"Trust Engineering:" From Requirements to System Design and Maintenance – A Working National Lottery System Experience

Elisavet Konstantinou[1,2], Vasiliki Liagkou[1,2], Paul Spirakis[1,2], Yannis C. Stamatiou[1,3], and Moti Yung[4,5]

[1] Computer Technology Institute, P.O. Box 1122, 26110 Patras, Greece
[2] Dept. of Comp. Eng. and Informatics, University of Patras, 26500 Patras, Greece
{konstane,liagkou,spirakis}@ceid.upatras.gr
[3] Dept. of Mathematics, University of Ioannina, 45110, Ioannina, Greece
istamat@cc.uoi.gr
[4] Computer Science, Columbia University, New York, NY, USA
moti@cs.columbia.edu
[5] RSA Laboratories, Bedford, MA, USA

Abstract. Based on our experience in designing, building and maintaining an information system for supporting a large scale electronic lottery, we present in this paper a unified approach to the design and implementation of electronic lotteries with the focus on pragmatic trust establishment. This approach follows closely the methodologies commonly employed in the development of general information systems. However, central to the proposed approach is the decomposition of a security critical system into layers containing basic trust components so as to facilitate the management of trust, first along the layers, and then as we move from layer to layer. We believe that such a structured approach, based on layers and trust components, can help designers of security critical applications produce demonstrably robust and verifiable systems that people will not hesitate to use.

Keywords: Electronic Lotteries, Security Critical Applications, Trust.

1 Introduction

Trust is a concept that plays a major role in the way people view and use information systems (especially financial applications). No matter how sophisticated or expensive an information system is, if people do not trust it, the system is bound to fail to deliver the services for which it was designed and built.

In most of the information systems that deliver e-services, trust is not based on some systematic design process but, rather on the reputation of the system's main stakeholder (e.g. lottery organization in the case of e-lotteries or bank in the case of e-banking). In particular, electronic lotteries are involved in the management of potentially huge amounts of money and thus, security *and* trust should be the top priorities. If the lottery fails then a great amount of money may be lost by the lottery operator and, which may be worse, the players lose their trust in the system. Building systems in a way that attracts users is at the basis of the success of any information system and the problem of building such systems is the focus of our paper. However, we should stress that we do not attempt to formalize the notion of trust in this work, but rather to base its emergence on a systematic design and implementation approach.

Regarding the e-lottery domain (that was the target of the application of the proposed methodology), many e-lottery and e-gambling protocols and systems have been proposed in the past. In [3] a lottery scheme is described that uses multiple dealers. In [4] the lottery uses delaying functions and places an upper bound on the number of tickets that a player can buy. E-casinos with secure remote gambling are described in [6], while in [7] an internet based lottery is presented that exploits blind signatures and hash chain techniques. The lottery in [9] uses as primitives a bit-commitment scheme and a hash function and it is suitable for a large-scale Internet operation (but it is essentially a protocol for Internet betting rather than lottery). In the protocol described in [11] users can participate in the process of the number draw and they must make perform some computations to see if they have won or not. In [21] an internet based lottery is presented, which uses the played coupons to generate the winning coupon. The protocol in [22] is based on a bit-commitment scheme and the winning coupon can be read by the players only after a predetermined amount of time has elapsed. The main features of the protocol in [24] is the preservation of the anonymity of the players and the existence of a mechanism for paying the winners. In [10], a national e-lottery system was presented using a protocol that starts from the generation of the winning numbers and ends with their verification and public announcement.

The system of [10] is in successful operation for over two years now. In this paper we draw on the experiences from the design and implementation of the system as well as its operation and maintenance. Based on these experiences, we propose a *trust preserving* approach for handling the increasingly difficult complexity issues of building trustworthy electronic lottery systems and, in general, any financially risky application. Most often, technical papers concentrate on the technical issues that support trust, mainly the use of cryptographic primitives or protocols. However, to the best of our knowledge, there are no approaches that analyze the trust from a *technological*, *policy* and *public awareness* point of view, based on a "trust life cycle" of a system that includes the design as well as the operation and maintenance phases.

