Trust management for public key infrastructures: implementing the X.509 trust broker

Ahmad Samer Wazan, Romain Laborde, David W. Chadwick, François Barrère, Abdelmalek Benzekri, Mustafa Kaiiali, Adib Habbal

Trust Management for Public Key Infrastructures: Implementing the X.509 Trust Broker

Ahmad Samer Wazan,[1] Romain Laborde,[1] David W. Chadwick,[2] Francois Barrere,[1] Abdelmalek Benzekri,[1] Mustafa Kaiiali,[3] and Adib Habbal[4]

[1]*Paul Sabatier University, Toulouse, France*
[2]*University of Kent, Kent, UK*
[3]*Queen's University, Belfast, UK*
[4]*Universiti Utara Malaysia, Kedah, Malaysia*

Correspondence should be addressed to Ahmad Samer Wazan; ahmad-samer.wazan@irit.fr

Received 25 July 2016; Revised 21 November 2016; Accepted 12 December 2016; Published 9 February 2017

Academic Editor: Barbara Masucci

A Public Key Infrastructure (PKI) is considered one of the most important techniques used to propagate trust in authentication over the Internet. This technology is based on a trust model defined by the original X.509 (1988) standard and is composed of three entities: the certification authority (CA), the certificate holder (or subject), and the Relying Party (RP). The CA plays the role of a trusted third party between the certificate holder and the RP. In many use cases, this trust model has worked successfully. However, we argue that the application of this model on the Internet implies that web users need to depend on almost anyone in the world in order to use PKI technology. Thus, we believe that the current TLS system is not fit for purpose and must be revisited as a whole. In response, the latest draft edition of X.509 has proposed a new trust model by adding new entity called the Trust Broker (TB). In this paper, we present an implementation approach that a Trust Broker could follow in order to give RPs trust information about a CA by assessing the quality of its issued certificates. This is related to the quality of the CA's policies and procedures and its commitment to them. Finally, we present our Trust Broker implementation that demonstrates how RPs can make informed decisions about certificate holders in the context of the global web, without requiring large processing resources themselves.

1. Introduction

The need to identify our partners on the Internet constitutes one of the major challenges in ensuring trust on the Internet. However, multiple recent stories show that such an objective is far from being reached.

On the 18th of February 2015, a security expert published an image on Twitter [1] showing that Superfish is delivering the certificate of Bank of America, instead of Verisign (Figure 1). Supposedly, Lenovo integrated Superfish software in some of its PC models, in order to inject advertisements related to Google search results for users of IE and Chrome web browsers. Doing this, Superfish can in fact intercept any encrypted traffic of Lenovo users. To solve this issue, Lenovo has issued a guide that helps users to remove Superfish [2].

In a similar way and more recently (Nov 23, 2015), another security expert showed how Dell has shipped computers that make their future owners vulnerable to MITM attacks [3]. He showed that Dell has injected a root CA called eDellRoot in two models of PCs along with its private key. The expert explained that anyone could extract the private key and use it to sign falsified certificates that will be accepted transparently by the Dell PCs having the eDellRoot CA. Dell has provided an official solution to remove the root CA as well as its private key [4].

In both stories, the solution is to remove the CA and/or software from the concerned computers. However, nothing prevents similar stories appearing again in the future, and these are not even malicious attacks. There are many more examples of these; for example, Ye et al. [5] have shown how

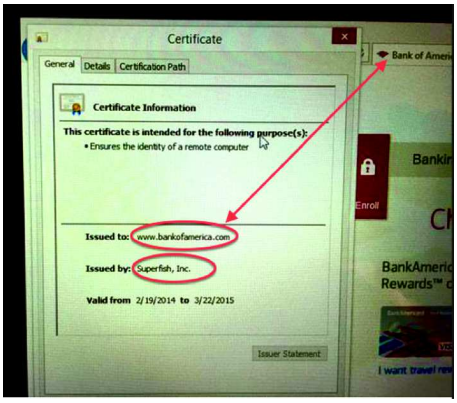

FIGURE 1: Bank of America's certificate signed by Superfish instead of Verisign.

malicious web sites can trick users into believing they have a secure SSL session when they do not.

On the other hand, many other stories in the news show that different CAs have either abused the trust that RPs have in them or their systems have been hacked to issue false certificates. For example, in June 2011, DigiNotar, a Dutch CA, was hacked. The hackers made DigiNotar sign hundreds of falsified certificates for high profile websites such as Google and Facebook. One year after, DigiNotar declared bankruptcy. Other stories showed how CAs have abused the trust of RPs. For example, on 23 March, 2015, Google discovered that China Internet Network Information Center (CNNIC) issued an unconstrained intermediate certificate to an Egyptian company that used this certificate to intercept communications of web users accessing Google domains (i.e., a TLS MITM attack).

We believe that the main problem of the web TLS system comes from the fact that web users must trust a multitude of entities in order to secure their transactions. First of all, they must trust their web browsers to validate web sites' certificates on behalf of them. Trusting the certificate validators is not limited to known web browsers because anyone has the right to validate certificates on behalf of web users or trick users into believing validation has occurred. Secondly, the same web user has to trust directly hundreds of unknown CAs provided by different OS/browser editors, because the latter does not want to assume any responsibilities if something goes wrong with any CA. In order to justify these bold statements, we first need to list the obligations [6] of web users before they should accept a public key certificate:

(1) Users should ensure the authenticity of the trust anchor or "root" CA (i.e., ensure that the public key of the CA belongs to the claimed CA).

(2) Users should trust the trust anchor CA to issue certificates.

(3) Users should know that the subject's certificate is appropriate to the context of use.

(4) Users should ensure that the subject's certificate is valid, as well as all the certificates in the chain up to the trust anchor's certificate or public key (i.e., conform to the right standards).

To realize task 1, web users must get the certificate of a "root" CA from a trusted source or by some out of band means. According to RFC 5280, web users are supposed to build their trust decisions (task 2) by analysing a set of CA documents (Certificate Policy (CP) and Certification Practice Statement (CPS)) to answer many technical and legal questions like what happens when the CA does not correctly check the identity of the certificate holder, or worse, when it issues a certificate to a person with a false identity? What happens if the certificate is false and makes me lose $1000? Is the CA responsible? [7]. Executing the validation obligation (task 4) is impossible for human users. Consequently, except for task 3, no user is able to realize these tasks and must be aided by trustworthy software with trustworthy configuration data. It should be noted that all these tasks must be executed when in fact most of users do not have any knowledge about what certification authorities are and, furthermore, when they are in the middle of performing some much more important application task (such as making a purchase on the Internet). This was demonstrated through several experimental studies [8–11].

Thus, web users must depend on other entities to help them achieve these obligations. We use the term recommender for those entities who provide the software and configuration data. Web browsers are one of the best known examples that users may use. Three categories of recommenders can be distinguished; the first category proposes only to realize the validation obligation (task 4) and partially task 3 by checking the key usage field, such as Chrome, Opera, and IE. The second category realizes tasks 1 and 2, such as Microsoft and Apple by distributing trust CA lists in their OS. The third category implements tasks 1, 2, 3 (partially), and 4 on behalf of users, such as Firefox.

While the aforementioned examples of recommenders are known entities, no countermeasure exists that may limit the dependence of web users on other unknown entities. For example, any unknown mobile application developer may also realize the aforementioned obligations on behalf of smartphone web users (how many unknown web clients exist on AppStore and GooglePlay stores?). Additionally, many mobile applications integrate embedded browsers into their primary services, like the Facebook application. All these kinds of recommenders may expose (intentionally or not) web users to MITM attacks. Finally, any computer manufacturer may also manipulate the list of CAs (realizing tasks 1 and 2) before shipping the computers to their clients (e.g., Dell).

Trusting different lists of CAs (trust list) provided by different OS/browser editors can make the web users confused. Indeed for the same website, a web user may get different responses depending on the application (IE, FF, and Chrome)/platform (Windows, Linux, and Android) adopted by the user to access the website. On one hand the list of CAs is different from one application/platform to another. On the other hand, the quality of the validation process depends on the understanding of the application/platform developer to

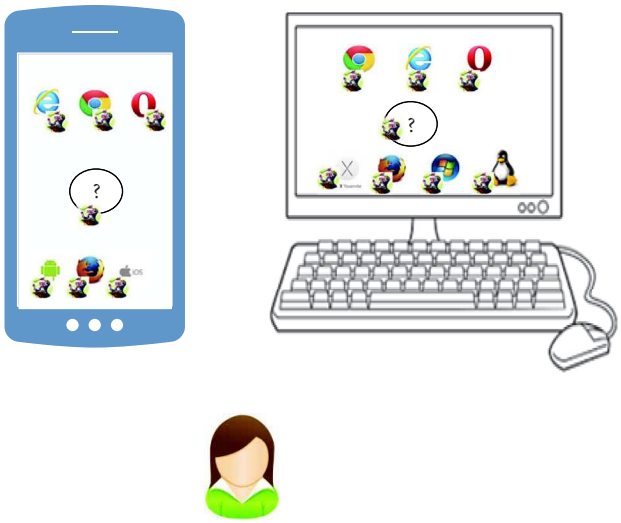

Figure 2: Web users dependence on almost anyone for validating certificates.

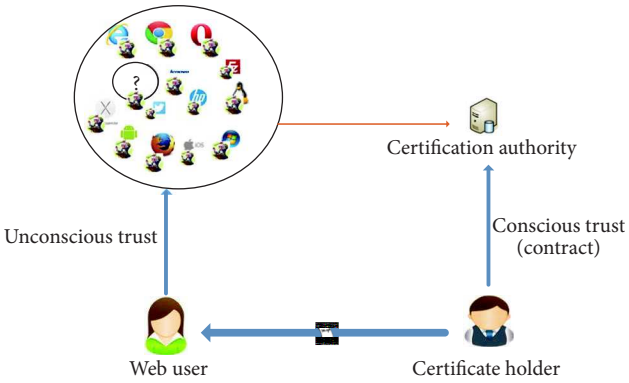

Figure 3: Current trust model for web users.

the related standards, even when the latter may not be clear about different points of validation [12].

From the trust point of view, the relation between the web user and the recommenders is constructed on an unconscious basis. Indeed, users are primarily concerned with their task in hand (Internet surfing, social network, FTP client, buying a computer, etc.) and they believe that no harm will come to them when accepting the services of their recommenders. We call this kind of relation *unconscious trust* (see Figures 2 and 3). This is in direct contrast to the relationship between the certificate subject and their issuing CA. In this case the subject has made a *conscious* decision to trust a particular CA and has cemented this trust by paying the CA a fee for their certificate.

In real life, all of us make unconscious trust decisions to handle the complexity of our world [13]. For example, we cross the streets without caring about car drivers and we go to the street without carrying firearms. In this case, our unconscious trust is justified by the rarity of bad events. However, this unconscious trust transforms to conscious trust only when the frequency of bad events increases. This supposes that humans are able to detect the dangers and bad events. However, on the Internet users are unable to detect these problems; they depend on experts to detect them and inform them. Clearly, web users on the Internet will continue to depend unconsciously on the services of unknown entities. The major risk of this kind of relation is that the unconscious trust in recommenders is usually transformed into unconditional trust that gives the recommenders complete discretionary power over the web users.

Thus, the repeated attacks happening every day come from the fact that web users trust almost everyone in the world to validate the X.509 certificates they receive. This fact leads us to ask the question: "*what is the benefit of a PKI if in the end we need to trust almost everyone in the world, in order to be able to use it*?" We believe that the current management of the web TLS system is broken and that PKI with the current management model is not fit for purpose.

Different programmes have been proposed to improve the current web TLS system (e.g., Certificate Transparency [14, 15], Sovereign Keys [16], and Public Key Pinning [17]). While those programmes prove the deficiency of the current web TLS system, they only partially handle the problems of the current TLS system. For example, the Certificate Transparency programme of Google proposes a public online monitoring and auditing system. The objective is to bring transparency to certificate issuing so that a web user can detect in real time any fake certificate. The success of this ambitious programme depends on the participation of all TLS system stakeholders (OS providers, CAs, web browsers, and domain owners). Currently only Google Chrome and CAs that are issuing EV certificates are included in this programme. Ultimately, this will help web users uniquely to realize task 1, but not the other tasks. Thus, this kind of solution increases the dependency of web users on other unknown entities who are partially handling the web users' needs.

It is important to consider the current TLS system as a whole, which is built on the benevolence of all the recommenders between the certificate subject and the web user. We believe that providing end-to-end security between certificate subjects and web users begins by the identification of all the responsibilities of all the recommenders intervening between web users and certificate subjects. The current web TLS system must be improved to remove any source of confusion for web users. The PKI industry soon realized that the PGP approach for distributing public keys would not work effectively or efficiently on the Internet. Having multiple recommenders in PKI is moving nearer to the PGP trust model. Entities that are providing the obligations of web users should not be computer programs provided by any unknown entity in the world.

Originally, X.509 was based on the 3-cornered trust model (see Figure 4): the certification authority (CA), the certificate holder (or subject), and the Relying Party (RP). In a previous paper [18], we have shown that the original X.509 trust model is not sufficient for the Internet. We proposed thus to add a new role of Trust Broker (TB) to the original X.509 trust model (see Figure 5). The TB is independent of CAs and plays the roles of both technical and legal expert

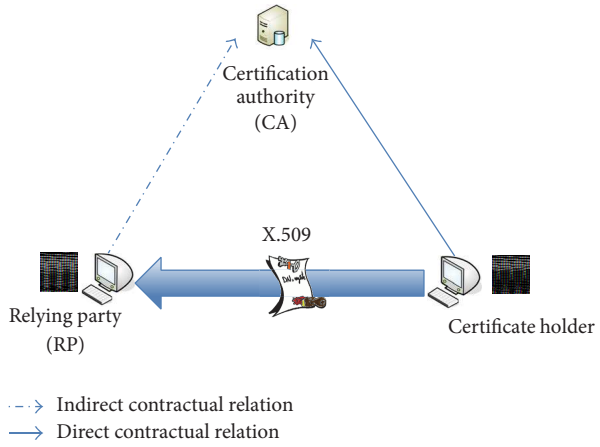

Figure 4: Original X.509 trust model.

