

A Trust-Centered Approach for Building E-Voting Systems*

A. Antoniou^{1,3}, C. Korakas¹, C. Manolopoulos¹, A. Panagiotaki¹, D. Sofotassios¹,
P. Spirakis^{1,3}, and Y. C. Stamatiou^{1,2}

¹ Research Academic Computer Technology Institute, N. Kazantzaki, University of Patras,
26500, Rio, Patras, Greece

² Mathematics Department, 451 10, Ioannina, Greece

³ University of Patras, Department of Computer Engineering, 26500, Rio, Patras, Greece
{antoniou,ckorakas,manolop,panagioia,sofos,spirakis}@cti.gr,
istamat@cc.uoi.gr

Abstract. eVoting is a challenging approach for increasing eParticipation. However, lack of citizens' trust seems to be a main obstacle that hinders its successful realization. In this paper we propose a trust-centered engineering approach for building eVoting systems that people can trust, based on transparent design and implementation phases. The approach is based on three components: the decomposition of eVoting systems into "layers of trust" for reducing the complexity of managing trust issues in smaller manageable layers, the application of a risk analysis methodology able to identify and document security critical aspects of the eVoting system, and a cryptographically secure eVoting protocol. Our approach is pragmatic rather than theoretical in the sense that it sidesteps the controversy that besets the nature of trust in information systems and starts with a working definition of trust as people's positive attitude towards a system that performs its operations transparently.

Keywords: Tools and Technologies for eParticipation and eVoting, Trust and Security: provisions and instruments, eDemocracy and eParticipation Challenges, Risk Assessment, Cryptographic Protocol, Security Architecture.

1 Introduction

The rapid growth of Information and Communication Technologies (ICTs) and the diffusion of Internet in people's everyday lives in conjunction with the need for more, better and economical government services to the citizens has led the past few years to the development of eGovernment throughout most of Europe. In this context, democratic societies face the challenge to improve public participation in political debate and policy formation processes, realizing the concept of *eParticipation*. One of the most important and critical facets of eParticipation is *Electronic Voting* or eVoting. eVoting has attracted lately the attention of many governments as an alternative to conventional voting with the hope to increase citizens' participation and reduce the costs.

While eParticipation initiatives have been deployed across the EU with mixed results so far, some encouraging signs come from few but important eVoting initiatives. In Switzerland, for example, eVoting and especially Internet voting, was recently introduced as a complementary channel for elections and referenda, with great success. One of the reasons might be that remote voting was largely practiced through postal voting for many years. The introduction of Internet voting came as an alternative and easier way to vote remotely and thus was rapidly accepted. In 2005, Estonia carried out the first Nation Wide online elections in the EU. It was the result of a bold political decision rather than a natural evolution as it came to be in Switzerland, but it placed Estonia on the forefront of the eVoting efforts in Europe. This, perhaps, would not have been possible if the government had not already implemented an advanced IT Strategy and a Nation Wide Digital ID scheme. In both cases, some basic conditions were met to allow the fruitful deployment of such initiatives, in terms of the necessary infrastructures, institutional measures and government policies employed to support large scale deployment of eVoting projects.

Recent efforts to implement eVoting in Greece, face in that respect many challenges, such as the lack of a specific institutional framework supporting the deployment of eVote applications at large scale (e.g. PKIs) or the low ICT and Internet penetration rates (around 25% [17]) and the resulting digital divide and “digital culture gap”. In addition, the general lack of trust in ICTs and the Internet, as a safe medium to conduct secure transactions, further hinders these efforts.

This lack of trust in ICTs and the Internet affects very seriously any effort to migrate from the conventional and long established voting procedures to an electronic voting system, since voting is a fundamental process in any democracy. Moreover, the abundance of cases of misconduct in electronic voting has resulted in severe decrease of *trust* among citizens [2]. However, eVoting, despite the critique, seems to be, still, a hot discussion issue and, possibly, a worldwide reality in the future.

According to the above, any successful eVoting system should target at increasing *citizen's trust*. Trust, however, is difficult to establish in the eVoting domain since eVoting is necessarily based on complex distributed information systems, involving complicated interactions between computers, between humans, and between humans and computers.