Our approach is *pragmatic*, i.e. it does not target definitional issues pertaining to trust, which is a concept hard to define and any attempt to do so may lead to philosophical controversy. Trust on a pragmatic level, in our perspective, consists, simply, of "assuring satisfactory implementation and operation of all system components in a way that ensures compliance with their requirements and specifications and its demonstratability". *Trust engineering*, in turn, consists of "handling the means, issues, processes, components and subsystems that contribute directly to achieving pragmatic trust". The goal of our paper is to propose design guidelines encompassing those two aspects, based on the experience we gained with the design, implementation and operation of a highly critical e-lottery system.

The rest of the paper is organized as follows. In Section 2 we motivate our approach and show how it relates to the system described in [10]. In Section 3 we discuss the lowest level of trust, which is based on cryptographically secure primitives. In Section 4 we move to the next level where the primitives and protocols are actually implemented and integrated as a system. In Section 5 we discuss the trust layer that has to do with the internal operation procedures of the main system stakeholder (lottery operator). In Section 6 we discuss elements of trust that are related to how the electronic lottery is guarded against attempts of fraud. In Section 7 we focus on people education and awareness issues. Finally in Section 8 we summarize our approach and argue for its generality.

2 Trust Engineering and Pragmatic Trust

The establishment of trust in a security critical system can be achieved along two directions: (i) by treating the system as an integrated software/hardware application and applying methodologies that ensure correctness during all phases of its life cycle (*trust engineering* direction), and (ii) by decomposing the system into different architectural layers that include its environment, the users, the owners, and, generally, all technical and social issues that interact with it (*pragmatic trust*).

In order to integrate these two directions and handle them in a unified manner, we propose a general trust building methodology which we will explain in detail later. The first direction can be handled using approaches that are frequently applied to information systems in general. The approach we propose is comprised of the following phases:

1. System initiation: define the system in general, evaluate risks, identify and rank consequences of failure, estimate impact of known attacks.
2. Trust requirements and specifications: specify the functionality of the system, define its operational capabilities, establish the desired performance characteristics, isolate the critical functions that should be guarded against attacks at all costs, define the critical system transactions, build the capability to demonstrate good behavior and to detect and eliminate attacks, provide facilities for attack recovery.
3. Trust design components: specify the overall software and hardware architecture, design the data flows, develop threats, adversarial models, trust structure and strategy, design network facilities for replication of critical assets, establish the quality of algorithms used, ensure isolation and availability of the critical system components.
4. Trust component construction and testing: code the system components, build the infrastructure, verify correctness and safety, revisit trust maintain-

ing mechanisms, establish logging facilities and scrutiny procedures, take accountability measures.

5. Operation and maintenance: install the system, maintain trust handling both, social and external issues, handle security and safety maintenance, establish a continual evaluation process.

Based on the five phases mentioned above, we developed a methodology, which was applied to the development and operation of a large scale e-lottery system. We assume that the prior trust engineering tasks have been completed successfully and the appropriate trust requirements and specifications have been established. The task of the initial architecture definition of the system is to define a set of candidate trust design components and their interrelations. Then in the implementation phase, we construct the specified services that satisfy both the functional and non-functional trust requirements and specifications. As we move from phase to phase, we add protection mechanisms (appropriate for each phase) that ensure the correctness of the system as defined in every phase.

Moreover, our central point of view is that the *pragmatic* approach to security-critical applications should be based on *layering*. The layered approach to trust reflects the above system phases by combining the *technology*, *policy* and *public awareness* issues of a trusted system. A variety of tools and techniques can be applied to each of the layers ensuring that every layer satisfies the trust requirements and specifications. The layers-of-trust approach that we propose can also be adjusted so as to be applicable to other security and trust critical applications. This is due to the fact that our focus is not on e-lottery related technology issues but on policy as well as user awareness, which concern a great variety of systems. We demonstrate that these three elements can be combined and lead to a system design and implementation that can be demonstrably trustworthy and, thus, have a market success for the owner.

Our layers-of-trust approach is the outcome of the design and implementation of a national electronic lottery system which is already in full scale operation for more than two years ([10]). In this system, the players buy their coupons from 6000 certified lottery agencies selling coupons across the country. The winning numbers are chosen at random from within a certain range. The coupons with the number choices of the players are dispatched to a central computer placed within the lottery organization and are stored in a *special database*. Moreover, we assume that the winning numbers are generated at specific draw times every day. At some specific point just before the winning numbers are generated, people are not allowed to play anymore while, obviously, they should not be allowed to play after the current draw is over (post-betting). Finally, after the winning numbers have been generated, they are sent over to the computer that stores all the played coupons, so that the winners can be selected and various statistics calculated.