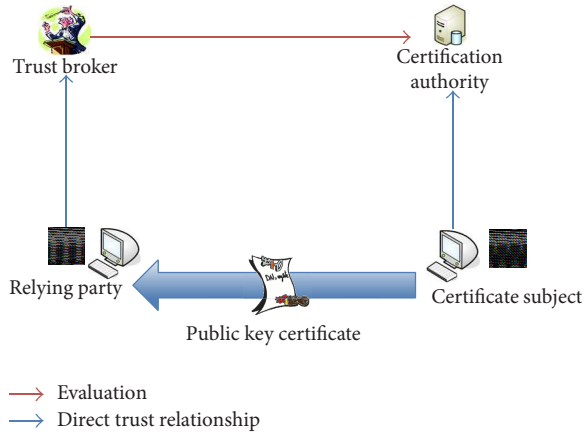

Figure 5: The new X.509 trust model.

for helping the RPs (web users). By explicitly adding this role to the original X.509 trust model, the task of RPs is simplified, and the responsibility of the entity acting as a Trust Broker can be formally engaged. From the user's point of view, whatever number of platforms and applications they may use to access a website, they will always get the same recommendation from their contracted TB to help them make an informed decision.

It should be noted that assessing the trustworthiness of a Trust Broker is much simpler than assessing the trustworthiness of all Internet based CAs. Firstly, RPs only need to assess their trust in a single TB who will then help them decide about all certificates from all CAs. Secondly, RPs will have a contractual agreement with their chosen TB, based on local contract law, rather than having to rely on the many different national contracts laws employed by the existing CAs. In essence, choosing a TB will be similar to choosing an insurance policy, which users are already familiar with.

Our contributions to the problem of trust management for PKIs are multiple. In [19], we clearly identified the reasons behind the interoperability problems of PKIs. This has helped us to understand the root causes behind the failure of PKIs on the Internet (open PKI deployment model). In [18], we have proposed to formally extend the original X.509 trust model, by adding the TB entity. This new model is now incorporated in the eighth edition of the X.509 standard [20].

In [21], we proposed to quantify the quality of certificate (QoCER) to allow RPs to make a decision about the certificate. The QoCER score is calculated based on the evaluation of the procedures announced by the CAs and their commitment to apply them. The QoCER value is completed by another parameter, called the quality of control (QoCTRL), which states the degree of confidence on the value QoCER. Although, the couple (QoCER and QoCTRL) can represent the information sent by the TB to the RP, this calculation model suffers from multiple issues.

Issue A. The calculation of CAs' reputations is subject to collusion attack. Some recommenders (in particular unknown RPs and certificate holders) can collude to improve the reputation of CAs or inversely to incriminate well-behaved CAs.

Issue B. This calculation model does not describe precisely the aggregation and the collection approaches of recommendations.

Issue C. The calculation model does not address the scalability issue. This work does not help the TB entity to evaluate a large number of CAs in a reasonable time.

Issue D. Finally, this work was theoretical only. We did not study the issues related to users' decision-making process.

The calculation proposed in [21] faces these issues because the four-cornered trust model was not defined at that time, and the problem of PKI interoperability was not clearly expressed.

In this article, we improve our original calculation model. The maturity that we gained from the PKI interoperability issue and from the 4-cornered trust model has allowed us to come up with a list of requirements that any trust calculation model must conform to. These requirements are independent from any trust calculation methods. In addition, we enhance our calculation model to comply with these requirements, in particular:

(1) This new calculation model addresses the problem of collusion attack (issue A) by defining two groups of recommenders: identifiable recommenders and unknown recommenders. The recommendations sent by identified entities can be automatically accepted but weighted according to the degree that the TB service believes in their recommendations. However, the TB must validate the recommendations provided by unidentified entities before being accepted. This increases the reliability in the calculation. Equations (5) and (8) have been updated in Section 4.2.2 to implement this issue.

(2) We add a preparation stage (Section 4.2.1) to the calculation of CAs' reputation to cope with the second issue (issue B). The TB entity can set the types of recommenders, the collection method (automatic,

manual, etc.), and the types of recommendations (positive or negative) for each trust factor.

(3) The scalability issue (issue C) is handled by proposing a semiautomatic evaluation process (cf. Figure 7). The process is designed to be open to allow TB services to deal with different kinds of CAs' policies and to avoid refusing CAs because of interoperability issues.

(4) Finally, we have implemented this calculation model to demonstrate the feasibility of our proposal in the context of the web (Section 5). This implementation highlighted some drawbacks in the original theoretical model in decision-making process (issue D). We improve this point by (i) sending contextual information to the RPs so that they can make informed decisions about certificates and (ii) returning only one quality value instead of two.

The rest of this document is structured as follows: Section 2 presents the existing trust building approaches that may help RPs to make informed decisions about certificates. We show that none of these approaches can be applied efficiently to help RPs with unknown CAs, so in Section 3 we present our unified approach along with a set of trust evaluation criteria that any effective trust building approach must fulfil. Section 4 presents our trust calculation model and we show how it satisfies the criteria. In Section 5 we present a prototype implementation demonstrating how Internet users (RPs) can make informed decisions about web server certificates. In Section 6, we show how the 4-cornered trust model can improve the security of web users. Finally, in Section 7, we present our conclusions and proposed future work.

2. Existing Approaches to Building Trust

There are several alternative approaches that permit a RP to trust a certificate but all entail two important mechanisms:

(i) A contractual process for recognizing CAs: this is used to prove that a given CA meets the legal and technical requirements of trustworthiness and interoperability.

(ii) A mechanism for conveying the recognition of trustworthy CAs into the RPs computer system: this is used to provide information about the trustworthiness of a CA in a machine-readable format, so that when the RP's software receives a digital certificate it can automatically decide to accept it or not. This is achieved via configuration of at least one root of trust, or trust anchor, into the RP's system by some out of band means. Subsequently certificate chains can be carried in an application level protocol. Providing the chain starts at an already configured root of trust, then the entire set of CAs in the certificate chain can be trusted.

The alternative approaches can be classified into three main categories: (1) trust topologies managed by CAs themselves, (2) a list of roots of trust managed by the RP or by a trusted third party (TTP) that is independent of the CAs and is acting on behalf of the RP, and (3) a hybrid approach in which roots of trust are managed by the RP or a TTP and subordinate CAs are managed by the CAs themselves. One of the main differences between these approaches is their applicability to deployment models of PKI, closed or open. The open deployment model is where all CAs on the Internet are able to be trusted by RPs, whereas the closed deployment model is where only a limited subset of CAs can be trusted.

The implementation of CA managed topologies in the open model is not feasible. One could imagine a topology composed of cross-certified national root CAs in which each root CA manages cross-certification processes with their subordinate CAs located in their jurisdictions. However, even this cannot be easily achieved for several reasons:

(i) Technically, this topology cannot be implemented because of the difficulty of managing long certification paths [22]. The validation process requires several checks to be made along the certification path (e.g., policy constraints, certificate status, and policy mappings). The complexity increases with the size of the certificate chain.

(ii) This topology is similar to a general accreditation system where all CAs must be certified by their national authorities. However, countries do not have the same viewpoint concerning the right organizational model of PKIs. For certain countries, national accreditation may limit innovation and competition between CAs.

(iii) Imagining that the national CAs (root or bridge) can cross-certify each other implies that a technical and legal harmonization can be conceived between different nations. In reality this is too difficult to achieve because of cultural and legal differences between countries.

(iv) This topology requires a standardization of the certification process so that a cross-certification realized by one national CA would be accepted by other national CAs. However, there is no standard cross-certification process today.

Alternatively, trust in a certificate can be recommended by any entity independent of CAs. Users in a given community of interest can obtain information and advice from the leader of this community about the relevance of certificates for their transactions. This recommender should have a technical and legal expertise sufficient to inform its users about the relevance of a certificate for a given type of transaction. The recommender could be a government (e.g., PKI Gatekeeper in Australia [23]) or any organization such as a software vendor (e.g., Microsoft or Mozilla).

In general, the recommenders create a list of minimum requirements and recognize all CAs whose certificates have assurance levels greater than the minimum requirements. Web browsers are the best known examples of this approach (Microsoft Root Certificate Program [24] and Mozilla CA Certificate Policy Inclusion [25]).

In contrast to the previous approach, this approach has only one mechanism used to transmit the recognition of certificates, which is the trust list. There is no homogeneous

way to define or formalize the trust lists. While some lists of certificates are just simple lists (e.g., stores of certificates in web browsers) where RPs can themselves add, edit, or delete certificates, others can be signed lists by the recommender where RPs cannot modify the list. From an interoperability viewpoint, the trust list replaces the cross-certificates used by CA managed topologies. The user trusts the issuer of the list and transitive trust extends this to the CAs contained in the list. As a consequence, the issuer of the list plays the role of trust anchor but is not a CA.

Thanks to the independence of the recommender from CAs and the absence of need to build certification paths for the validation of certificates, the recognition approach is more convenient to the open deployment model of PKIs. However, the current application of this approach is not optimal for the open deployment model, for several reasons:

(i) The nature of the RP's relation with the recommender is not formally defined. It can be formal as in the case of the Gatekeeper strategy [23] or nonformal as in the case of web browsers.

(ii) The cross-recognition process is a manual nonreproducible process; it is performed manually by experts who should examine very large documents that include a lot of political and legal information.

(iii) This approach provides only a binary response, recognized or not. Unrecognized certificates are not banned to RPs since they are constantly exposed to them and a decision must be made. For unrecognized certificates, RPs may still be invited to inspect the policies of CAs to decide whether the certificates are suitable for their transactions or not. The best known example is the web browser, when RPs receive certificates signed by CAs that are not included in the trust list of their browser. The RP is asked to take a decision about the untrusted CA's certificate.

In the hybrid approach, the roots of trust are managed by the RP or a TTP on the RP's behalf, and additional CAs are managed by the root CAs themselves. These additional CAs are termed subordinate CAs (of the root CA) and are fully trusted by the root CA. Consequently certificate chains are received by the RP and certificate path processing is required by the RP's software. The hybrid approach is the one used on the Internet today in the open deployment model.

3. The Unified Approach: A New Approach for Building Trust in X.509 Certificates

Establishing trust in a certificate requires managing technical, organizational, and legal issues. This task is complex, so that only technical and legal experts can perform it. It is not conceivable to delegate this task to unskilled people acting as RPs. To address this challenge, we propose a new approach for managing the trust in certificates, which we call the "unified approach." This can help RPs to take efficient decisions about certificates for both the open and closed PKI deployment models.

Our approach combines the advantages of the current trust topologies. It goes further by realizing new criteria that can increase the efficiency of the RP's trust decision, such as the reliability of recommendations.

In the closed model, the administrators of PKIs and the jurists of organizations play the roles of technical and legal experts to help their respective employees to decide about certificates coming from other organizations. The trust relationship between RPs and their experts is naturally created because they belong to the same organization. The trust of the RPs in their administrators is not only related to the quality of the certificates they provide but also on their ability to recommend the CAs of other organizations. In addition, the decisions of the RPs can be automatically configured because the interconnection topologies are often built for a predefined number of services related to the nature of the collaboration between the organizations.

In the open model, the situation is far more complex for several reasons:

(i) There is no explicit and balanced predefined trust relationship between RPs and experts. For example, web browser editors play implicitly this role as they manage a list of trusted CAs, but there is no agreement between the RPs and the editors to hold the editors responsible for the information they provide.

(ii) The scope of the certificate's usage is open (i.e., not limited to predefined specific services). The consequence is that web browsers do not provide enough information to make an informed decision. The recommendation is binary (trusted or not recognized, e.g., an icon in the URL bar is blue or not). All trusted CAs are stored in the same trusted list; therefore CAs with different trust levels are equally trusted regardless of the usage of the certificate.

All these ad hoc solutions, either for the open (e.g., web browser approach) or for the closed model implicitly, include the role of expert. The differences lie in the nature of the entities playing the role of expert, in the type of trust linking the expert with the RPs, and in the nature of the information that the expert supplies to RPs. The role of the expert has been added to the latest X.509 trust model, as shown in Figure 5. This explicitly separates the role of certificates' manager from the role of expert. Thus, the new trust model for X.509 PKIs is composed of four entities: the Trust Broker, RPs, certificate holders, and CAs. Each entity in this approach has a specific task/responsibility as follows:

(i) CAs are responsible for managing certificate lifecycles.

(ii) Certificate holders must responsibly use the certificates given to them by the CAs.

(iii) RPs must take decisions whether to accept certificates or not.

(iv) TBs are responsible for evaluating CAs on behalf of RPs (analysis of CP/CPS, auditors, etc.).

The TB evaluates the CAs and sends recommendations to RPs for helping them to take informed decisions about

certificates. In this case, the trust model becomes fairer to RPs because they are protected by one entity, that is, their technical and legal expert. According to this model, RPs rely only on the recommendations of their technical and legal expert and not on each and every CA presented by the certificate holders. The relation between the TB and the RP must be regularized by an explicit contractual agreement. In such an agreement, the TB recognizes its responsibility to the RP about its provided recommendations and respects the privacy of the RP. The TB must be independent from the CAs. However, its relationship with CAs may be regularized by explicit agreements, so that the TBs can transfer the responsibility to a CA when a false recommendation is given due to incorrect information provided by the CA. The contractual agreements between the RPs and the TBs can be achieved in several ways:

(i) By commercial services, similar to insurance services, whose business model is to sell recommendations about certificates

(ii) By national organizations whose role is to protect consumers

One of the main advantages of the TB approach is that it resolves the interoperability problem of PKIs by transforming it into a trust management problem. The persistence of interoperability problems creates a trust management problem; if there was a compatibility between PKIs at the juridical, organizational, and technical levels, there would not be a trust management problem because there would be a limited number of classes of globally accepted certificates, where each class met a specific context of use. However, the cultural juridical and technical differences between countries are profound. Thus this theoretical solution cannot be implemented in practice. The TB approach does not remove the interoperability obstacles, but rather it admits their existence and tries to inform RPs about the risks resulting from the interoperability problems. This approach accepts interoperability problems because it handles all certificates regardless of the technical and legal rules applied when generating the certificates. In the following sections, we first give a definition of trust in a CA and then define a list of criteria that the unified approach must meet. Finally we present our underlying trust calculation model.