There is much ongoing research in the development and analysis of new trust management models for complex and dependable computer systems. Blaze *et al.* in [3] proposed the application of automated trust mechanisms in distributed systems. In [9] the focus is on the strong relationship between the notions of trust and security. Moreover, a number of schemes for the design of secure information systems have been proposed (see, for example, [5], [8]) which are based on automated trust management protocols. The composition and propagation of trust information between elements of information systems is also of pivotal concern and a number of research works are devoted to them (see [18], [11], [24],[7]).

With regard to trust in the eGovernment domain, specifically, there are specialized research efforts in building trust models based on distributed trust agents, much like as in PKIs [23]. There are many open issues both conceptual and practical, however, that pertain to eGovernment trust, many of which are discussed in [19] and [21].

There are even less efforts for trust management in the eVoting domain. Due to the complexity of an eVoting system, most efforts are focused on the study of specific

system security requirements such as, for instance, establishing uncoercibility of the voters ([1]). Also, as a common practice for strengthening trust, many approaches focus on the existence of a voter verifiable paper copy of the ballot or the design of strong cryptographic protocols (e.g. [20],[6]). Finally, the work done by the OASIS consortium [16] is a first step towards the standardization of secure eVoting architectures based on formal modelling and risk assessment methodologies (e.g. use of the EML language and threat evaluation techniques).

In this paper, we propose a system-oriented trust management approach that handles eVoting at a system engineering level, as a whole. The approach targets all the phases of system design, implementation and testing, using trust modelling and risk assessment methodologies in conjunction with strong cryptographic protocols. This approach is currently being applied for the implementation of an Internet-based eVoting system that will be initially deployed in an actual voting process by the Technical Chamber of Greece.

2 Trust in the eVoting Domain

Since trust, as people's attitude, plays a major role in the way people view and use information systems, lack of trust renders even expensive and sophisticated information systems completely useless. In most of the information systems that deliver e-services, trust is based not on some publicly available systematic design process, but rather on the reputation of the system's implementer (e.g. a well-known company) and operator (e.g. the government).

On the other hand, trust is a hard to formalize concept that also raises philosophical and social (i.e. non-engineering) concerns. For instance, Luhmann's research [15] considers trust as a mechanism which causes the reduction of complexity. Coleman [4] distinguishes certain elements that define a trust situation between a trustor and a trustee. By definition a voting procedure is a trust situation, and in this case trust properties have to be reflected both on individual and system level, independently of the voluntary, custom/norm based, institutional or obtruded nature of the procedure. Trust is an emergent social property based on interactions between actors and for this reason, an eVoting procedure could, in principle, be established, if and only if, actors are convinced that it complies with certain trust properties.

Given the multifaceted nature of trust, in our approach the concept of trust is *pragmatic* in the sense that we rely on a *plausible* working definition and proceed in order to satisfy the definition's prerequisites for trust. One possible definition of trust is the following:

Trust of a party A in a party B for a service X is the measurable belief of A in that B will behave dependably for a specified period within a specified context.

In the eVoting domain, A is the voter, B is the eVoting system and X is the eVoting service. Most importantly, by *dependably* we will imply ensuring the following basic requirements (which apply to both eVoting and conventional voting): *democracy* (only voters who have the right to vote can vote and one vote per voter is included in the election outcome), *accuracy* (the election outcome is correct and includes all valid votes), *secrecy* (a voter's vote cannot be seen by any other voter),

receipt-freeness (no evidence is given to the voter that can be used in order to disclose his/her vote to another party), *uncoercibility* (protection from outside enforcement of opinion), *fairness* (the outcome of the election is made public only after all votes have been received and tallied), *verifiability* (all critical stages of the election process are logged for auditing and the election outcome can be verified by the voters), *verifiable participation* (the participation of a voter can be checked by the election authority, in cases where voting is compulsory), and *robustness* (the election process cannot be hindered either accidentally or on purpose by outside intervention). Given these definitions, we can define the means by which the trust prerequisites, i.e. the word “dependably” above, can be satisfied:

Trust management/engineering is a unified approach to interpreting, specifying and incorporating security requirements in a transparent way that allows direct authorization of security-critical actions on behalf of the user.