In Figure 1 a decomposition of the e-lottery application is shown in terms of layers of the trust architecture. The role of the layers is as follows (the details will be given in the relevant sections):

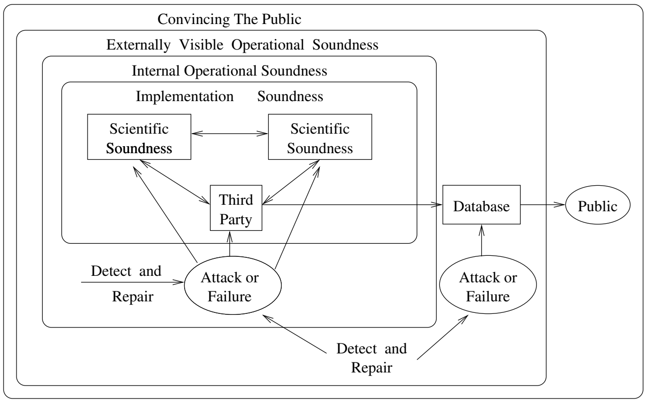

Fig. 1. The layers of the architecture

1. Scientific soundness: All the components of the system should possess some type of security justification and be widely accepted within the scientific community. For instance, the random number generation components of the system should be *cryptographically secure*, i.e. hard to predict.
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.
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. It should be difficult to interfere, from the inside, with system data (either log, audit, or draw information stored in the system's database) and if such an interference ever occurs, it should be detectable. Also, there should be a trusted third party that can verify that the numbers were produced without external intervention.
4. Externally visible operational soundness: It should be impossible for someone to interfere with the normal operation of the lottery system from the outside. If such an interference is ever effected, it should be quickly detectable.
5. Convincing the public (social side of trust): It is crucial for the market success of the lottery that the public trusts it when it is in operation. This trust can be, in general, boosted if the lottery organization makes available the details of the design and operation of the lottery system to public bodies that organize campaigns in support of security and fairness of the lottery.

The goal of our layers-of-trust approach is, mainly, to handle in a structured way the complexity of the security threats that beset modern, high risk financial applications. The focus is on designing and building the application in a fashion that will establish a sufficient and verifiable (demonstratable) security level at each layer that, in turn, is capabale of maintaining the trust in all involved agents: technical people, stakeholders and the general public who will bet using the electronic lottery. In the sections that follow we explain in detail the layers of the trust architecture and how it can be implemented in the case of electronic lottery systems.

3 Scientific Soundness

The most important requirement for an electronic lottery is the *unpredictability* of the generated winning numbers. In order to ensure the fairness and consequently the reliability of the system, all the components of the random generation must be provably random. Randomness can be achieved, in general, through the following three approaches: (i) using a *pseudorandom number generator*, (ii) digitizing noise from a physical process, thus obtaining *true random number generators*, and (iii) using a combination of approaches (i) and (ii). Approach (iii) seems to be a good tradeoff between the deterministic nature of approach (i) and the potentially bias of the noise used in approach (ii) and we believe that it should be followed in an e-lottery design.

3.1 Pseudorandom Generators

The winning number generation process should employ a number of generators based on *different* design and security principles so that breaking some, but not all, of the generators would not imply breaking the whole generation process. All the algorithms used should be reconfigurable in terms of their parameter sizes. These sizes can be changed at will as frequently as necessary in order to keep up with cryptanalytic advances.

There are, in general, two pseudorandom number generator types: (i) generators whose security is based on the difficulty of solving some number theoretic problem, and (ii) generators employing symmetric (block) ciphers or secure hash functions. Both types of generators are considered cryptographically strong.

Generators belonging to the first type produce number sequences that can be proved to be indistinguishable from truly random sequences of the same length, by any polynomial time statistical test. The use of such *robust* generators handles the basic requirement of assuring randomness under various cryptanalytic advancements and operational attacks. Three such generators that can be used in an e-lottery are the BBS, RSA/Rabin and Discrete Log generators (see [10]). The BBS generator, named after the initials of its inventors Blum, Blum, and Shub (see [2]) is one of the most frequently used cryptographically strong pseudorandom number generators. It is based on the difficulty of the Quadratic Residue problem. The RSA/Rabin generator is cryptographically secure under the RSA function assumption (see [1]) and the Discrete Log generator is based on the difficulty of the discrete logarithm problem (see [18]).