3.1. Trust in Certification Authorities. The phrase "trust in a CA" has been used without explaining what it means exactly from the perspective of RPs. It is important to define this concept before presenting the calculation model. One should differentiate between the terms "trust in a CA" and "trust in a PKI." Thus, "trust in a PKI" implies trust in all the CAs that a PKI contains. However, for the RP that is executing a transaction, it is only important to evaluate the trust it can have in the CA that has signed the certificate used in the transaction and in every CA between this CA and the root of trust.

Although trust seems intuitive to humans, there is no consensus on one single definition. The concept of trust suffers from an imperfect understanding, a plethora of definitions, and informal use in the literature as well as in everyday life [26]. This is compounded on the Internet, where different meanings and terminologies can be identified by language and/or culture. For example, the English language provides two words to express two dimensions of trust: "trust" and "confidence," while the French language knows only one word "confiance." The English language also provides concise and accurate terms to refer to the partners in a trusting relationship, namely, trustor and trustee, whereas the French language lacks these nouns.

The differences between the definitions of trust are also found depending on the discipline of the authors. Psychology [27], sociology [28, 29], and philosophy [30, 31] are all disciplines that have devoted efforts to the study of trust. However, by inspecting these definitions, we find that they are generic and applicable to many areas. They may implicitly include many aspects. It is therefore necessary to specify the definition of trust that explicitly details all the important aspects of trust in a given area.

In our view, trust in a CA from the perspective of a RP must be established in terms of the security and reliability of the CA's services. This depends upon both human and computer systems. However, the characteristics on which we rely to trust technological systems are different from those to trust humans. Jøsang [32] explains this difference in the field of information security as follows:

(i) The security that emerges from a human being is benevolence to that person, while the security of a system is the ability to resist attacks. The benevolence of a person means that he/she is honest and straight. (S)he is honest if (s)he respects their words and straight if (s)he respects the rules.

(ii) The reliability of a person is represented by his/her qualities such as experiences and skills, while the reliability of a system is its ability to continually perform a specific task.

Thus, the security and reliability of a CA's services are dependent upon the security and the reliability of all the entities involved in the certification process, both human and technological. As a consequence, we define the trust in a CA as "Dependence on the ability of people, systems, physical locations, and software of a CA, as well as on the benevolence of the CA provider to provide the required security services while complying with the relevant legislations."

In this definition, we consider people are the individuals working in both the CA's and PKI provider's organizations. Their security characterizes their commitments to the security policies (CP/CPS) and the relevant legislations, while the reliability represents their skills and experiences in the field of PKI. The security of systems and software is the ability of these entities to resist attacks, while their reliability means that they are capable of performing tasks continually and without errors.

The given definition demonstrates the expectations of RPs towards CAs. The expectations of certificate holders towards their CAs are defined through contracts. Similarly,

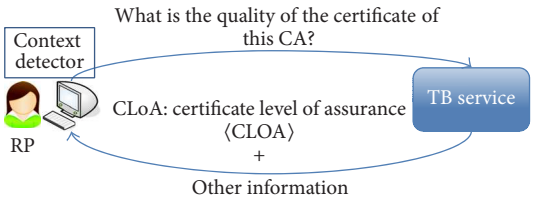

Figure 6: The TB service.

the relationship between RPs and their TBs will be regularized through contracts.

3.2. Trust Evaluation Criteria. Trust evaluation in both the closed and the open models of PKIs should realize the following six criteria (note that we do not specify general software engineering criteria such as efficiency and ease of use):

Criterion 1 (the evaluation process should be carried out by an expert on behalf of the RP). The recognition of a CA at the technical and legal level is too complex for most users; therefore it should be made by an expert working on behalf of the RPs.

Criterion 2 (recommendation retrieval should be simple and dynamic). The process of retrieving trust recommendations from an expert should be as simple as possible for RPs and should be dynamic to cater for changing situations.

Criterion 3 (certificate evaluation should be global in scope). The approach should be able to analyse all certificates that RPs may receive, regardless of the technical, legal, or geographic position of the issuing CA.

Criterion 4 (recommendations should be relevant to the context of use). Trust recommendations must be as relevant as possible to the context of use (e.g., authentication of FTP server, bank server, or merchant server for a payment transaction). This allows RPs to take the most effective decision without applying considerable mental effort.

Criterion 5 (the privacy of the RP should be respected). The expert should not learn anything about the transaction the RP wishes to undertake.

Criterion 6 (the reliability of the recommendations). The trust evaluation must consider the reliability of the trust recommendations.

4. Trust Calculation Model

To help RPs decide about the trustworthiness of subject certificates, a set of quantitative and qualitative information is sent to them. The Trust Broker (TB), as proposed in the new X.509 trust model, fulfils the first criterion and can set up a service that provides this information to RPs (see Figure 6). The retrieval of recommendations can therefore be made simple and dynamic (Criterion 2). Furthermore, there is no need to handle long certificate validation paths as is the case for CA managed topologies. In the TB model, the TB is the root of trust for all CAs. Consequently, the RP only needs to send the CA's certificate to the TB service. The subject's certificate is not needed, since the CA applies the same procedure to all its issued certificates. Furthermore, this satisfies the 5th criterion or privacy, since the TB does not know which certificate holder the RP is communicating with.

The TB service returns other information that can help the RP to make an informed decision. For example, when an RP needs to know the liability of the CA in case of problem. The determination of any liability information is obtained from the CP of the CA by the TB service and relayed as other information to the RP. In addition, we have added a context detector at the side of the RP in order to detect the actual application context. By doing this, the TB service realizes the 4th criterion.

At a purely quantitative level, the TB service sends a score between 0 and 1 that represents the trustworthiness of the subjects' certificates in general, called the certificate level of assurance (CLoA). This satisfies the 3rd criterion. When the CLoA is 0, the CA's procedures for managing the subjects' certificates are judged by the TB to be very weak or nonexistent. When the CLoA is 1, the applied procedures are judged to be very strong and faultless. The calculation of CLoA depends on multiple factors, namely, the CA's published procedures (QoCPS), the CA's actual procedures (QoCA), and the confidence the TB has in the CA to adhere to its procedures (CL). This satisfies Criterion 6.

To calculate the certificate level of assurance (CLoA), we propose the following formula:

$$\text{CLoA} = \sqrt[n]{\text{CL} * \text{QoCA}} * \text{QoCPS}, \tag{1}$$

where (i) $\text{QoCPS} \in [0, 1]$ represents the robustness of the CA's published procedures in its CP/CPS documents. The value 0 represents the weakest procedures and 1 the strongest procedures. (ii) $\text{QoCA} \in [0, 1]$ represents the degree of the CA's commitment to its published procedures. It is based on recommendations provided by third parties that monitor the real practices of a CA such as audit agencies and the RPs themselves. The value 0 represents that there is no evidence to indicate that any statements in the CP/CPS have been respected, while 1 indicates that every statement in the CP/CPS has been implemented according to the recommenders. (iii) $\text{CL} \in [0, 1]$ represents the degree of confidence that the TB has in its calculation of QoCA (we assume the TB always has 100% confidence in its calculation of QoCPS). The value 0 means either there is no evidence on which to calculate QoCA or the TB has zero confidence in the evidence that is there, while the value 1 indicates that there is adequate evidence for the TB to validate every statement in the CP/CPS. CL can be 1 when QoCA is zero, meaning that the TB is certain that QoCA is low. (iv) n is an integer value that allows the TB to control the impact of $\text{CL} * \text{QoCA}$ on the score of CLoA.

The maximum value of CLoA is QoCPS because QoCPS represents the published robustness of the CAs procedures for managing certificates. However, this maximum value can

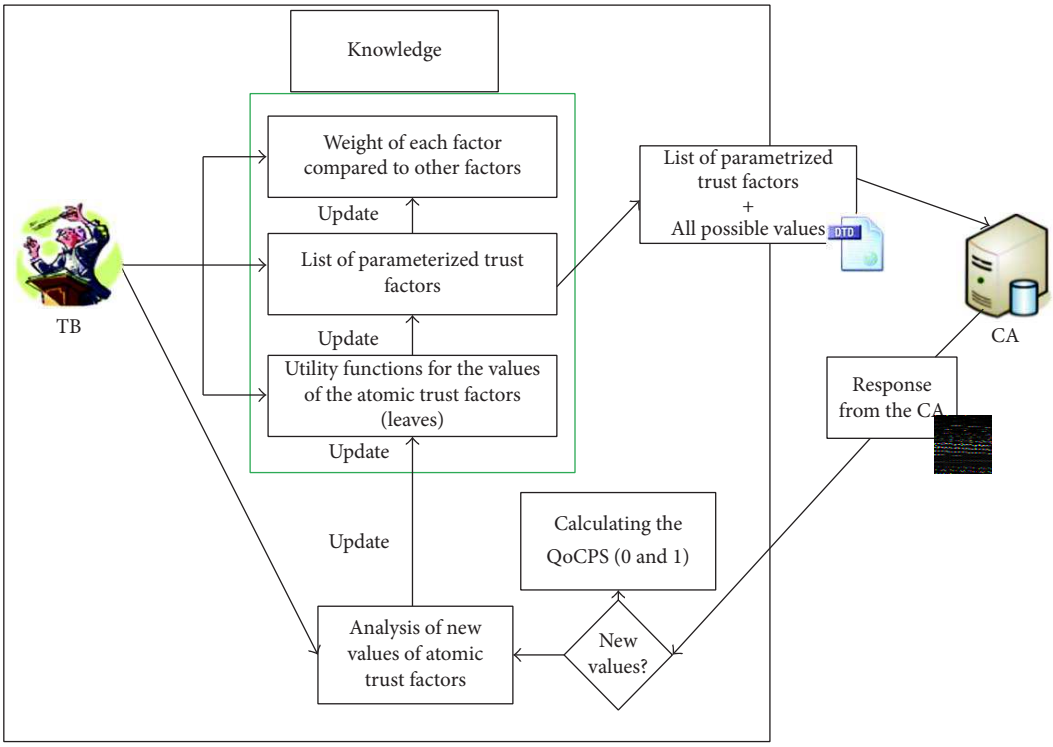

Figure 7: The semiautomatic process for computing QoCPS.

be degraded because either the CA does not fully respect its own procedures, or the TB does not have full confidence in either the CAs stated procedures or the recommendations about them. This means the value of both CL and QoCA can decrease the value of QoCPS to reflect the TB's assessment of the overall CLoA.

4.1. Computing the Quality of the CPS (QoCPS). We introduce a semiautomatic process used to determine the quality of the CP/CPS documents, as shown in Figure 7. We propose a technique for structuring CP/CPS documents so that they can be understood by computers, and then an algorithm will be used to determine the quality level of the CP/CPS documents (QoCPS).

4.1.1. CP/CPS Structuring. The natural language used for describing the Certificate Policy and practices of a CA is one of the main obstacles in determining the trustworthiness of a CA. In order to automatically interpret the CP/CPS documents, we model the CP/CPS documents as a tree structure (as illustrated in Figure 8) inspired by the de facto standard RFC 3647, which defines a common framework for CP/CPS documents. The structure is composed of nodes and leaves, where leaves are atomic trust factors and nodes are complex trust factors (i.e., a combination of atomic and complex factors).

For example, "Technical Security Controls" is a node composed of the following nodes, where ≪ ≫ represents a node and < > represents a leaf:

(i) ≪Key Pair Generation and Installation≫
(ii) ≪Private Key Protection and Cryptographic Module Engineering Controls≫
(iii) ≪Other Aspects of Key Pair Management≫
(iv) ≪Activation Data≫
(v) ≪Computer Security Controls≫
(vi) ≪Life Cycle Security Controls≫
(vii) ≪Network Security Controls≫
(viii) ≪Time Stamping≫

The node ≪Key Pair Generation and Installation≫ is composed of the following nodes:

(i) ≪Key Pair Generation≫
(ii) ≪Private Key Delivery to Subscriber≫
(iii) ≪Public Key Delivery to Certificate Issuer≫
(iv) ≪CA Public Key Delivery to Relying Parties≫
(v) ≪Key Sizes≫
(vi) ≪Public Key Parameters Generation and Quality Checking≫
(vii) ≪Key Usage Purposes≫

The node ≪Key Sizes≫ may have the following trust factors (leaves):

(i) <6.1.5.f1[X, Y]> represents size X of the public key of the CA certificate for algorithm Y, for example, [1024, ElGamal].