Thus, this applied view of trust, as pertaining to the eVoting domain, is a property of an eVoting system that emerges in citizens’ minds as a result of a systematic process and manifests itself in their will to use the system in order to participate in an election. This emergence is made possible through the proper trust engineering approach. This approach has been applied to the design and development of the eVoting system described below.

3 The Trust-Centered Approach

Our approach relies on two general methodologies and one cryptographic eVoting protocol. The two methodologies are the *layers of trust decomposition* of a system (see [12], [13]) and the *CORAS risk assessment framework* (see [22]). The eVoting protocol is the protocol described by Warren Smith in Section 7.3 of [20] which is based on the homomorphic properties of the El Gamal encryption function (see [14] for details on this function). Below we will provide a brief account of these three elements, which are shown in Figure 1.

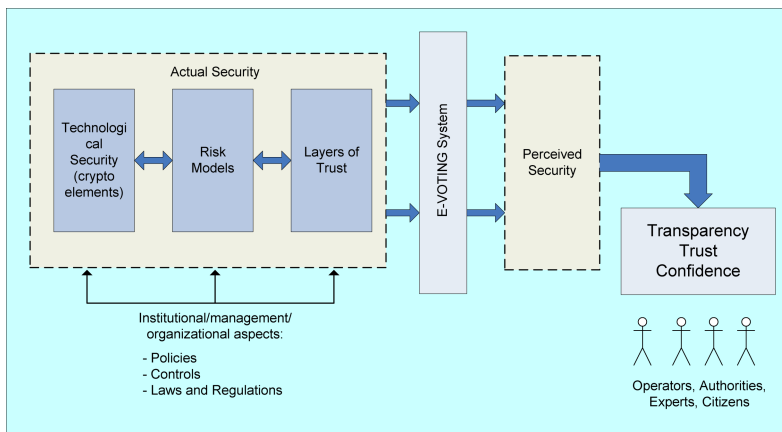


Fig. 1. The trust-centered approach

3.1 Layers of Trust

The layers of trust view of the eVoting system is a view complementary to the other formal views and models of ordinary IT systems (e.g. business view, technical view etc.) and is employed in order to handle the complexity of the security issues pertaining to eVoting, as defined by its security requirements. This complexity can be as high as the complexities that arise in other architectural views of such systems and the layers of trust approach can be used as a tool for managing these issues successfully.

The role of the layers, and the correspondence to the e-voting system, is as follows:

1. *Scientific soundness*: All the components of the system should possess some type of security justification and be widely accepted within the scientific community. This layer corresponds to the selection of a cryptographically strong eVoting protocol, based on provably secure cryptographic primitives, such as the El Gamal encryption scheme and zero knowledge proofs.
2. *Implementation soundness*: A methodology should be adopted that will lead to the verification of the implementation of the separate system components as well as the system as a whole. In addition, such a verification methodology should be applied periodically to the system. This layer corresponds to the adoption of the CORAS methodology (see below) for designing and building the eVoting system.
3. *Internal operation soundness*: The design and implementation should offer high availability and fault tolerance and should support system self-auditing, self-checking, and self-recovery from malfunction. Interference from the inside with the normal operation of the system should be, ideally, impossible to accomplish and, if ever accomplished, it should be readily detectable. The employment of the cryptographically secure eVoting protocol involves the use of proofs of correctness for all the executed steps.
4. *Externally visible operational soundness*: It should be possible for everyone to check log and audit information at some level. The employed cryptographic protocol employs a number of publicly accessible bulleting boards where information is appended concerning the votes cast as well as the proof that the votes were taken into consideration for the computation of the vote outcome.
5. *Convincing the public (social side of security)*: It is crucial for the wide acceptance of the eVoting system that the public will trust it when it is in operation. This trust can be, in general, amplified if the eVoting authority publicises the details of the design and operation of the eVoting system to the public. There is provision for publicizing all the details of the system architecture and implementation as well as provide the software source code for scrutiny. In addition, in order to facilitate the system's wide acceptance, the first trials will be conducted on a voluntary basis with closed groups or local associations, whose opinions can be easily gathered and analyzed.