Generators of the second type are built from secret key ciphers and secure hash functions and are not, usually, accompanied by a formal proof of cryptographic security. However, they are assumed to be strong because any statistical deviation from truly random sequences would mean a cryptanalytic weakness of the corresponding secret key cipher or hash function. DES and AES (see [14] for a library that contains their implementations) or the hash functions SHA and MD5 are good candidates from this generator class. However there should be an ongoing effort of keeping up-to-date with cryptanalytic advances of hash functions and revisit, accordingly, their use in the production of random numbers or data integrity.

In order to confuse cryptanalysts, the generation process can periodically and unpredictably use different combinations of algorithms for the generation of the winning numbers. For example, two shuffling algorithms that can be used are Algorithm M and Algorithm B, described in detail in Knuth's book [8]. In addition, the XOR operation can be used, which can mix the output of two (or more) generators.

3.2 Physical Random Number Generators

It is necessary to have some true random bits in order to start the software generators. These initial values, or seeds, of any software random number generator must, eventually, be drawn from a source of true randomness. Thus, one needs hardware generators relying on some physical process. Moreover, it is important to combine more than one such generators to avoid problems if some (but not all) of the generators fail.

Some well-known hardware-based random number generators are: (i) a generator based on the phase difference of the computer clocks. It is implemented as a function called VonNeumannBytes() (by Adam L. Young), (ii) a commercial device by Westphal called ZRANDOM (see [23]) placed on an ISA slot of a PC, and (iii) a serial device by Protego called SG100, see([19]). The outputs of the generators can be combined using the XOR operation. One may rectify deviations of the generators using the Naor-Reingold pseudorandom function (see [16]) for processing the combination of the seeds that are obtained from the physical random number generators.

3.3 Mapping Random Bits to the Required Range

A problem that arises in all electronic lotteries is how the random bits (0s and 1s) that are generated by the pseudorandom generators can be mapped to integers within a specified range (the winning number range). Having a variety of possible ranges ensures the generality of the number generation system which, in turn, increases the range of potential applications that may employ it. This variability of design should be part of any design approach based on trust. What is important is that the mapping should be done in a way that preserves randomness.

The solution that we propose is simple. Let the range of winning numbers be from 1 to r ($r > 1$). Let n be the unique number such that $2^{n-1} < r \leq 2^n$. Then the output bits of the generation process are grouped into n bits at a time, forming an integer. Thus, there are $2^n - r$ out of range integers. Every such integer has to be mapped to r integers uniformly at random with equal probability ($\frac{1}{r}$). If the formed integer is larger than r, n bits are drawn from another, independent generator in order to select a number within the allowed range. After this is done, we return to the first generator.

4 Implementation Soundness

Building cryptographically secure blocks is certainly an important security requirement but it is not the only one. This is because the theoretically established cryptographic security by itself disappears if a single implementation error occurs in the implementation code. Testing the implementation of the cryptographically secure generators is a crucial step in the efforts to build a *secure* and *trustworthy* electronic lottery system. There is a number of verification methodologies and tools that can be applied, that are based on various statistical tests. These tests will be described below.

Moreover, in order to assure that there are no backdoors inserted by the designers in the implementation code, a properly accredited body (either person or organization) can be employed to thoroughly examine and verify the implementation by signing, in the end, a certificate. Measures for tamper proofing the software should be taken as well.

4.1 Verifying Randomness Using Statistical Tests

Although no software/hardware tool claims (or can possibly claim, on theoretical grounds) that it can perfectly detect (non)randomness, the randomness test platforms that have appeared in the literature are in position to capture bad implementations through the application of well-established and easy to apply statistical tests. Some very popular software platforms that provide randomness tests are the Diehard platform [13], CRYPT-X [5], and the platform of the National Institute of Standards and Technology (NIST) of USA [20]. We propose the use of the Diehard platform because of its high portability and adaptability.

The Diehard platform, that was proposed and programmed by Marsaglia, includes a number of powerful and widely accepted statistical tests for testing randomness of sequences of integers. The randomness tests applied by the Diehard platform are indicative for the randomness of the generator under analysis and, in practice, no generator can be claimed random if it does not pass the Diehard tests.