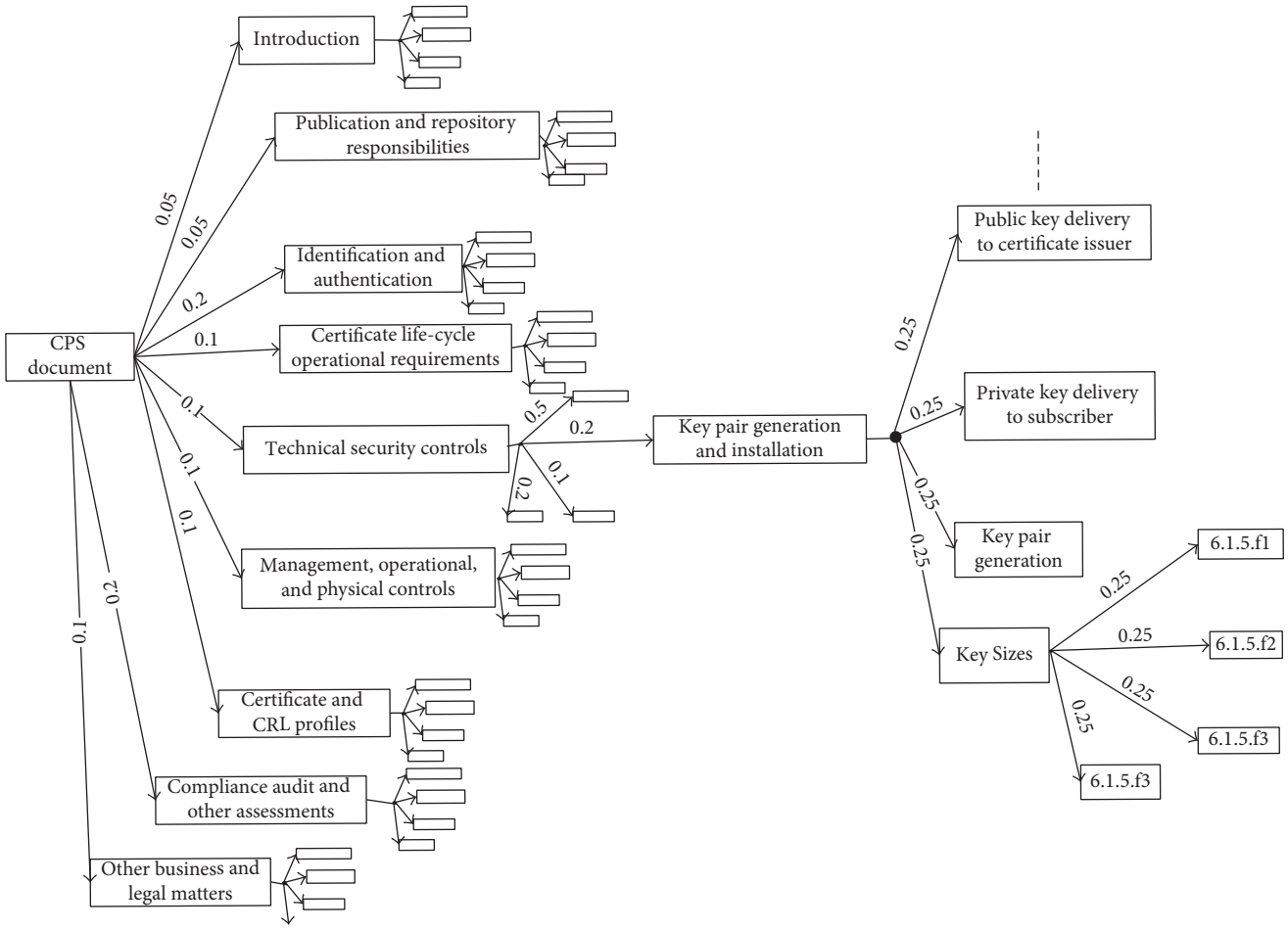

Figure 8: Weighted CP/CPS tree structure.

(ii) <6.1.5.f2[X]> represents the hash algorithm used by the certificate of CA, where $X \in$ {SHA-1, SHA-224, MD5}.

(iii) <6.1.5.f3[X, Y]> represents the public key size X of certificate user for algorithm Y.

(iv) <6.1.5.f4[X]> represents the hash algorithm used by the certificate user, where $X \in$ {SHA-1, SHA-224, MD5}.

The node ≪Key Pair Generation≫ may have the following leaves:

(i) <6.1.1.f1[X]> represents cryptographic modules used by CAs for the generation of keys according to the requirements of standard X, where $X \in$ {FIPS 140-1 level 1, FIPS 140-1 level 2, FIPS 140-1 level 3}.

(ii) <6.1.1.f2[X]> represents cryptographic modules used by users for generation of keys according to the requirements of standard X, where $X \in$ {FIPS 140-1 level 1, FIPS 140-1 level 2, FIPS 140-1 level 3}.

(iii) <6.1.1.f3[X]>: generating the key pairs of the end user is performed by the user himself, where $X \in$ {Oui, Non}.

(iv) <6.1.1.f4[X]>: generating of the key pairs of the end user is performed by the CA or RA, where $X \in$ {Oui, Non}.

In each leaf, the TB service defines a set of possible values that are semantically known by the TB and the CA. The types of values can be simple answers (yes/no), numerical values, dates or names of standards, and so forth. The trust factors must be independent of each other; if not, they must form a complex node. We give each trust factor a reference number that corresponds to the section number of RFC 3647.

The obtained structured file constitutes the knowledge of the TB at time t. This knowledge can be represented in an XML file. The knowledge about a particular CA can be completed in one of two ways. Either the TB can download the CA's CP/CPS, read it, and complete the XML file himself with the CA's published values, or the TB can send the XML file to the administrator of the CA and ask the latter to complete it. If the answerer finds the XML file is sufficient to represent a CA's CP/CPS, then the quality of the CPS can be automatically calculated (see next section). Otherwise the answerer should indicate to the TB that new values from the CP/CPS are missing in the XML knowledge file, so that the TB can analyse them and extend his knowledge. For example, if

Table 1: Trust Utility function for the atomic trust factor 6.1.5.f1 of Key Sizes.

Utility	Algorithm
0	(256, RSA)
0	(512, RSA)
0,25	(1024, RSA)
0,8	(2048, RSA)
0,9	(2048, ElGamal)
1	(4096, RSA)

one CA uses a hash algorithm XYZ that is not present in the knowledge file, then this should be indicated to the TB so that it can be considered for the next version of the file.

4.1.2. The Calculation of the Quality of CP/CPS Documents. To determine the QoCPS value, the TB must first define the relative importance of each factor with regard to the other factors (Figure 8) that comprise a complex node. The relative importance of all arcs leading to a node must add up to 1.0.

For each atomic trust factor, the TB uses a utility function to define the relative importance of each possible value with regard to the other values. For example, Table 1 represents the utility function for the atomic trust factor <6.1.5.f1[X, Y]> of node ≪Key Sizes≫. It gives the value 0.25 if the key size is 1024 bits, and the algorithm is RSA. It gives the value 0.9 when the key size is 2048 bits and the algorithm is ElGamal. The QoCPS value is the recursive weighted sum of all utility functions for all the atomic trust factors:

$$\mathrm{QoCPS}(\varphi_i) = \sum_{j \in \mathrm{children}(i)} \mu_{\varphi_i \varphi_j} * \mathrm{QoCPS}(\varphi_j), \quad (2)$$

where (i) φ_i is a node and φ_j is a node or a leaf and (ii) $\mathrm{QoCPS}(\varphi_j) = U(V_{\varphi_j})$ where φ_j is atomic trust factor. U is the utility function. The utility function allows TBs to define different strategies for trust calculations and forms part of their intellectual property. V_{φ_j} represents selected value for the atomic trust factor φ_j; (iii) $\mu_{\varphi_i \varphi_j}$ is the weight between the factors φ_j and φ_i in the CP/CPS tree.

Clearly, the values of atomic trust factors will evolve over time. These values may be added, changed, or even deleted. Similarly, the utility function must be able to evolve over time. For example, the strength of a cryptographic algorithm today will not be the same after several years. The values given to the various atomic trust factors and utility functions are determined by the TB according to his expertise. This is his intellectual property and will determine in part the value of his TB service in the market place.

Finally, in order to enable the TB to handle all CAs regardless of their technical and juridical level, the list of input values for the atomic trust factors must be published by the CAs in their CP/CPSs. For example supposing that the TB service has the following values {SHA-1, SHA-224, MD5} for the hash algorithm trust factor. If a CA uses an algorithm not included in the list (such as SHA-512), the TB service analyses this value, and it updates the corresponding utility function and references it in the revised list of values.

4.2. Computing the Quality of CA (QoCA). The objective of the quality of CA (QoCA) is to show the degree of commitment of the CA to its CP/CPS documents. Naturally, the audit agencies are the main entities that can provide information about the real commitments of CAs. In fact, the role of the audit agency is very important because a lot of practices can be only understood and reviewed by it. By virtue of the audit agency, the authentication of the evaluated CA is guaranteed and its claims in the CPS file are ensured. For example, the audit agency is the only entity that is able to verify the claim of the CA when it states that its private key is generated and stored in a physically secured environment.

However, the audit agency is not sufficient to provide a reliable guarantee of total conformity for different reasons [33]:

(i) *Time*. The verification made by an auditing agency is neither continuous nor permanent. An audit agency evaluates a CA every year or two. This means that the evaluation conducted a year ago may not reflect the current state of the CA, for example, how to ensure that the list of revoked certificates is available 24/7.

(ii) *Number of Certificates*. An audit agency cannot verify all certificates issued by a CA, for example, how to verify that a CA has really respected the announced certificate profile in its CP/CPS for each one of thousands of certificates and how to verify that none of these certificates are free of errors (e.g., DSA certificates with 2048-bit primes or RSA certificates with a public exponent equal to 1).

(iii) *The Independence of Audit Agencies*. It is difficult to guarantee the independence of audit agencies from CAs, especially because audit agencies are paid by the CAs to perform the evaluation. We recognize that auditors may need the permissions of the CAs to release their results to the TB services. Governments must ensure that CAs give this right to auditors or at least that a summary of their results is made available to TB services.

In addition, we propose new entities that can help in verifying the real commitments of a CA to its CP/CPS documents:

(i) *Clients of TB Service*. An RP depends on the TB service for providing her with the necessary information to take an informed decision. (S)he can also play the role of a recommender to the TB service by providing it with information about the correctness of some of the parameters announced by a CA during certificate validation, for example, availability of CRLs. The RP may also be a certificate holder so that it can send the TB service some recommendations about the commitments of its CA. TB clients can only supplement the assessment of the audit agencies because many parameters of the CP/CPS such as physical controls (e.g., fire prevention and protection of premises), procedural controls (e.g., procedures to ensure segregation of duties), or personal checks (e.g., qualification and experience) can only be verified by

an audit agency. The recommendations sent by the TB's clients may need to be verified by the TB service to ensure their veracity. Among the parameters that can be recommended by TB clients are the following:

(a) The availability of a CA's 24/7 revocation service: this can be analysed by the RP when validating a certificate. When the revocation service is not available the RP can automatically send (negative) recommendations to the TB service.

(b) The certificate profile can be verified automatically by an RP, to say whether it conforms to the declarations in the CA's CP/CPS documents or not (e.g., key usage extension). The TB provides the RP with this when it sends the CLoA.

(ii) *Competitor TB Services*. TB services can share their experiences with other TB services who they know or compete with, in order to help determine the real commitment of a CA. This is similar to insurance companies sharing information today. In addition, cooperation between TB services can facilitate certain actions, such as the confirmation of the actual existence of a CA or an audit agency. The relationships between TB services could be regulated through bilateral agreements or trade associations. Bilateral agreements can be easily constructed, because all TB services have the same motivation, which is increasing the scope and efficiency of their evaluations. Each TB service can control the impact of other TB services on its final result (CLoA, CL, etc.) according to the trust it has in the competitor TB service. Competitor TB services may be reluctant to cooperate with other TB services, especially when they are competing in the same market, but trade associations help competitors to collaborate. TB services that are dominant in different markets may cooperate for mutual benefit. For example, a French TB service may cooperate with a Japanese TB service so that they can exchange useful information about CAs' certificates used by their clients.

Thus, we have a participative system that can be used to compute the QoCA. We propose to apply these trust and reputation management approaches in order to compute this value. We have selected the REGRET model [34], which is a modular approach for managing trust and reputation. It takes into account three dimensions:

(i) *The Personal Dimension*. It refers to direct interaction between entities. When entity A gives certain promises to entity B, then entity B scores entity A according to its real commitment to its promises. For example, seller A sets a date for the delivery of a product to customer B. If the product arrived after that date, then entity B negatively scores seller A based on the negative impact of the delay. If the product arrived on time, entity B positively scores seller A because it has respected its promises.

(ii) *The Social Dimension*. With the social dimension, the REGRET model adds the ability to reflect the characteristics of complex social relationships using the group concept. In many societies, a person inherits the reputation of the group to which it belongs. When direct experiences with an entity are missing, the reputation of its group gives initial expectations about the behaviour of the entity. In the same way, an entity may use the experiences of the members of its own group, or the group of the unknown entity, to complete its expectations about the unknown entity. However, we do not consider the group that the CA is a member of when calculating the reputation of that CA. In our case, the social dimension is calculated based on the recommendations sent by clients of the TB and other TBs it has a relationship with.

(iii) *The Ontological Dimension*. The REGRET model assumes that the reputation of a person is not a single and abstract concept, but rather a multifaceted concept. For example, the reputation of an airline is based on the reputation of its aircraft, its baggage handling, its check-in procedures, and its on board catering. In turn, the reputation of an aircraft summarizes the reputation of the maintenance service, the manufacturer, the engine, and other characteristics. These types of reputation and the way they are combined is the ontological dimension of reputation. Note that each person could have a different ontological structure to combine reputations and a different way to moderate their importance. In our case the ontological dimension of trust in a CA is the amalgamation of many different atomic trust factors arising from the CP/CPS documentation.

In order to calculate the QoCA using this model, the nature of the recommendation values should be changed. Instead of the recommenders providing their personal evaluations to the TB, they should provide the actual values of the CA parameters; then the evaluation of these parameters can be made by the TB service. For example, when a CA states in its certification policy that the download time of a CRL list should not exceed 30 ms, the recommenders value should be the actual time, for example, 40 ms, and not its recommendation score, for example, a number between 0 and 1. When calculating the QoCA, the following stages are proposed for the TB service.