3.2 Choosing CORAS as the Risk Assessment Framework

CORAS is a framework that permeates the design process in all the layers described above and aims at the precise, unambiguous, and efficient risk assessment of general

security critical systems, during their design, implementation and operation phases. The framework focuses on the integration of viewpoint-oriented UML-like modelling in the risk assessment process. The integration of this state-of-the-art modelling technology in the risk assessment process - referred to as model-based risk assessment - is motivated by the need for cost reductions, efficiency improvement and improved quality of risk assessment results. To achieve its goals, CORAS employs a variety of risk analysis methods, including failure modes, effects and criticality analysis (FMEA/FMECA), fault tree analysis (FTA), Hazard and operability analysis (HaZOP), Cause Consequence Analysis (CCA), Markov analysis etc. In addition, CORAS can produce detailed system documentation and a system security policy based on the outputs of the tools that it employs. This documentation can be publicized in order to increase the transparency of the implementation process of the eVoting system leading, thus, in its wider acceptance by technical and non-technical people alike. Moreover, this documentation provides an open view of the system to the public, in contrast with most “closed-design” commercial eVoting systems.

There is a number of other general approaches to model-based risk assessment include, for instance, CRAMM and Common Criteria among the most widely used ones. The particular angle of the CORAS approach with its emphasis on security and risk assessment tightly integrated in a UML and RMODP is however new. In particular, the issue of maintenance and reuse of assessment results has received very little attention in the literature. Since 1990, work has been going on to align and develop existing national and international schemes in one, mutually accepted framework for testing IT security functionality. The Common Criteria (CC) [10] represents the outcome of this work. The CC is generic and does not provide methodology for security assessment. CORAS, on the other hand, is devoted to methodology for security assessment. Both the CC and CORAS place emphasis on semiformal and formal specification. However, contrary to the CC, CORAS addresses and develops concrete specification technology addressing security assessment. The CC and CORAS are orthogonal approaches. The CC provides a common set of requirements for the security functions of IT systems, as well as a common set of requirements for assurance measures applied to the IT functions of IT products and systems during a security evaluation. CORAS provides specific methodology for one particular kind of assurance measure, namely security risk assessment.

The *Risk Analysis and Management Methodology* (CRAMM) was developed by the British Government’s Central Computer and Telecommunications Agency (CCTA) as a structured and consistent approach to computer security management (<http://www.cramm.com/>). The UK National Health Service considers CRAMM to be the standard for the risk analysis of information systems within healthcare establishments. CRAMM is an important source of inspiration for CORAS, and aspects of CRAMM have been incorporated in CORAS. Contrary to CRAMM, CORAS provides a risk analysis process in which modelling is tightly integrated with the process, not only to document the target system, but also to describe its context and possible threats. Moreover, CORAS employs modelling to document the results from risk analysis and the assumptions on which these results depend.

3.3 Voting Protocol

With regard to the eVoting protocol that is employed, it is based on strong cryptographic primitives, including zero-knowledge proofs that, essentially, provide the guarantees (without violating the vote secrecy requirement) that votes are correctly received and included in the voting outcome. The protocol (see Section 7.3 of [20]) is based on multiparty computations and threshold cryptography, involving mutually distrusting agents who control the voting process.

There are four main entities involved in the protocol: the *Election Authority*, the *Voter*, the *Key Holders*, and the *Bulletin Boards*. The Election Authority is responsible for interacting with the Voter in order to obtain his/her vote in encrypted form. The encryption uses a publicly known key that is formed by the Key Holders using a jointly computation on their private keys. The encrypted vote is then re-encrypted with the authority's secret key, to prevent disclosure of the vote from the voter (e.g. for selling the vote). At the same time, the Election Authority provides the voter with zero knowledge proofs for the vote's re-encryption validity/integrity while timestamping the vote in order to allow the voter to cast multiple votes, with only the last vote being the one that will be included in the vote count (so as to avoid vote coercion). The Bulletin Boards are employed for making available to the public all the details of the interaction between voters in order to support a voting process with all information flow transparent and readily available to all involved parties.