4.2 Online Randomness Testing

As a precaution against future malfunction of any of the random number generators included in the electronic lottery, an on-line testing procedure can be applied on the numbers that the lottery system makes publicly available as winning numbers.

In general, statistical tests require very large amounts of data. We would like, however, on-line tests to give an indication of non-randomness with as few data bits as possible. For on-line testing, we can select the elementary 0-1 frequency test which can give meaningful results with a relatively small amount of data. Alternatively, one can always use the tests provided by the *FIPS PUB 140-2* standard published by the Federal Information Processing Standards Publication Series of the National Institute of Standards and Technology (NIST). If the results of these tests raise suspicions for non-randomness a special notification message can be sent to authorized personnel in the e-lottery organization.

5 Internal Operational Soundness

One of the most important issues in an electronic lottery system is the ability to self-check its internal operation and issue appropriate warnings when needed. Self-checking reduces human intervention and increases the responsiveness of the system in case of deviations from normal operation. Such a self-checking capability can be built in a distributed manner as described below.

The system should, preferably, be composed of *at least* two random number generation servers connected in a way that allows them to know which one is active generator and if the active one fails, then how the other one will take over. A possible approach can be based on adapting the "mon", "heartbeat", "fake" and "coda" solution given for building robust LINUX servers with distributed file systems (see [12]). The "mon" is a general-purpose resource monitoring system, which can be used to monitor network service availability and server nodes. The "heartbeat" code provides "electronic" heartbeats between two computers using the TCP/IP protocol or the serial line. "Fake" is an IP take-over software using ARP spoofing. Finally, the "coda" part handles the distributed file system, which was of no concern in the case of the electronic lottery.

We assume that the two servers are connected to the lottery owner's Intranet/VPN by means of the TCP/IP protocol and they are also interconnected through their serial ports. A possible approach to the fail over mechanism is the following: one of the two computers is initially set as the active server while the other one is the inactive. A daemon process is started on both servers (running concurrently with the random number generators) to monitor the status of the applications (random number generation) that are executed on the computer. If the inactive server does not receive a response from the active one, then a failure must have occurred in the active server and it activates the process of IP takeover in order to take the role of the failed server. In addition, if the active server detects a problem in the inactive server, it notifies the e-lottery personnel so that immediate action can be taken. All this activity is supervised by another trusted computer that is placed between the generator computers and the outside worlds. This computer mainly performs auditing and checking and protects the number generation computers from attacks.

In case of malfunction, an approval officer must be online in order to instruct the authorized personnel to repair the damage in the servers. Moreover, a personnel security plan must be deployed so that every person in the e-lottery organization is responsible for a different action.

In addition, it is important to have only authorized personnel in the computer room where the servers are kept, so that the probability of malicious interference with the machines is minimized and each such interference can be tracked down to a small set of people.

We also suggest the use of a biometric access control system to the room that houses the electronic lottery. An access control system must also use surveillance cameras that record activity on video tapes as well as display it on a screen for the security officer on duty. In addition, movement detectors should be placed on each of the computers containing the random number generation system and special vibrator detectors should be placed on the room’s walls as well as the floor and ceiling. Logging information may be viewed with the help of an Internet browser by authorized personnel, possibly the same personnel that monitors the activity within the computer room through the camera or from a video.

6 Externally Visible Operational Soundness

At any point during the operation of the e-lottery, it should be possible to detect erratic behavior or to ascertain that everything is as expected. In this section we will describe some frequently occurring e-lottery system failures and we will propose solutions. These failures cover incidents such as failure of individual computers to operate properly as well as corruption (incidental or purposeful) of the system. The failures must be detected as fast as possible in order to prevent any loss of money or damage to the reputation of the e-lottery.

6.1 Failure of the Winning Number Generator

The heart of an electronic lottery system is the generator of the winning numbers. It is highly critical for the reputation of the lottery organizer to be able to produce the required numbers when they are needed, at the expected draw time. Thus, the high-availability configuration described in the previous section contributes to this issue too.

In Section 5, we proposed the use of two random number generator servers. Under normal operation only one of them is active and awaiting the draw time in order to produce the winning numbers. The presence of a third computer is required (called *third party*) in order to implement an *automatic fail-over* configuration between the two generators. It will continuously monitor the main generator by sending it a polling signal at regular (and relatively short) intervals to see if it is active. If no acknowledgement is received within a predetermined time interval then the main generator is declared “dead” and the second generator is activated in order to perform the draw at the specified time. In the rare situation where the backup generator is also found inactive, the monitoring computer can take the role of the generator and produce the numbers.