4.2.1. Preparation Stage. The TB service should consider a number of important points for each trust factor:

(i) Types of recommenders: the TB must determine for each trust factor which recommenders can validate it. The list of recommenders depends on the nature of the trust factor and some can be validated only by audit agencies.

(ii) The collection method of recommendations: this may differ according to the nature of the trust factors and recommenders. Audit agencies could send periodic reports to the TB service containing all the real values

found during the audit (subject of course to the agreement of the CA). The RPs, the certificate holders, and other TB services could send their recommendations either spontaneously, periodically, or on request. For example, for the trust factor "the availability of the 24/7 revocation service," the RPs could periodically notify the TB service about this factor, while for the trust factor "in case CA is compromised, the CA should spontaneously notify all subscribers and RPs about the compromise" the collection could be spontaneous. On request collection can be achieved through various techniques including the use of questionnaires, web forms, and e-mail messages.

(iii) Types of recommendations: the TB must determine whether the recommenders should send positive recommendations (when confirming promises) or negative ones (in case of nonfulfilment of promises).

4.2.2. Calculation Stage. To calculate the value of QoCA, the TB service should take into account the personal, social, and ontological dimensions. Two types of recommenders should be considered in the calculation: identifiable recommenders, which are audit agencies, competitor TB services, certificate holders, and client RPs, and unknown recommenders, which are RPs who are not clients and whose identities are not known by the TB service.

The identification of the recommenders helps in implementing anticheating mechanisms that neutralize suspect recommendations. The recommendations sent by identified entities can be automatically accepted but weighted according to the degree that the TB service believes in their recommendations. The weight factor indicates the impact that the recommender can have on the final score. Determining this weight factor is part of the intellectual property of the TB and is one of the factors in distinguishing between TBs. Recommendations provided by unidentified entities must be validated by the TB before being accepted. If the cost of the validation of a specific trust factor is low, for example, checking that an OCSP server is available or not, then both negative and positive recommendations from unidentified RPs can be accepted; otherwise the TB service should only validate negative recommendations as these show that the CA is failing in some respect.

Each recommender sends a recommendation r that has the following form:

$$r = (s, \mathrm{ca}, \varphi, v, t), \tag{3}$$

where the entity s (sender) indicates the set of actual values v of the atomic trust factors φ when it had an experience with a certification authority ca at time t.

Let R be the set of all possible recommendations. We define $R_{\varphi_j}^{s\to\mathrm{ca}} \subseteq R$ as the set of the recommendations sent by s about ca for the atomic trust factor φ_j:

$$R_{\varphi_j}^{s\to\mathrm{ca}} = \{r = (s_r, \mathrm{ca}_r, \varphi_r, v_r, t_r) \in R \mid s_r = s,\ \mathrm{ca}_r = \mathrm{ca},\ \varphi_r = \varphi_j\} \tag{4}$$

The QoCA for the personal dimension of trust factor φ_j for the certification authority ca can be calculated by the TB service (ε) as follows:

$$\mathrm{QoCA}^{\mathrm{Personal}}\left(R_{\varphi_j}^{\varepsilon\to\mathrm{ca}}\right) = \frac{\sum_{i\in R_{\varphi_j}^{\varepsilon\to\mathrm{ca}}} \rho(t_c, t_i) * W_{\varphi_j}^{i}}{n}, \tag{5}$$

where (i) $R_{\varphi_j}^{\varepsilon\to\mathrm{ca}}$ represents all the personal evaluations of the TB service (ε) about the ca for the trust factor φ_j. (ii) $W_{\varphi_j}^{i}$ is the difference between the value promised in the CPS and the actual value calculated by the TB service for the recommendation i. (iii) $\rho(t_c, t_i)$ is a time-dependent function that gives more importance to the most recent recommendations, where t_c is the current time and t_i is the time when recommendation i has been stored. We do not fix this function, because this function can be defined in several ways depending on the type of trust factor. This is part of the intellectual property of the TB. (iv) n is the number of evaluations stored in $R_{\varphi_j}^{\varepsilon\to\mathrm{ca}}$.

$W_{\varphi_j}^{i}$ can be calculated using this function:

$$W_{\varphi_j} = \begin{cases} \dfrac{U\left(V_{\varphi_j}\right) * 100}{U\left(V_{C_{\varphi_j}}\right)} & U\left(V_{\varphi_j}\right) < U\left(V_{C_{\varphi_j}}\right) \\ 1 & \text{otherwise,} \end{cases} \tag{6}$$

where (i) V_{φ_j} is the measured value of trust factor φ_j, (ii) $V_{C_{\varphi_j}}$ is the promised value by the CA in its CPS for the trust factor φ_j, and (iii) U is the utility function.

$\mathrm{QoCA}^{\mathrm{Soc}}(\varphi_j)$ is the QoCA value for the social dimension of each atomic trust factor (φ_j); it can be calculated as follows:

$$\mathrm{QoCA}^{\mathrm{Soc}}\left(\varphi_j\right) = \sum_{s\in S} \xi_s \mathrm{QoCA}^{\mathrm{Personal}}\left(R_{\varphi_j}^{s\to\mathrm{ca}}\right), \tag{7}$$

where (i) S is the global set of identifiable recommenders, including auditors, RPs, and competitor TBs and (ii) ξ_s is a parameter associated with each recommender s. It is used to control the impact of each recommender on the final score where $\sum_{s\in S} \xi_s = 1$. This forms yet another component of the intellectual property of the TB.

For each trust factor, the personal dimension and the social one can be combined to obtain one trust score as follows:

$$\mathrm{Eval}\left(\varphi_j\right) = \alpha * \mathrm{QoCA}^{\mathrm{Personal}}\left(R_{\varphi_j}^{\varepsilon\to\mathrm{ca}}\right) + \beta * \mathrm{QoCA}^{\mathrm{Soc}}\left(\varphi_j\right), \tag{8}$$

where (i) α, β are configurable parameters that control the impact of the personal and social dimension on the final score, where $\alpha + \beta = 1$.

The ontological dimension allows the TB to calculate the final value of QoCA. The $\mathrm{QoCA}^{\mathrm{Onto}}$ value for a leaf or a node can be calculated as follows:

$$\mathrm{QoCA}^{\mathrm{Onto}}\left(\varphi_i\right) = \sum_{\varphi_j\in\mathrm{children}(\varphi_i)} \mu_{\varphi_i\varphi_j} * \mathrm{QoCA}^{\mathrm{Onto}}\left(\varphi_j\right), \tag{9}$$

where (i) $\mathrm{QoCA}^{\mathrm{Onto}}(\varphi_j) = \mathrm{QoCA}^{\mathrm{Soc}}(\varphi_j)$ when φ_j is an atomic trust factor, (ii) φ_i is a leaf or a node, and (iii) $\mu_{\varphi_i\varphi_j}$ is the weighting factor between nodes φ_i and φ_j in the tree CP/CPS. Each TB will have their own values for the various weighting factors.

The final value of QoCA is $\mathrm{QoCA}^{\mathrm{Onto}}$ for the factor CPSdocument which is the root of the CP/CPS tree:

$$\mathrm{QoCA} = \mathrm{QoCA}^{\mathrm{Onto}}(\mathrm{CPSdocument}). \tag{10}$$

4.3. Calculation of the Confidence Level (CL). The confidence level (CL) states to which extent the TB service is confident about the calculation of QoCA. Many factors can be considered, but here we considered the three major factors:

(i) *Number of Recommendations.* For each trust factor, a minimum threshold of recommendations is set. If the threshold is not reached then the reliability of the trust factor cannot be established. When a trust factor can only be evaluated by audit agencies, the threshold is set to 1, otherwise it should be greater than 1.

(ii) *Recommendations Heterogeneity*. The more the values of recommendations are homogenous, the more the calculation of QoCA is reliable. This factor is not taken into account when the recommender is an audit agency.

(iii) *Recommendation Dates.* The more recent the recommendations are, the more reliable they are. This parameter is considered for all types of recommenders.

The CL for the personal dimension and for the trust factor φ_j is the convex combination of the three functions representing the three aforementioned factors:

$$\begin{aligned}\mathrm{CL}^{\mathrm{Personal}}\left(R_{\varphi_j}^{\varepsilon\to\mathrm{ca}}\right) &= \gamma_N * N_i\left(R_{\varphi_j}^{\varepsilon\to\mathrm{ca}}\right) + \gamma_{Dt} \\ &\quad * Dt\left(R_{\varphi_j}^{\varepsilon\to\mathrm{ca}}\right) + \gamma_{\mathrm{Decay}} \\ &\quad * \mathrm{Decay}\left(R_{\varphi_j}^{\varepsilon\to\mathrm{ca}}\right),\end{aligned} \tag{11}$$

where (i) $\gamma_N + \gamma_{Dt} + \gamma_{\mathrm{Decay}} = 1$ allows the TB to control the impact of each factor on the final score and forms part of its intellectual property. (ii) $N_i(R_{\varphi_j}^{\varepsilon\to\mathrm{ca}}) = \{\sin((1/2 * itm)|R_{\varphi_j}^{\varepsilon\to\mathrm{ca}}|), |R_{\varphi_j}^{\varepsilon\to\mathrm{ca}}| \in [0, itm]; 1, \mathrm{Otherwise}\}$; (iii) $|R_{\varphi_j}^{\varepsilon\to\mathrm{ca}}|$ is the carinality of $R_{\varphi_j}^{\varepsilon\to\mathrm{ca}}$; (iv) itm is the threshold after which the confidence value always becomes 1; (v) $Dt(R_{\varphi_j}^{\varepsilon\to\mathrm{ca}}) = 1 - \sum_{i\in R_{\varphi_j}^{\varepsilon\to\mathrm{ca}}} |w_i - \overline{w}|$ gives a value between 0 and 1. The value 0 indicates that the recommendations are so different (i.e., not reliable). The value 1 indicates that the recommendations are homogenous and can be considered reliable. $\overline{w}$ is the average of the recommendations values. (vi) $\mathrm{Decay}(R_{\varphi_j}^{\varepsilon\to\mathrm{ca}})$ is a time-dependent function that gives values between 0 and 1. It is used to indicate the freshness of recommendations owned by an entity.

$\mathrm{CL}^{\mathrm{Soc}}$ represents the CL value after considering the social dimension. It can be calculated as follows:

$$\mathrm{CL}^{\mathrm{Soc}}\left(\varphi_j\right) = \sum_{s\in S}\xi_s \mathrm{CL}^{\mathrm{Personal}}\left(R_{\varphi_j}^{s\to\mathrm{ca}}\right), \tag{12}$$

where (i) S is the global set of identifiable recommenders, including auditors, RPs, and competitor TBs and (ii) ξ_s is a parameter associated with each recommender s. It is used to control the impact of each recommender on the final score where $\sum_{s\in S}\xi_s = 1$. This forms yet another component of the intellectual property of the TB.

For each trust factor, the personal dimension and the social one can be combined to obtain one trust score as follows:

$$\begin{aligned}\mathrm{Eval}\left(\varphi_j\right) &= \alpha * \mathrm{CL}^{\mathrm{Soc}}\left(\varphi_j\right) + \beta \\ &\quad * \sum_{s\in S}\xi_s \mathrm{CL}^{\mathrm{Personal}}\left(R_{\varphi_j}^{s\to\mathrm{ca}}\right).\end{aligned} \tag{13}$$

(i) α and β are configurable parameters that control the impact of the personal and social dimension on the final score, where $\alpha + \beta = 1$.

The ontological dimension allows the TB to calculate the final value of CL. The $\mathrm{CL}^{\mathrm{Onto}}$ for a leaf or a node can be calculated as follows:

$$\mathrm{CL}^{\mathrm{Onto}}(\varphi_i) = \sum_{\varphi_j\in\mathrm{children}(\varphi_i)} \mu_{\varphi_i\varphi_j} * \mathrm{CL}^{\mathrm{Onto}}\left(\varphi_j\right), \tag{14}$$

where (i) $\mathrm{CL}^{\mathrm{Onto}}(\varphi_j) = \mathrm{CL}^{\mathrm{Soc}}(\varphi_j)$ when φ_j is an atomic trust factor, (ii) φ_i is a leaf or a node, and (iii) $\mu_{\varphi_i\varphi_j}$ is the weighting factor between nodes φ_i and φ_j in the tree CP/CPS. Each TB will have their own values for the various weighting factors.

The final value of CL is $\mathrm{CL}^{\mathrm{Onto}}$ for the factor CPSdocument which is the root of the CP/CPS tree:

$$\mathrm{CL} = \mathrm{CL}^{\mathrm{Onto}}(\mathrm{CPSdocument}). \tag{15}$$

4.4. Discussion. Our trust model offers two principal advantages; first, it reflects the different points of view of TBs by allowing them to configure their own expertise into their computations. Second, it resolves the problem of interoperability by adopting a calculation method based on utility functions and weighting factors. Indeed, the utility functions $U(V_{\varphi_j})$ and the weight factors $\mu_{\varphi_i\varphi_j}$ allow several different strategies for trust calculations.