The protocol, as described in [20], leaves many implementation issues open, for which our project team should make choices as early in the project as possible. Although some of these issues have not been determined yet completely, some decisions have already been made. For instance, all Voters should go through an initial stage of registration and authentication using a PKI (either an already established PKI or one operating for the election alone). The Voters are allowed to be authenticated using a simple username/password combination, a smart card or a secure hardware token. In addition, the Election Authority actually monitors and controls a number of distributed local authorities that form a network of vote gathering and processing elements operating in parallel and in a high availability, replicated configuration. Also, the Key Holders are implemented using a number of strong cryptographically secure random number generators (both hardware and software) that form their keys privately (on separate machines) and then perform a secure distributed computation on their private keys in order to produce the election key. Timestamping is also an important, as well as difficult to handle, issue. Our project is considering a number of solutions, including the employment of reputable timestamping service providers or even GPS timing information (obtained by all distributed authorities independently). All the design and implementation details will be made available in a future report of our project.

With regard to our eVoting project status, it is in the detailed design phase. The architectural design and the first steps of the CORAS methodology have been accomplished in conjunction with the system decomposition into the layers of trust, currently focusing on the scientific soundness layer (eVoting specific protocol).

4 Architectural Aspects

In this section we will provide a high level view of the architecture of the eVoting system that is based on the approach outlined in Section 3.

In Figure 2.a we see the overall system's architecture. It consists of a number of *local Election Authorities* (local EAs), which control the election process at a local (e.g. municipality) level, a *central Election Authority*, which controls all the local EAs and verifies their operation, a *VPN over the Internet* that handles the communication among the EAs and the *clients*, which are the computers accepting the votes. In the same figure, also appear the entities that may attempt interference with the system since, by taking the worst case scenario, we assume their existence and their will to attempt disruption of normal operation.

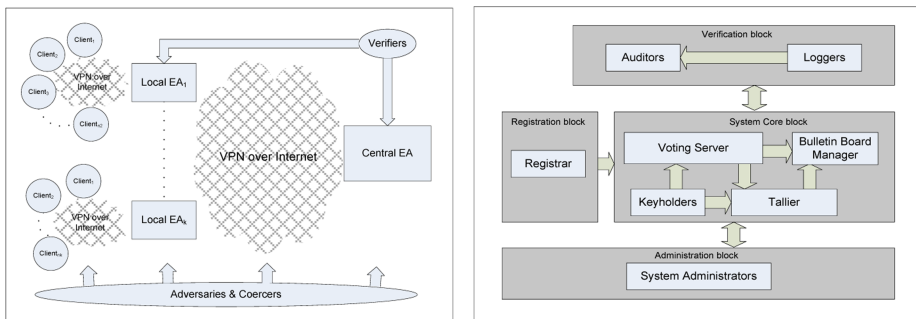


Fig. 2. (a) The distributed architecture of the eVoting system (b) The EA block

In Figure 2.b the components of an EA are shown. Each EA implements, at its core, the eVoting protocol described in [20], which has guaranteed strong cryptographic properties. The components of an EA are the following (most of which directly dictated by the protocol): the *registrar*, which is responsible for checking the voter's eligibility through a connection to a database server containing the id's of eligible voters, the *voting server*, which accumulates and verifies the votes sent by the clients over the VPN, the *key holders*, which cooperatively provide the critical vote encryption key, the *tallier*, which sums the votes and provides the election total, the *bulletin board manager*, which makes publicly available proofs that all votes are taken into account unchanged, the *loggers*, which store critical information about the election process, and the *auditors* which use the information stored by the loggers in order to provide publicly verifiable proof of correctness of the election process. Finally, there is the system administration block that is responsible for the configuration, initialization and coordination of all the other blocks.

