised range [0,1].

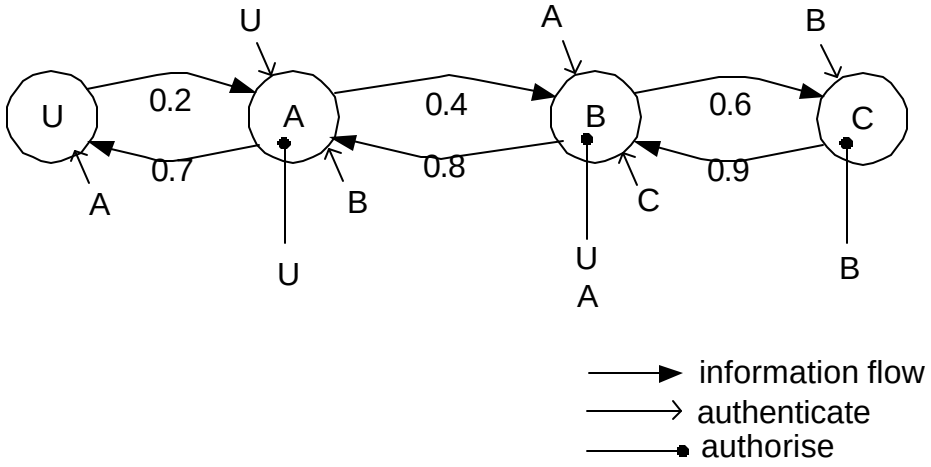

Figure 3. A probabilistic trust model.

Setting the trust threshold to, say 0.5, all trust relationships above that value are trusted, whereas all other values yield untrusted relationships.

Deciding on the trust threshold is a process which involves the stakeholder and depends on the industry where the system operates, as well as the organisation's culture relating to risk taking. Consequently, this process inherits the weaknesses and pitfalls of risk classification.

The observation of Beth et al. (1994) indicates the trust between two entities is decreasing when the entities are further apart. More analytically, Beth et al. proved what was intuitively true; the more parties intercepted a communication channel between party A and party B, the higher

the chance of the information passing from A to B being corrupted, in terms of integrity. However, integrity is required for performing authentication successfully, since if the integrity of the authentication data is corrupted, the authentication data itself would not be valid.

This translates on the above model as follows: A trusting U is 0.2, B trusting A is 0.4, but B trusting U is expected to be less than 0.2. For simplicity, if it is assumed that the events which relate to risks of broken trust on the entities of the trust system are independent, then the trust level of B to U would be Pr(*B trusts U*)=Pr(*B trusts A*|*A trusts U*)=0.2 x 0.4 = 0.08. It must be highlighted that in real systems the events will rarely be independent, but such a study is outside of the scope of this paper.

Concluding remarks

A practical approach on using a trust model for deriving access control is described in this paper. The proposed methodology can be used as a tool to identify where the access control mechanisms should be placed, in order to avoid redundant authentication and authorisation operations. Such an approach has an impact on the cost of the security investment, both financial and computational, since the security functions can be very costly and the benefit – i.e. the risk reduction - is not as high. However, the implementation of the access control mechanisms depend on the system instance and may not necessarily be technical. On the contrary, security mechanisms must be placed on all levels ranging from the infrastructure to the organisational level, in order to have effective risk management.

Security design based on the notion of trust may have a social impact, since trust is one of the main drives in decision making in all aspects of life. Consequently, an effective transfer of trust into the e-business environment will allow a successful integration between technology and society, since trust will serve as a common point of reference. In the literature trust models are primarily associated to public key infrastructures. This paper extends the use of a trust model outside the scope of a public key infrastructure, allowing trust to participate in the security design of architectures which do not involve PKIs. Trust is an attribute of consumers, users and businesses and the inherent limitations originating from technology implementations should not bound the use of the trust concept. This paper attempts to maintain trust implementation-free. Therefore trust should appear on the design level of a system, whereas the security mechanisms must appear on the implementation level, in order to deliver the required trust levels.

Finally, the proposed methodology can be used for assessing the security of an existing system, in terms of its access control functions. Furthermore, due to the dependence between trust and access control, the fact that trust decreases with the introduction of entities in a communication path, may result into a rearrangement of the authentication and authorisation operations.

References

Beth, T., Borcherding M., Klein B., (1994). Valuation of Trust in Open Networks, *ESORICS 94*, pp.3-14.

Chadwick D., Young A., Kapidzic Cicovic N., (1997). Merging and Extending the PGP and PEM Trust Models – The ICE-TEL Trust Model, *IEEE Networks*, pp.16-24.

ITU (1989). *Recommendation X.509, The Directory Authentication Framework*, International Telecommunications Union.

NIST (1992). *An Introduction to Computer Security: The NIST Handbook*, Special Publication 800-12, National Institute of Standards and Technology.

Zimmermann, P., (1995). *The Official PGP User's Guide*, Massachusetts: MIT Press.

---

[1] Dr. Vasilios Katos is a Security Architect at Cambridge Technology Partners, A Novel Company, at The Netherlands. He can be reached at Het Kasteel 1, Woerden 3441 BZ, The Netherlands. Email: vkatos@ctp.com; Phone: +31(20)5750575; Fax: +31(20) 575-0500.

[2] Mr. William Alyea is Head of the Security Group at Cambridge Technology Partners, A Novel Company, at The Netherlands. He can be reached at: Het Kasteel 1, Woerden 3441 BZ, The Netherlands. Email: walyea@ctp.com; Phone: +31(20)5750575; Fax: +31(20) 575-0500.