It should be noted that our evaluation system is designed to consider not only technical issues but also juridical ones. Thus, it is impossible to fix these trust metrics as they reflect the flavour of TBs and their own expertise and preferences. It is true that some trust factors can be objectively measured, but their relevance for a given application remains a subjective matter. For example, it is clear that SHA-512 is stronger than SHA-256 for the trust factor of the hash algorithm. But the relevance of SHA-256 for an application such as web server authentication is still a subjective question for experts. Chadwick and Basden [35] have demonstrated this phenomenon by asking PKI experts to prioritize the

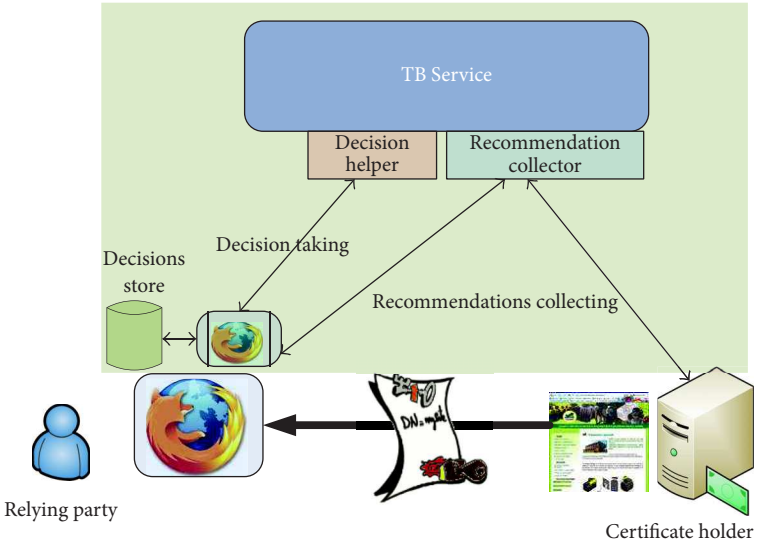

Figure 9: The prototype architecture.

PKIs trust factors. The study concluded by demonstrating the difficulty in reaching a consensus among experts. Chadwick and Basden gave several reasons for this difficulty.

Our trust model enables the resolving of interoperability problems. In fact, TBs can adapt their calculation models to meet the needs of their clients as well as the context of use. For example, in Greece, a certificate has legal effect, if the retention period of electronic records is over 30 years, while in Spain it is only 15 years [36]. The utility function that processes this trust factor can give a high importance for a certificate that has a retention period of 15 years when the certificate is used in Spain and low importance when the certificate is used in Greece. Another example is the use of pseudonyms in certificates. Most European countries, except Estonia and Bulgaria [36], authorize the use of pseudonyms in certificates. The TB service is able to deal with this problem of interoperability between European countries using utility functions. When a certificate with a pseudonym is used in Estonia or Bulgaria, the utility function that processes this trust factor gives a value of 0 for this certificate.

Finally, our calculation system prevents collusion from unidentified entities. However, collusion from identified entities is difficult to prevent, especially if the number of conspirators is greater than the number of honest entities. However, each TB will be contractually linked to the identified entities (audit agencies, other TBs, and users). If something goes wrong, the conspirators can be prosecuted.

5. Prototype in the Web Context

We have implemented a prototype in the context of the web, where the RPs are human entities accessing various websites via a web browser. Certificate holders are the users who purchase certificates for their web servers. Our prototype TB web service comprises two principal components (as shown in Figure 9):

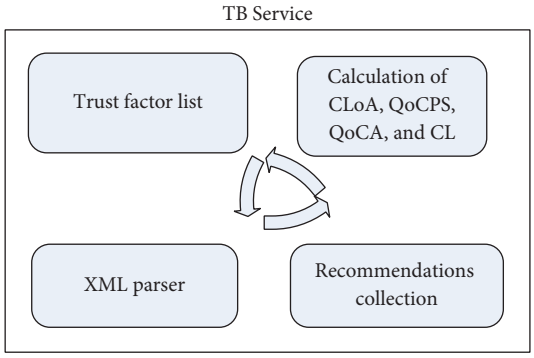

Figure 10: The TB service modules.

(i) Decision helper component: this component helps RPs make contextually informed decisions about certificates.

(ii) Recommendation collector component: this component collects recommendations sent by client RPs.

The TB service consists of the following modules (Figure 10):

(i) The trust factors list module contains the TB's list of trust factors in XML format.

(ii) The XML parser module processes the XML trust files provided by either the cooperating CAs or the TB (for noncooperating CAs). Each trust file contains the answers to the list of trust factors for one CA, reflecting its certification policies and procedures (i.e., values for the trust factors).

(iii) The recommendations collection module collects the recommendations sent by client RPs.

(iv) The calculation module computes the CLoA, QoCPS, QoCA, and CL.

```
<!ELEMENT KeySizes (f1_615, f2_615, f3_615, f4_615)>
<!ATTLIST KeySizes id (6.1.5) #REQUIRED>
<!ELEMENT f1_615 (#PCDATA)>
<!ATTLIST f1_615 KeySize (256| 512| 1024| 2048| 4096| other)
#REQUIRED algorithm (RSA| ElGamal | Merkle-Hellman | other) #REQUIRED>
<!ELEMENT f2_615 (#PCDATA)>
<!ATTLIST f2_615 Hash (SHA-1 | SHA-224| MD5 | other) #REQUIRED>
<!ELEMENT f3_615 (#PCDATA)>
<!ATTLIST f3_615 KeySize (256| 512| 1024| 2048| 4096| other)
#REQUIRED algorithm (RSA| ElGamal | Merkle-Hellman | other) #REQUIRED>
<!ELEMENT f4_615 (#PCDATA)>
<!ATTLIST f4_615 Hash (SHA-1| SHA-224| MD5 | other) #REQUIRED>
```

Listing 1: The Key Size trust factor for a particular TB service.

```
<KeySizes id=" 6.1.5 ">
        <f1_615 KeySize=" 1024" algorithm="RSA"/>
        <f2_615 Hash="MD5"/>
        <f3_615 KeySize=" 1024" algorithm="RSA"/>
        <f4_615 Hash="MD5"/>
</KeySizes>
```

Listing 2: An example trust data structure for a particular CA.

```
<KeySizes id="6.1.5" >
        <f1_615 KeySize=" 1024" algorithm="RSA"/>
        <f2_615 Hash=" other"> SHA-512</f2_615>
        <f3_615 KeySize=" 1024" algorithm="RSA"/>
        <f4_615 Hash="MD5"></f4_615>
</KeySizes>
```

Listing 3: An example trust data structure returning a value unknown to the TB service.

In addition we have implemented a Firefox extension module that modifies the way that Firefox handles certificates, by first communicating with the TB service. The TB service interacts with CAs and with RP clients. In the following section we give more details about these interactions.

5.1. Interaction between TB Service and CAs. The list of trust factors is made public by the TB service. It is constructed using the XML DTD format. For example, we have defined the DTD component for the leaf "Key Sizes" as shown below in Listing 3. This depicts the structure of the leaf "Key Sizes" in the CP/CPS tree. It contains four atomic factors of trust: "the size of the CA's public key and the key algorithm," "the hash algorithm used for the CA's certificate," "the size of the public key of the user's certificate and its key algorithm," and "the hash algorithm used for the user's certificate."

Each cooperating CA picks up the trust factor list and returns an XML file that contains answers to the atomic factors (see Figure 11). The cooperating CA sends this file back to the TB service in order for it to determine the QoCPS. For noncooperating CAs, the TB must retrieve the CA's CP/CPS and answer the questions themselves. (S)he can then submit the resulting file to the XML parser. The XML component for the node ≪Key Sizes≫ trust element can take the form presented in Listing 2. If a CA uses a value that is not already referenced in the DTD list of trust factors, then the actual value must be labelled as "other" before the trust file is returned to the TB (see Listing 3). For example, for the trust factor "Key Sizes," a CA may use the value "SHA-512." This

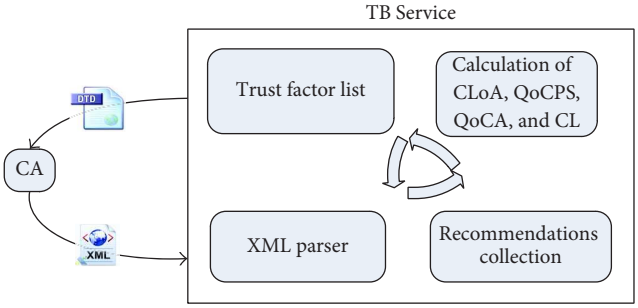

Figure 11: Interaction of TB service with CAs.

value is not included in Listing 1 for this particular TB service, so the QoCPS cannot be analysed automatically. When the TB service receives the trust file, it should update the utility function for this new value and update the list of referenced values in the DTD.

5.2. Interaction between TB Service and Clients. There are two types of interactions between the TB service and its RP clients: interaction for helping clients to make informed decision about certificates and interaction for receiving recommendations from clients about certain trust factors of CAs.

5.2.1. Helping Clients to Make Informed Decision. There are four main actors in this case (Figure 12):

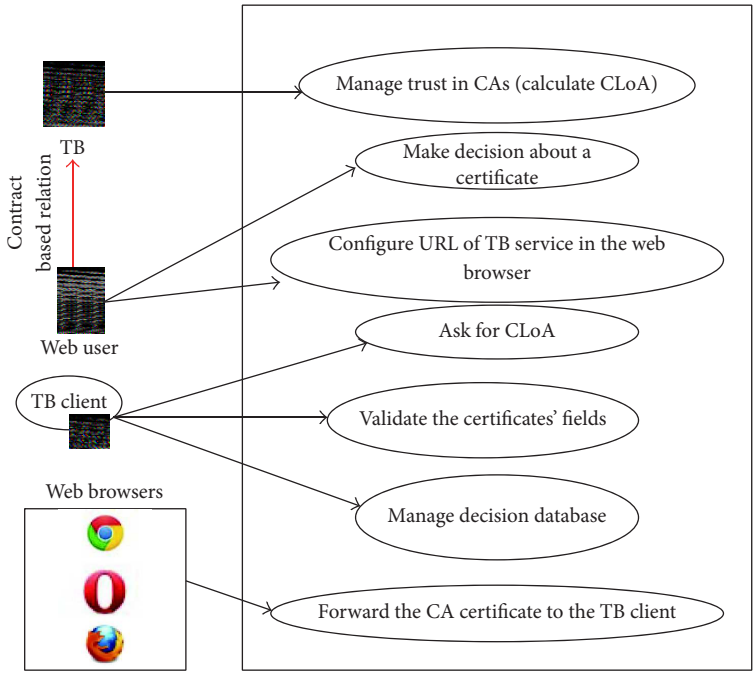

Figure 12: The main actors in 4-cornered based validation system.

(i) The RP: the end user that will ultimately make the decision to accept or reject a certificate. The RP chooses its own web browser.

(ii) The browser extension module (TB client): the entity that allows RPs to communicate with TB services. It manages the decisions of RPs about certificates. It should also provide an interface to enable RPs to select the TB service that they wish to depend on. Our objective is to make web browsers totally transparent in the decision-making process.

(iii) The TB service: the entity that calculates the certificate quality information.

(iv) The web browser: the entity that forwards the CA certificate to the TB client.

Figure 13 illustrates the different steps that we have implemented to realize the validation of certificates.

When a user contacts a web service via an TLS connection, for example, to make an online payment, the user wants to be assured that their information is sent to the correct web service and that any received information comes from the right service.

The web server initially sends its certificate to the web browser. It is not possible at this point in time for the web browser to discover the RP's intended use of the certificate. This is because the server's web page can define several contexts of certificate use. For example, when a user visits the site https://www.somesite.com, the homepage may have two forms that define two different contexts of use; the first form allows the user to connect to the server https://www.login.somesite.com and the second form allows the user to register by sending information to the server https://www.register.somesite.com. Before sending them personal or confidential data to either of these servers, the user's browser must retrieve the server's address and extract the certificate that is used to determine if (s)he can trust the certificate of the server for the specific context. We have identified three different contexts of use for a user sending personal or confidential data to a web server:

(1) Connection login: the user will send a username and password to the web server and thereafter will establish a secure session during which many different types of transaction may take place, for example, database access, file access, and online transaction.

(2) Registration: the user will send her personal information to the web server in order to create an account.

(3) Payment: the user will send her bank account or credit card details to the server in order to buy a product. The user may or may not be logged in.

Our extension module monitors the behaviour of the user to see when (s)he will send their personal information to a server. When the user clicks on the send button, the module extension suspends the sending, extracts the certificate, and requests the quality information about the certificate from the TB service. The TB service returns the quality information signed by its public key (which was configured into the extension when the user chose her trusted TB service). The module extension will then filter the returned information based on the certificate's context of use. When the context of use cannot be determined then it shows the most informative/qualitative information about the certificate to the user.