As an example of the application of CORAS in the design phase, the Table 1 below shows a fragment of the security critical assets we have identified using HAZOP:

Table 1. Security critical assets of the eVoting system identified by HAZOP

Asset	Description	Entities Involved
Voters List	Contains the voters which are eligible to vote.	EA
Candidates List	Contains the candidates' credentials or alternatively the offered choices for a referendum.	EA
Voter Credentials	The information required for a voter to be identified and authenticated by the eVoting system.	EA, EA _i ,Voter
Configuration Files	Contains information that defines issues such as the opening and closing time of the voting process, the ballot format, etc.	EA, EA _i
Voting opening and closing announcements	Messages that control the opening and closing of the eVoting.	EA, EA _i
Random generated numbers used in key generation	Numbers that must be provably random.	EA, EA _i , Voter
Encryption/Decryption Keys	Decryption and encryption keys must be produced under strict integrity constraints. Decryption keys must remain secret, safe and unaltered throughout the whole eVoting process.	EA, Key holders
Empty ballot form	The form that a voter must fill in order to submit a vote.	Voter
Encrypted and Re-encrypted vote	The message containing the vote is sequentially encrypted by the voter and the EA _i , and consequently verified by both for its integrity and time of submission.	Voter, EA _i ,
ZKPs	Most of the entities in the system provide Zero Knowledge Proofs in order for their actions to be verifiable.	Voter, EA _i , EA, Key holders

Table 1.*(continued)*

Asset	Description	Entities Involved
Multiple votes	The proposed eVoting system supports the submission of multiple votes per user. Only the final vote is valid.	Voter

With regard to the implementation choices, we have adopted the use of as many free and open source libraries as possible. Our choices include the Java programming language, the use of the Bouncy Castle Java crypto library ([http:// www.bouncycastle.org/](http://www.bouncycastle.org/)), Open VPN (<http://openvpn.net/>), OpenCA tool for building PKIs (<http://www.openca.org/>), and the use of the PostgreSQL (<http://www.postgresql.org/>) data base. This ensures that the system's software can be independently audited and verified by any interested third party (government agencies, expert groups, researchers, industry etc.).

5 Conclusions

In this paper we have described a framework that can be applied to the design and implementation of eVoting systems in order to achieve increased trust from the citizen's side (perceived security). This approach relies on the layers of trust decomposition of the system, on the CORAS risk management methodology and on the choice of cryptographically strong eVoting protocols. The goal of the layers-of-trust approach is, mainly, to handle in a structured way the complexity of the security issues that beset all security critical applications. The focus is on designing and building the application in a transparent way that produces a sufficient and verifiable security level at each layer, able to establish and maintain trust in all involved agents: technical people, government and the people who will use the system. The goal of the CORAS methodology is to assure that all threats to the system are discovered in time, before the deployment of the system, and to provide sufficient documentation of the system that can be made publicly available. Finally, the cryptographic protocol (any other protocol could be used in its position) assures that all the basic requirements of eVoting are secured, at least in principle.

We believe that this "three-pillar" systematic approach can lead to the design and development of eVoting systems that can "prove themselves" in the citizens' eyes providing evidence for their reliable and secure operation. Of equal importance to the wide acceptance of the system, is the demonstration of its secure operation within the context of elections within small, closed groups on a voluntary basis and a gradual deployment to a larger scale.

We should, however, stress the fact that our approach to trust does not cover non-engineering issues. For instance, our approach does not address the issue of how a citizens' right to verify that his/her vote was included in the final voting result can be exercised, although there is some piece of evidence (digital or paper-based) that is provided to all voters that can be potentially used for verification purposes. We believe, however, that the proposed approach could be extended in order to address all

these issues (such as, for instance, by appointing external system and eVoting process evaluator experts), beyond the engineering level, in order to enable citizens reach a trust level similar to the trust level enjoyed by the conventional voting procedure.