Other possible failures of the system can be related to failures of electrical power. It is important that the lottery system is protected from this type of failures by means of Uninterrupted Power Supply (UPS) units.

6.2 System Database Damage

All the player's coupons are collected in a single system database so that the lottery organization can identify the winners and compute various statistical figures. This database should not interfere with the generator system. Thus, it should be stored in a separate computer system. Also, to preserve integrity, the database contents should, preferably, be also stored in a non-volatile storage medium such as a non-rewritable Compact Disk.

6.3 Operational Physical Security

System operators that are involved in the day-to-day management as well as upgrade or maintenance processes constitute a delicate personnel type. They are entitled to perform, virtually, any action on the system and, thus, their actions should be subjected to monitoring and logging. Measures that can help towards this direction include visual monitoring of the system as well as strict access control. In addition, there should be a strict maintenance process for modifications of any part of the system so that all know who did what to the system, when and at what time. This bookkeeping process will help to deter administrators from abusing their power as well as to detect their interference with the system if they decide to maliciously tamper with it.

6.4 Forging Coupons

It must not be possible for any player to force a coupon directly into the coupon database either just before (i.e. after the current draw is closed) or just after the draw (postbetting). In order to guard against such an incident the coupon database should be locked, after the current draw is closed. This locking can be realized by computing a hash function on the database's contents which, essentially, forbids any alteration of its contents after the computation of the hash value. In order to detect changes on the locked coupon database the third party computes a hash value of the played tickets just before the draw takes place. If a change is detected that occurred *after* the draw (which is more important than the case where insertion takes place before the draw) the third party can still perform a legitimate selection of winners by using the copy of the database that was transferred on non-volatile storage media (which does not contain the forced coupon).

6.5 "Bogus" Servers

The lottery system should be protected from intrusions from the outside network (both in the case of a VPN within the lottery organization or the Internet in case the lottery also operates through the Internet).

First of all, the generator component of the system, the two generators and the third party, must be equipped with strong firewall software. The permitted incoming and outgoing IP addresses must be confined to be the addresses of the three computers that comprise the generator part. As it was mentioned in previous sections, the third party is connected to both generator computers as well as the coupon database. Therefore, one of the goals of the third party is to operate as a firewall placed between the generator computers and the computer that stores the coupon database. Thus, if this computer is attacked, the generator system should not be affected.

Attempts to interfere with the generator computers can be detected if strict timing requirements for completion of various tasks are imposed on the generator computers. The third party is responsible for monitoring the operation of the generators. If an intruder attempts to interfere with the generator computers then the third party can detect it because the generator computers will fail to respond within the predetermined timing constraints.

In addition, it is highly desirable to be able to verify that the seeds claimed to have been used by the generator computers were actually used in the generation process. For example, an intruder may have changed the seeds that drive the generation of the winning numbers. To detect this change, we suggest the use of a *bit-commitment* cryptographic protocol that ensures that the claimed seeds were actually used. The commitment must be performed by the generators and send to the third party. The third party checks this commitment to detect any modification on the seeds. Since it is also possible to affect the winning numbers in a way other than manipulating the seeds, the third party can reproduce the winning numbers (announced by the generator computers) using the seeds (to which the generator committed) and check if the resulting numbers are the same with the numbers returned by the generator. We should note, at this point, that an alternative to our construction for seed commitment would be to use a *verifiable random function* as proposed in [15]. We choose not to include such a function mainly because of the fact that there was already, in our design, an entity (the trusted third party) that could easily take the role of verifying the correctness of the seed value and, at the same time, raise an alarm signal in case of discrepancy.

Finally, a public key cryptographic scheme can also be employed between the generator computers and the third party that will enable the identification among the three computers. This will also decrease the vulnerability of the lottery system to efforts of interference from other computers.