The browser extension tries to automatically determine the context of use based on the content of the personal data

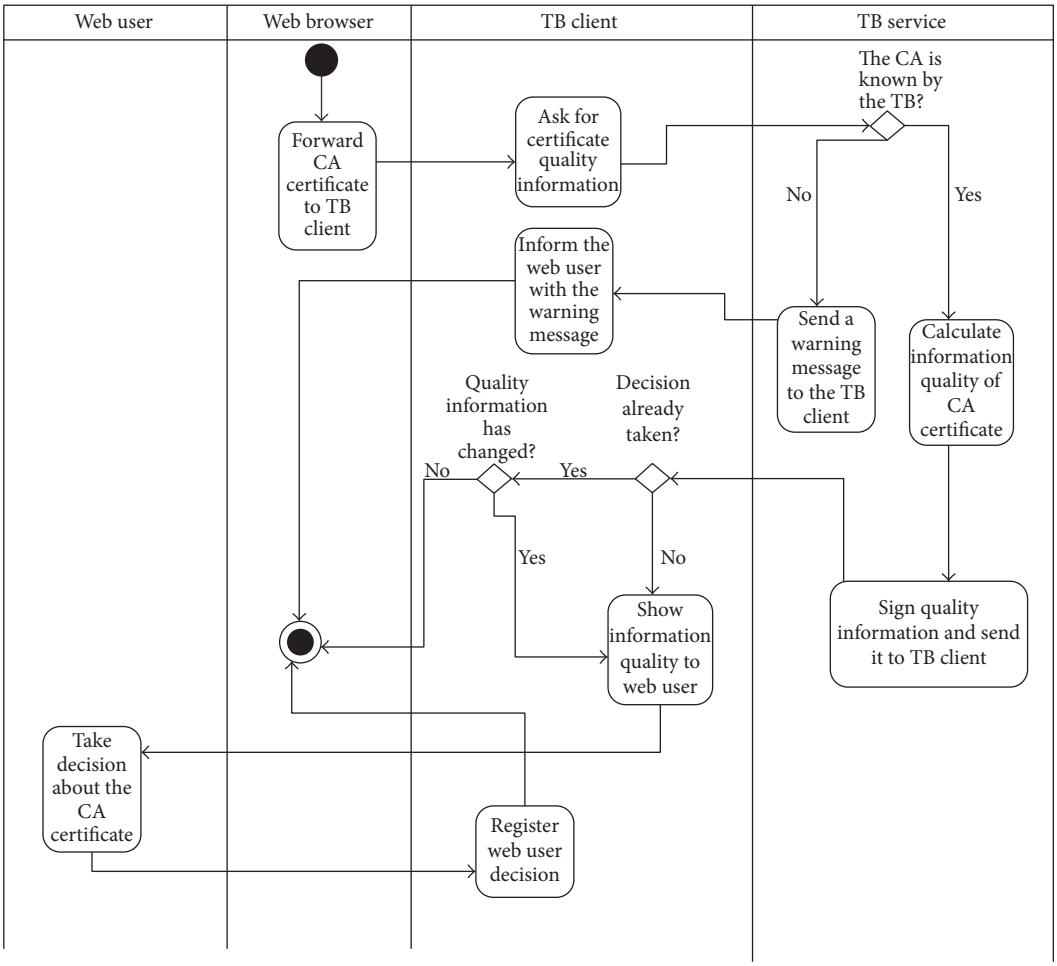

Figure 13: Illustration of certificate validation steps.

being requested. For example, when a single field of type "password" exists in a web form, we assume the form is used to allow users to login to the server. When there are two fields of type "password," we assume the form is used for the registration of new users. When there is a field of type credit card type or credit card number, we assume a payment is about to be made. However, we recognize that this is not always the case and that there are other cases where we cannot determine the intended use of the certificate. Consequently, we have prepared a simple questionnaire (see Figure 15) that appears when the user first wants to send information to a server over a TLS connection. This allows the user to confirm or alter the automatically determined context of use.

The quality information returned to the user in case of connection login (see the GUI in Figure 14) is as follows:

(i) CLoA, called Identity Assurance in the GUI

(ii) The name of the certificate holder, called Contacted Web Site in the GUI

(iii) The name of the certification authority, called Identity Verified By in the GUI

(iv) A link towards the certificate's CP/CPS, called Agreement of Certificate Usage in the GUI

Figure 14: Making a decision about a certificate used for connection login.

(v) The financial protection offered to the client by the CA, in case of false certificate information

Any additional information that might be important to the client is sent by the TB service. For example, when the user wants to buy a product, the allowed maximum money to pay for the purchase is critical information (Figure 16). If the

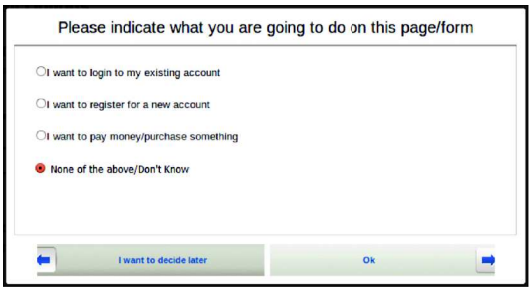

Figure 15: Determining the context of use for a certificate.

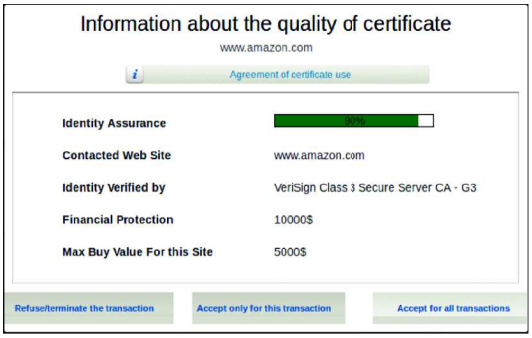

Figure 16: Making a decision about a certificate used for providing payment information.

client wants to purchase a product whose price is higher than the protection offered by the CA in its CP, then he/she loses the right to be fully covered by financial protection. Thus it is important to inform the client about this fact.

When the user makes a decision about a certificate, we record this decision and the context of use in a local store of the web browser along with the CA's certificate and CLoA score. In this way, the extension module will not ask the user to remake a decision about the same context of use for any web server certificate issued by the same CA. However, the extension will ask the user to remake a decision about any of the CA's issued certificates if its quality information has changed since the last time the decision was made or the context of use is different.

When the user wants to connect to or register with a server, the allowed decisions are only "accept" or "reject" the certificate. But when the client wants to send information related to a commercial transaction, the decision becomes "refuse/terminate the transaction," "accept only for this transaction," or "accept for all transactions." We have added the middle option allowing a decision to be taken for each transaction because the risks associated with a payment vary according to the amount of the transaction. We therefore adapt our user interface to the context of the certificate's use.

5.3. Recommendations Collecting. The TB service can automatically collect recommendations when its RP clients are using server certificates. There are several trust factors that can be automatically verified by the browser extension. In our prototype, we monitor the profiles of certificates and the availability of OCSP servers.

The TB service sends to the browser extension module the profile of certificates issued by the server's CA. The extension checks the server's certificate against the certificate profile and notifies the TB service when a violation is detected.

In addition, we verify the requirements imposed on certificate profiles of "extended validation" certificates. According to the guidelines of the CA/Browser Forum [37] Extended Validation (EV) certificates must meet a number of strict requirements concerning the subject name, the cryptographic algorithms, the key usage field, and revocation information. Our TB service defines the XML profile presented in Listing 4 that contains all these requirements.

When the browser extension detects any problem related to a certificate's profile, it asks the user for permission to send the certificate to the TB service, along with the identified problem. In addition, the browser extension also checks the availability of OCSP servers and reports to the TB service when one is not available.

Before recording any recommendations about a CA, the TB service checks if they are correct or not. If correct, it records a positive or negative recommendation about the concerned trust factor.

6. How Can the 4-Cornered Trust Model Improve the Security of Web Users?

Superfish and the other incidents demonstrate one fact: no one has assumed the responsibility or liability for any consequences related to these incidents. From a theoretical point of view, only the CAs are liable to the web users in case of problems. In practice, web users are not able to prosecute CAs because they are not technically able to prove the responsibility of CAs.

Nevertheless, even when the web users are able to prove the responsibility of CAs, they will not be protected completely. Superfish and eDellRoot incidents have shown how it is possible to intercept the TLS communications of web users without necessarily compromising the systems of CAs or asking them to issue false certificates.

The main reason for this situation is that the web TLS system is designed so that web users must depend on a multitude of entities, other than CAs, in order to secure their transactions. As an example we will compare the intermediate entities in the case of the current validation system with the entities implied in the case of our proposal.

Figure 17(a) shows an example of the entities that are intervening in the current validation system. This represents a web user that uses a Windows OS and uses different web browsers for surfing the web, including Firefox. The first important issue to note is that web users do not have any assigned tasks to achieve. Everything is executed without their knowledge. From a usability point of view, this issue can be seen as an advantage. However, from the security point of view, the current validation system compromises the security of web users because they depend on unknown entities to secure their transactions. In Figure 17(a), we recognize that

```
<?xml version= 1 .0? >
<CertProfile>
        <! Verisign CA id of EV Certificate −−>
        <CAId value=" 3 C : 48 : 42 : 0D : FF : 58 : 1A : 38 : 86 : BC : FD : 41 : D4 : 8A : 41 : DE" />
        <Profile Version value=" 5.0" />
        <Subject type=" field" component="O" presence=" Obligatory" />
        <Subject type=" field" component="CN" presence=" Obligatory"
        value=" dnshostname" valueExclude="∗" />
        <Subject type=" field" component="C" presence=" Obligatory" />
        <Subject type=" field" component="L" presence=" Obligatory" />
        <Subject type=" field" component="ST" presence=" Obligatory" />
        <! −− : this field MUST contain the Registration (or similar)
        Number assigned to the Subject by the Incorporating or Registration
        Agency in its Jurisdiction of Incorporation or Registration−−>
        <Subject component=" Object␣Identifier.∗2␣5␣4␣5.∗" Type=" field"
        presence=" Obligatory"/>
        <!−− : The validity period for an EV Certificate SHALL NOT exceed
        twenty seven months.−−>
        <Validity type=" field" value=" 27" />
                <DigestSignatureAlgorithm value=" (SHA-1|SHA-256|SHA-384|SHA-512)" />
                <KeySize component=" Key␣Size" value=" (1024|2048)" />
        <!−− : MUST be present and SHOULD NOT be marked critical. The set of
        policyIdentifiers MUST include the identifier for the CAs extended
        validation policy.−−>
        <Certificate␣Policies type=" extension" critical=" Not␣Critical"
        presence=" Obligatory" value=" oid" />
        <!−− : SHOULD be present and MUST NOT be marked critical. It MUST
        contain the HTTP URL of the CAs CRL service. This extension MUST
        be present if the certificate does not specify OCSP responder.−−>
        <CRL␣Distribution␣Point type=" extension" critical=" Not␣Critical"
        presence=" Obligatory" value=" httpservicehost" />
        <!−− : SHOULD be present and MUST NOT be marked critical. SHALL
        contain the HTTP URL of the CAs OCSP responder. This extension
        MUST be present if the certificate does not contain a
        cRLDistributionPoint extension.−−>
        <Authority␣Information␣Access type=" extension" critical=" Not␣Critical"
        presence=" Obligatory" value=" httpservicehost" />
        <!−− : the presence of key usage extension is optional. If present,
        the CA field MUST be set false.−−>
        <Basic␣Constraints type=" extension" critical=" Not␣Critical"
        presence="optional" value=" false" />
        <!−− : the presence of key usage extension is optional. If present,
        bit positions for keyCertSign and cRLSign MUST NOT be set−−>
        <key␣Usage type=" extension" critical=" Not␣Critical"
        presence=" Optional" valueExclude=" (Certificate␣Signer|CRL␣Signer)" />
</ CertProfile>
```

Listing 4: Example of XML profile for EVS certificate.

Microsoft and Mozilla are the "official" TBs because they are the entities that realize tasks 1 and 2 on behalf of the web users. We have put the term official in quotation marks because the web users did not delegate officially Microsoft or Mozilla to achieve tasks 1 and 2 on behalf of them. If something goes wrong, Microsoft and Mozilla will not refund web users in case of problems. In addition, certificate distributers do not have the obligation to help web users to prosecute malevolent CAs in case of loss.

The Superfish incident was produced because Lenovo had an access to the list of CAs provided by Microsoft. Lenovo injected the certificate of a self-signed CA and used its software Superfish to intercept the TLS communications of Lenovo users (i.e., MITM attack). Naturally, users are not able to detect such kind of problems, because the current validation system is designed so that everything is executed transparently without the knowledge of web users.

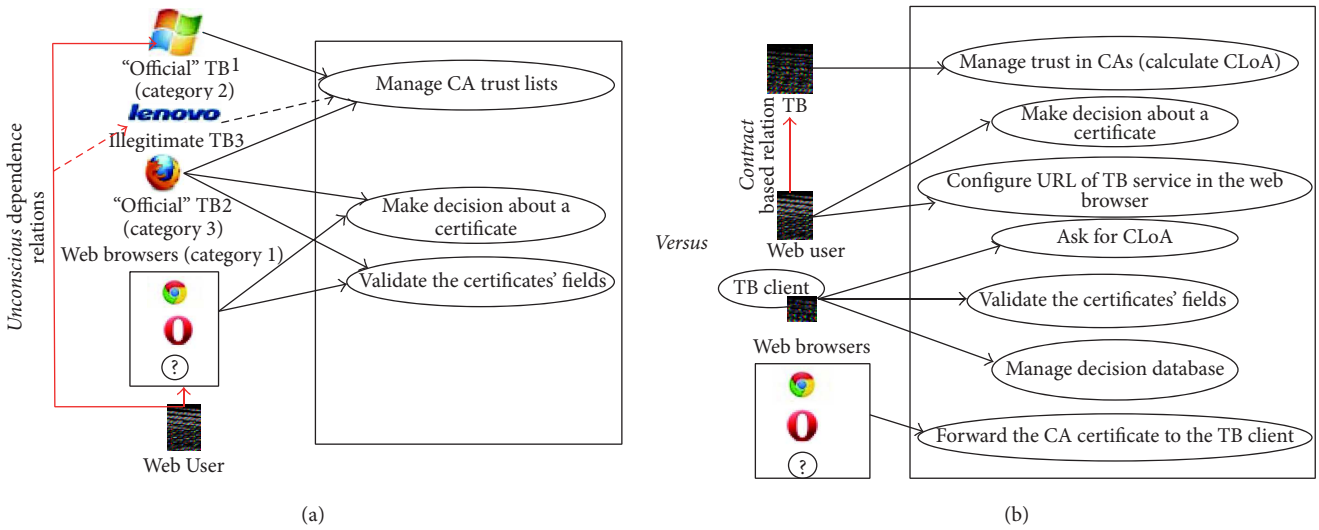

FIGURE 17: A comparison between the current validation system and the 4-cornered validation system.

The MITM attacks would not be possible if the integrity of the trust list is guaranteed, that is, ensuring that the OS/Browser editors and web users are the only authorized entities to modify the trust lists. Unfortunately, this is not enough because web users have to use web browsers to complete the validation process. Web users may use known web browsers such as Google Chrome and Firefox, but they may use also unknown ones for completing the validation process. Georgiev et al. [38] demonstrate that SSL certificate validation is completely broken in many security-critical applications and libraries. The main reason is that developers tend to badly configure the APIs of different SSL implementations (such as JSSE, OpenSSL, and GnuTLS) and data-transport libraries (such as cURL).