References

1. van Acker, B.: Remote e-Voting and Coersion: A risk Assesemnt Model and Solutions. In: *Electronic Voting in Europe - Technology, Law, Politics and Society*. LNI Proc., pp. 53–62. GI-Editions (2004)
2. The problems and potentials of voting systems: *Communications of the ACM*, Special Issue on eVoting 47(10) (October 2004)
3. Blaze, M., Feigenbaum, J., Lacy, J.: Decentralized trust management. In: *Proc. IEEE Symposium on Security and Privacy*, Oakland, CA, USA, pp. 164–173 (1996)
4. Coleman, J.S.: *Foundations of Social Theory*. The Belknap Press of Harvard University Press, Cambridge, MA (1990)
5. Eschenauer, L., Gligor, V.D., Baras, J.S.: On trust establishment in mobile ad-hoc networks. In: *Proc. Security Protocols Workshop*, Cambridge, UK, pp. 47–66 (2002)
6. Gritzalis, D.A.: *Secure Electronic Voting*. *Advances in Information Security*, vol. 7. Kluwer Academic Publishers, Dordrecht (2003)
7. Guha, R., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of trust and distrust. In: *Proc. International Conference on World Wide Web*, pp. 403–412 (2004)
8. Hubaux, J.-P., Buttyan, L., Capkun, S.: The quest for security in mobile ad hoc networks. In: *Proc. ACM International Symposium on Mobile ad-hoc networking and computing*, pp. 146–155. ACM Press, New York (2001)
9. Josang, A.: The right type of trust for distributed systems. In: *Proc. New Security Paradigms Workshop*, pp. 119–131 (1996)
10. Information technology security evaluation criteria (ITSEC): version 1.2, Office for Official Publications of the European Communities (June 1991)
11. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The eigentrust algorithm for reputation management in p2p networks. In: *Proc. International Conference on World Wide Web*, pp. 640–651 (2003)
12. Konstantinou, E., Liagkou, V., Spirakis, P., Stamatiou, Y., Yung, M.: Electronic National Lotteries. In: Juels, A. (ed.) *FC 2004*. LNCS, vol. 3110, pp. 147–163. Springer, Heidelberg (2004)
13. Konstantinou, E., Liagkou, V., Spirakis, P., Stamatiou, Y., Yung, M.: Trust Engineering: from requirements to system design and maintenance – a working national lottery system experience. In: Zhou, J., Lopez, J., Deng, R.H., Bao, F. (eds.) *ISC 2005*. LNCS, vol. 3650, pp. 44–58. Springer, Heidelberg (2005)
14. Lenstra, A.K., Lenstra, Jr., H.W.: Algorithms in number theory. In: van Leeuwen, J. (ed.) *Handbook of Theoretical Computer Science*, vol. A, pp. 673–715. North-Holland, Amsderdam (1990)
15. Luhmann, N.: Familiarity, confidence, trust: Problems and alternatives. In: Gambetta, D. (ed.) *Trust: Making and Breaking Cooperative Relations*, pp. 94–107. Blackwell, Oxford (2000)
16. OASIS Standard: EML Process and Data Requirements, ver. 4.0 (February 2006)
17. Observatory for the Greek Information Society: *Remarks and Conclusions for the penetration of broadband in Greece and Europe (1st semester 2007)*

18. Richardson, M., Agrawal, R., Domingos, P.: Trust management for the semantic web. In: Proc. International Semantic Web Conference, pp. 351–368 (2003)
19. Reinhard Riedl: Rethinking Trust and Confidence in European E-Government, White paper
20. Smith, W.D.: Cryptography meets voting (September 2005)
21. Kim, D.J., Song, Y.I., Braynov, S.B., Rao, H.R.: A multidimensional trust formation model in B-to-C e-commerce: a conceptual framework and content analyses of academia/practitioner perspectives. *Decision Support Systems* 40, 143–165 (2005)
22. Stølen, K., den Braber, F., Dimitrakos, T., Fredriksen, R., Gran, B.A., Houmb, S.-H., Stamatiou, Y.C., Aagedal, J.Ø.: Model-based risk assessment in a component-based software engineering process: the CORAS approach to identify security risks. In: Barbier, F. (ed.) *Business Component-Based Software Engineering*, pp. 189–207. Kluwer, Dordrecht (2003)
23. Tassabehji, R., Elliman, T.: Generating citizen trust in e-government using a trust verification agent: a research note. In: CD-ROM/Online Proceedings of the European and Mediterranean Conference on Information Systems (EMCIS) 2006, Costa Blanca, Alicante, Spain (2006)
24. Theodorakopoulos, G., Baras, J.S.: Trust evaluation in ad-hoc networks. In: Proc ACM Workshop on Wireless security, pp. 1–10. ACM Press, New York (2004)