7 The Social Side of Trust

The attitude of people towards e-services is the attitude of the typical individual against technology: reluctance to accept it due to ignorance of the underlying principles as well as suspicion about its fair operation. In other words, people's negative viewpoint about information technology stems from the fact that information systems are presented to people as inexplicable black boxes locked in some place where they are not allowed to go and see what is going on. This holds true especially for applications that handle people's money, as it is the case with the electronic lottery. We believe that a successful trust building methodology should address, apart from design, implementation, and verification issues, the social side of trust which consists in reassuring the public that all measures have been taken in order to produce an error-free, secure and useful application. Such measures can include the following, which were also taken for the design and implementation of the electronic lottery:

1. *Trust by increasing awareness.* Our experience from the lottery project indicates that one of the best practices to fight people's mistrust against information systems is to educate them about security and data protection issues in non-technical terms. At least the black-box picture of the information system should disappear. For instance, the classical argument that randomness is not possible by an algorithm (the "state of sin" of von Neuman) as well as the mistrust stemming from the fact that system details are known only to its designers can be dealt with a series of non-technical articles that explain cryptographical security as well as how the system operates.
2. *Trust by continual evaluation and accreditation.* In order to preserve the correct operation of a system and maintain people's trust, there should be a process of continual evaluation and certification of its operation. Our view is that there should be at least on such evaluation of e-Lotteries at the design stage and right after the implementation. During its operation, there should be regular evaluations with the results publicly available.
3. *Trust by independence of evaluators.* The system should preferably be verified by experts outside the organization that developed the system. This eliminates people's suspicion that the evaluators and the organization are in some secret agreement. In the electronic lottery case two internationally accepted experts in cryptographic security as well as financial systems design were appointed. These experts evaluated both the design and the implementation, issuing formal certificates. These certificates were publicly available by the lottery organization.
4. *Trust by open challenges.* Organize open challenges (call for hackers). Although we had not time to organize a "call for hackers" event, we believe that by giving the system details to the public and calling all interested (by setting a prize too) to "break" the system's security you make people feel more comfortable using the system and remove people's their mistrust. The challenge could be, in the electronic lottery case, the correct guess of the draw numbers after, say, 20 draws in a raw.
5. *Trust by extensive logging and auditing of system activities.* It is important that logging and auditing activities are scheduled on a daily basis whose results are available for public scrutiny. This is important since it will persuade people that things are transparent with the operation of the electronic lottery and that there exist strict auditing protocols within the organization. The electronic lottery system has the capability of storing number draw data on non-rewritable CDs, producing hard-copies as well for cross-examination.
6. *Trust by contingency planning.* Failures in systems that offer e-services (especially services of a financial nature) are not acceptable and if they occur they may cause a great loss (financial or reputation) for the system operator. However, since unexpected events can always occur, it is important to

be able to handle them fast end effectively so that the system appears to people that operates normally. Care should be taken for back-up power supplies, for high-availability of critical system components, for regular backups etc. Making publicly available such contingency plans and demonstrating how the are put into action should they be ever necessary, can contribute significantly towards increasing people's trust in the system.

7. *Trust by regulation and laws.* No matter how much effort is devoted on taking all measures mentioned above, if they are not supported by suitable governmental legislation people may always think that they are not legally protected in case of malfunction of security breaches. Therefore, it is important that the system operator introduces suitable legislation for the protection of the public in case of mishaps.
8. *Trust by reputation and past experience.* The involvement of engineers and experts in a security critical project should be accompanied by credentials that prove their expertise. These credentials may, for instance, demonstrate their involvement in other, similar successful projects as well as research activities on issues related to the project.

8 Conclusions

Building systems that are demonstrably trustworthy serves two main purposes: (i) it increases people's trust in the system and, thus, reduces their reluctance to use it, and (ii) it gives the possibility to the system owner to expand the system so as to include more functionality and more capabilities than before while preserving trust. Drawing from experiences in the design and operation of a large-scale electronic lottery (with trust playing a central role in the whole project), we have presented in this paper a systematic approach towards trust building in security critical applications. This approach is based on a design process following closely the process used for building a general information system. According to this approach, the target system is decomposed into layers whose trust properties are easier to establish and demonstrate. These layers cover the trust issues of low-level cryptographic components of the system as well as trust issues of the environment of the system (i.t. people – social trust). The systematic system design and implementation approach, in combination with the system decomposition into layers of trust, provide a unified framework for integrating trust in security critical systems and their operational environment. This paradigm can also be applied to other types of security critical applications, apart from the electronic lottery to which it was first applied (e.g. e-voting), in which trust is a central issue, encompassing technology, policies and people's awareness in an harmonized, integrated manner.