Finally, from the usability point of view, the current validation system confuses the web users. In the current validation system, the roles that the intermediate entities should play are not clearly defined. Any intermediate entity has the right to execute one or more of the obligations of web users. For example, Firefox has selected to achieve the four tasks whereas Google Chrome achieves only tasks 3 (partially) and 4. If one web user uses Google Chrome on two different platforms (e.g., Windows and Linux) (s)he may get different validation results for the same website because the trust list of Windows is not the same as Linux. On the other side, if one web user uses Firefox and Google Chrome on the same platform (e.g., Windows) (s)he may get also two different validation results for the same website because the trust list of Mozilla is not the same as Windows.

The 4-cornered trust model solves the aforementioned problems by creating a physical relation between the RP and the TB, who is a technical and legal expert in the domain of PKI. RPs (web users) need only one certificate, which is the public key certificate of the TB. Instead thus of having to observe the integrity of infinite trust lists, RPs need only to observe the integrity of one public key certificate, regardless of the nature and number of platforms and applications used by them.

With the 4-cornered trust model, whenever a user gets a new platform, (s)he should link their platform to the validation service proposed by the TB with whom the user has a contract. It should be possible to choose from a set of available TBs. Configuring the TB consists only in setting the URL and the certificate of the TB service. Any software installed on the platform of the user must use the selected validation service. The OSs have to remove the ability of software to realize validation services, especially when a user selects a TB.

With 4-cornered validation system, the web browsers are completely neutralized. The only task that they should make is to forward the certificate of the website to the TB client. Thus, if the web user has chosen an unknown web browser to surf the web, this will not affect the security of his TLS transactions (Figure 17(b)).

In real scenarios, the TB might be misconfigured or has poor software. This will not cause problems for the web users because the web user is insured by the TB for any loss or damage (s)he may suffer. Therefore, the user does not lose out if the TB makes a mistake for any reason. This is not the case for a misconfigured CA trust list, where the user does lose out. The only way to compromise the security of user's transactions is to misconfigure the TB setting. This can be easily protected. For example, the X.509 V3 certificate has an extensions section that allows adding additional information to X.509 certificate. Consequently, TB may include a new extension in its certificate that contains the URL of its validation service. During each request, the TB client shall compare the requested URL with the one contained in the certificate of the TB. The advantage of our proposal is that each user is linked to only one TB and not to multiple TBs. It is a fundamental improvement of the PKI's trust model that has been adopted in the new X.509 standard.

In addition, multiple implementations might be proposed by different TBs. This is good from a security perspective. If one implementation is used by all TBs, one flaw in this implementation means that all TBs are flawed.

7. Conclusion and Future Works

The original X.509 trust model is only appropriate for the closed deployment model of PKIs, in which the RPs and subjects both have predefined relations with the CAs. It is not appropriate for the open deployment model where the RP has no explicit relationship with any CA.

The existing trust approaches are not adapted to the needs of RPs to make informed decisions in the PKI open deployment model. As a result, PKIs remain isolated islands in the open model. Each PKI seeks to comply only with the requirements of the jurisdiction where the premises of its root CA are located. Thus, the RPs have to handle this PKI (lack of) interoperability issue. The various harmonization attempts at regional and international level have not come up with a solution to the PKI interoperability problem.

PKI trust management is extremely complex; therefore only technical and legal experts can perform it. It is not conceivable to delegate this task to the RPs who generally are "normal" people. Thus, X.509 has defined a new entity in its upcoming trust model, called the Trust Broker (TB), who is a technical and juridical expert. This new approach to trust management is applicable to both the closed and open deployment models of PKI. We have defined six criteria for trust evaluation that this approach should follow.

Contrary to the 3-cornered trust model, the applicability of the 4-cornered model does not depend on the good practices of the CAs and their certificate holders. The TB varies its scores about the CAs, according to their commitments to their policies and to their commitments for ensuring that their certificate holders remain responsible or take appropriate actions in case of malpractice.

Based on these six criteria, we have proposed a trust calculation model, which identifies and quantifies the quality of certificates. This quality information is represented by a score between 0 and 1 indicating the quality of the certificate (CLoA). This is computed from the quality of the CA's procedures (QoCPS), the ability of the CA to conform to its published procedures, and a confidence level between 0 and 1 indicating the reliability of the quality calculation and other information that depends on the context of use of the certificate. To calculate the values of CLoA we have proposed the following:

(i) To transform policy documents (CP/CPS) into a format that can be understood by computers since the natural language used to describe the policy of a CA is one of the main obstacles to determining trust in a CA

(ii) Based on RFC 3647, we have structured CP/CPS documents as a hierarchical tree composed of nodes and leaves

(iii) To integrate the role of RPs, certificate holders, and competitor TB services to supplement the work of audit agencies in evaluating the real practices of CAs

(iv) To calculate the quality of CAs (QoCA) based on the REGRET model, while allowing TBs the flexibility to incorporate their own intellectual property in order to gain competitive advantage over other TBs

The trust relationship between the RP clients of a TB service and the TB is an important issue. We propose that it should be a contractual relationship that gives warranties and commitments to the RP clients. The trust that a client must have in its TB should not be based on the evaluation strategy adopted by the TB. Chadwick and Basden [35] has showed that even PKI experts cannot reach a consensus about the importance of a CA's security parameters. As a consequence, the RP's trust should be that the contracted TB will make its best efforts to protect them, but if something goes wrong, the warranties of the signed contract will effectively alleviate the problem.

Finally, we have chosen to implement a prototype system in the context of the web to show how Internet users can make informed decisions about web server certificates, aided by a TB, without compromising their privacy.

In the short-term, we propose to conduct some usability experiments to measure the advantages that our prototype offers to end users.

In the long-term, further work must be conducted to enrich the assurance information presented to users. The CLoA information is important but not sufficient to give the overall assurance level for a transaction. The provision of assurance services requires the intervention of other entities whose natures and roles depend on the context of a transaction between the interested parties. For example, attribute authorities may be required to assure end users that the services do possess certain attributes. The role of these entities can be extended according to the context of a transaction. We need to extend this work to deal with all elements of the chain of a transaction. Each element, depending on its role, must meet certain criteria. Thus, instead of providing only the CLoA score, the system could provide a score including the level of quality of all elements of the chain.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] Lenovo, "Superfish attack," 2015, https://twitter.com/kennwhite/status/568270748638318593.

[2] Lenovo, Guidelines for removing superfish, 2015, https://support.lenovo.com/fr/en/product_security/ps500066.

[3] Dell, Dell edellroot, 2015, http://www.dell.com/support/article/us/en/19/SLN300321.

[4] Dell, "Guidelines for removinf the ca of dell," 2015, http://www.dell.com/support/article/fr/fr/frbsdt1/SLN300321?c=fr&l=fr&s=bsd&cs=frbsdt1.

[5] Z. Ye, S. Smith, and D. Anthony, "Trusted paths for browsers," *ACM Transactions on Information and System Security*, vol. 8, no. 2, pp. 153–186, 2005.

[6] J. Marchesini and S. Smith, "Modeling public key infrastructures in the real world," in *Public Key Infrastructure*, D. Chadwick and G. Zhao, Eds., vol. 3545 of *Lecture Notes in Computer Science*, pp. 118–134, Springer, Berlin, Germany, 2005.

[7] A. Jøsang, I. Glenn Pedersen, and D. Povey, "PKI seeks a trusting relationship," in *Information Security and Privacy: 5th Australasian Conference, ACISP 2000, Brisbane, Australia, July 10–12, 2000. Proceedings*, vol. 1841 of *Lecture Notes in Computer Science*, pp. 191–205, Springer, Berlin, Germany, 2000.

[8] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*, pp. 581–590, ACM, Montréal, Canada, 2006.

[9] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, and L. F. Cranor, "Crying wolf: an empirical study of SSL warning effectiveness," in *Proceedings of the 18th Conference on USENIX Security Symposium (SSYM '09)*, pp. 399–416, USENIX Association, Montreal, Canada, August 2009.

[10] R. Dhamija and L. Dusseault, "The seven flaws of identity management: usability and security challenges," *IEEE Security and Privacy*, vol. 6, no. 2, pp. 24–29, 2008.

[11] R. Biddle, P. C. Van Oorschot, A. S. Patrick, J. Sobey, and T. Whalen, "Browser interfaces and extended validation SSL certificates: an empirical study," in *Proceedings of the ACM Workshop on Cloud Computing Security (CCSW '09)*, pp. 19–30, New York, NY, USA, November 2009.

[12] A. S. Wazan, R. Laborde, D. W. Chadwick, F. Barrere, and A. Benzekri, "Which web browsers process ssl certificates in a standardized way?" in *Emerging Challenges for Security, Privacy and Trust*, D. Gritzalis and J. Lopez, Eds., vol. 297 of *IFIP Advances in Information and Communication Technology*, pp. 432–442, Springer, Berlin, Germany, 2009.

[13] N. Luhmann, "Familiarity, confidence, trust: problems and alternatives," in *Trust: Making and Breaking Cooperative Relations*, D. Gambetta, Ed., chapter 6, pp. 94–107, Department of Sociology, University of Oxford, 2000.

[14] Google, Certificate transparency project, 2015, https://www.certificate-transparency.org.

[15] L. Chuat, P. Szalachowski, A. Perrig, B. Laurie, and E. Messeri, "Efficient gossip protocols for verifying the consistency of certificate logs," in *Proceedings of the 3rd IEEE International Conference on Communications and Network Security (CNS '15)*, pp. 415–423, IEEE, Florence, Italy, September 2015.

[16] Electronic Frontier Foundation, Sovereign keys project, 2015, https://www.eff.org/fr/sovereign-keys.

[17] R. Sleevi, C. Evans, C. Palmer, and Google Inc, "Public key pinning extension for HTTP," Tech. Rep. Rfc7469, 2015, https://tools.ietf.org/html/rfc7469.

[18] A. S. Wazan, R. Laborde, F. Barrere, and A. Benzekri, "The X.509 trust model needs a technical and legal expert," in *Proceedings of the IEEE International Conference on Communications (ICC '12)*, Ottawa, Canada, June 2012.

[19] A. S. Wazan, R. Laborde, F. Barrere, A. Benzekri, and D. W. Chadwick, "PKI interoperability: still an issue? A solution in the X.509 realm," in *Information Assurance and Security Education and Training*, pp. 68–82, Springer, Berlin, Germany, 2013.

[20] ITU, *Current Status of the Eighth Edition of x.509 Standard*, 2016, http://www.itu.int/itu-t/aap/AAPRecDetails.aspx?AAPSeqNo=5686.

[21] W. A. Samer, L. Romain, B. Francois, and B. AbdelMalek, "A formal model of trust for calculating the quality of X.509 certificate," *Security and Communication Networks*, vol. 4, no. 6, pp. 651–665, 2011.

[22] W. T. Polk and N. E. Hastings, "Bridge certification authorities: connecting b2b public key infrastructures," 2000, http://csrc.nist.gov/groups/ST/crypto_apps_infra/documents/B2B-article.pdf.

[23] Gatekeeper PKI Framework, "Cross recognition policy," 2009, https://www.finance.gov.au/sites/default/files/Cross_Recognition_Policy.pdf.

[24] Mozilla, "Mozilla CA certificate inclusion policy," https://www.mozilla.org/en-US/about/governance/policies/security-group/certs/policy/inclusion/.

[25] Microsoft, Microsoft root certificate program, 2009, http://technet.microsoft.com/en-us/library/cc751157.aspx.

[26] S. P. Marsh, *Formalising trust as a computational concept [Ph.D. thesis]*, Department of Computer Science and Mathematics, University of Stirling, 1994.

[27] M. Deutsch, *Cooperation and Trust: Some Theoretical Notes*, Nebraska University Press, 1962.

[28] N. Luhmann, *Trust and Power*, John Wiley & Sons, 1979.

[29] B. Barber, *Logic and Limits of Trust*, Rutgers University, 1983.

[30] A. Baier, "Trust and antitrust," *Ethics*, vol. 96, no. 2, pp. 231–260, 1986.

[31] D. Gambetta, "Can we trust trust?" in *Trust: Making and Breaking Cooperative Relations*, D. Gambetta, Ed., pp. 213–237, Blackwell, 1988.

[32] A. Jøsang, "The right type of trust for distributed systems," in *Proceedings of the the workshop on new security paradigms (NSPW '96)*, pp. 119–131, Lake Arrowhead, Calif, USA, September 1996.

[33] A. S. Wazan, R. Laborde, F. Barrére, and A. Benzekri, "Validating X.509 certificates based on their quality," in *Proceedings of the 9th International Conference for Young Computer Scientists (ICYCS '08)*, pp. 2055–2060, IEEE, Hunan, China, November 2008.

[34] J. Sabater and C. Sierra, "Regret: reputation in gregarious societies," in *Proceedings of the 5th International Conference on Autonomous Agents (AGENTS '01)*, pp. 194–195, Montreal, Canada, June 2001.

[35] D. W. Chadwick and A. Basden, "Evaluating trust in a public key certification authority," *Computers & Security*, vol. 20, no. 7, pp. 592–611, 2001.

[36] J. Dumortier, S. Kelm, H. Nilsson, G. Skouma, and P. van Eecke, "Study for the European Commission: the legal and market aspects of electronic signatures," Tech. Rep., European Commission, 2003.

[37] CABrowser Forum, "Guidelines for the issuance and management of extended validation certificates v1.5.2," https://cabforum.org/wp-content/uploads/EV-V1_5_2Libre.pdf.

[38] M. Georgiev, S. Iyengar, S. Jana, R. Anubhai, D. Boneh, and V. Shmatikov, "The most dangerous code in the world: validating SSL certificates in non-browser software," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS '12)*, pp. 38–49, ACM, Raleigh, Calif, USA, October 2012.