References

1. W. Alexi, B. Chor, O. Goldreich, and C. Schnorr, RSA and Rabin Functions: Certain Parts are as Hard as the Whole, *SIAM J. Computing*, 17(2), pp. 194–209, 1988.

2. L. Blum, M. Blum, and M. Shub, A Simple Unpredictable Pseudo-Random Generator, *SIAM J. Computing*, 15(2), pp. 364–383, 1986.
3. P.A. Fouque, G. Poupard, and J. Stern, Sharing Decryption in the Context of Voting or Lotteries, *Proc. Financial Cryptography 2000*, LNCS 1962, pp. 90–104, Springer, 2001.
4. D.M. Goldschlag and S.G. Stubblebine, Publicly Verifiable Lotteries: Applications of Delaying Functions, *Proc. Financial Cryptography 1998*, LNCS 1465, pp. 214–226, Springer Verlag, 1998.
5. H. Gustafson, E. Dawson, L. Nielsen, and W. Caelli, *A computer package for measuring the strength of encryption algorithms*, Computers and Security, 13, pp. 687-697, 1994.
6. C. Hall and B. Schneier, Remote Electronic Gambling, in: *Proc. 13th ACM Annual Computer Security Applications Conference*, pp. 227–230, 1997.
7. W. Ham, and K. Kim, A Secure On-line Lottery Using Bank as a Notery, *CISC 2002*, pp. 121–124, 2002.
8. D.E. Knuth, *Seminumerical Algorithms*, Third Edition, Addison-Wesley, 1997.
9. K. Kobayashi, H. Morita, M. Hakuta, and T. Nakanowatari, An Electronic Soccer Lottery System that Uses Bit Commitment, in: *IEICE Trans. Inf. & Syst.*, Vol. E 83-D, No. 5, pp. 980–987, 2000.
10. E. Konstantinou, V. Liagkou, P. Spirakis, Y.C. Stamatiou, and M. Yung, Electronic National Lotteries, *Financial Cryptography 2004*, LNCS 3110, pp. 147–163, Springer Verlag, 2004.
11. E. Kushilevitz and T. Rabin, Fair e-Lotteries and e-Casinos, in *Proc. CT-RSA 2001*, LNCS 2020, pp. 100–109, Springer Verlag, 2001.
12. `http://www.linuxvirtualserver.org/HighAvailability.html`
13. G. Marsaglia, *Diehard: A Battery of Tests for Randomness*, 1996. Available at `http://stat.fsu.edu/geo`.
14. Mcrypt cryptographic library: `ftp://mcrypt.hellug.gr/pub/crypto/mcrypt`
15. S. Micali, M.O. Rabin, and S.P. Vadhan. Verifiable Random Functions. In Proc. *40th IEEE Symp. on Foundations of Computer Science*, pp. 120–130, 1999.
16. M. Naor and O. Reingold, Number-theoretic constructions of efficient pseudo-random functions, *Proc. 38th IEEE Symp. on Found. of Computer Science*, 1997.
17. P.G. Neumann, The Problems and Potentials of Voting Systems, *Communications of the ACM* **47:10**, 2004.
18. S. Patel and G. Sundaram. An Efficient Discrete Log Pseudo Random Generator. *CRYPTO 1998*, LNCS 1462, pp.304–317, 1998.
19. Protego, product information, `http://www.protego.se/sg100\_en.htm`
20. A. L. Rukhin, J. Soto, J. Nechvatal, M. Smid, M. Levenson, D. Banks, M. Vangel, S. Leigh, S. Vo, and J. Dray, *A Statistical Test Suite for the Validation of Cryptographic Random Number Generators*, Special NIST Publication, National Institute of Standards and Technology, Gaithersburg, MD, 2000.
21. K. Sako, Implementation of a digital lottery server on WWW, in: *Proc. CQRE 1999*, LNCS 1740, pp. 101–108, Springer Verlag, 1999.
22. P. Syverson, Weakly Secret Bit Commitment: Applications to Lotteries and Fair Exchange, in: *Proc. IEEE Computer Security Foundations Workshop (CSFW11)*, pp. 2–13, 1998.
23. Westphal Electronics, product information, `http://www.westphal-electronic.de`
24. J. Zhou and C. Tan, Playing Lottery on the Internet, in: *Proc. ICICS 2001*, LNCS 2229, pp. 189–201, Springer Verlag, 2001.