A Distributed and Collaborative Approach for Evaluating New Root CA Certificates

Mohammad A. Asmaran, Sulieman Bani-Ahmad
Department of Information Technology
Al-Balqa Applied University
Al-Salt, Jordan
m_asmaran@hotmail.com, sulieman@bau.edu.jo

Abstract- A Certification Authority (CA) is a trusted organization that digitally signs certificates according to a standard process to ensure the identities of their sub-certificate owners. CA's are formed as tree of parties signing each other except top parent nodes that are called the Root CAs. In this tree, any certificate at any level should be verified through the root CA through its direct parent (the super CA) CA. verifying root CAs is quite different; *Cross Certification* is used to generate certificates between two root CA's to ensure the identity of each other. This, however, has multiple security threats. In this paper, a new collaborative approach is proposed for certificating *new* root CAs. In the proposed approach *expert users*, e.g. proxy servers' administrators, collaboratively participate in the process of validating new root CAs. The advantages of the proposed certification mechanism are (i) providing help to naïve or inexperienced users to evaluate certificates before installing them to their systems. (ii) Helping root CAs to publish their certificates according to a collaborative human-based experience without any help of previously installed certified root CAs. (iii) Preventing the installation of *faked* root CAs certificates by determining the level of trust of new root CA.

Keywords: Operating systems; Security; Certification Authority; Root CA; Cross Certification.

I. INTRODUCTION

Security is a very sensitive term in network and Internet. Security is currently maintained through the usage of various types of software. Examples are: firewalls, antivirus, anti-spywares …etc. Such software types use various approaches of approaches to provide security to Internet users.

The Secure Socket Layer (SSL) protocol [1] is used as a base in variety of Internet protocols such as Secure Hyper Text Transfer Protocol (HTTPS) [2]. HTTPS is usually used to secure user sign-in in addition to providing a verity of secure services over the Internet such as installation of signed software and drivers [4]. This process is used to increase the sense of security over the Internet which is used as a proof to user that a given website is *safe and secure*.

In order to deliver secure websites and services SSL-based protocols are used. SSL protocol uses *Asymmetric Key Algorithms* such as RSA by Rivest, Shamir and Adleman [3]. The goal is to initiate secure connection between the client and the server by exchanging Symmetric Keys. This is done to increase the speed of secure-connections by minimizing the complexity in terms of time and space. Any symmetric algorithm won't work without initial key exchange which is difficult and impossible over the Internet [4]. For this issue, asymmetric key algorithms are usually used to exchange symmetric keys first to ensure the security of the exchange process and then the remaining process of the connection is handled by the symmetric algorithms.

Asymmetric key algorithms, however, need to publish their public keys first by using a standard *Public Key Infrastructure* (PKI) [5] that involves issuing standard certificates called *X.509 Certificates* that contain the public key of the party in addition to signing party which is called *Certification Authority* (CA). Certification authority is a trusted organization that digitally signs certificates according to a standard process to ensure the identities of their sub-certificate owners [2]. Signing certificates is produced by computing the hash value of the certificate file using one of hash algorithms such as [6, 7] and later encrypting it using the private key of the CA. The resultant hash value is then appended to the certificate file. Clients could check this sign by computing the hash value again and decrypt the appended value of the sign using CA public key. The resultant two hash values are then compared to each other to ensure that the certificate in hand is the actual certificate signed by that CA which is considered trusted.

Accordingly, CA's should distribute their certificates to the Internet clients as X.509 Certificates. Moreover, CA's are formed as tree of parties signing each other except top parent nodes that are called the *Root CA*s. The certificates of the root CAs are self signed so that owner and signer are the same party. Any sub-CA can sign certificates if it has the authority to do so. Further, any certificate at any level should be verified through its super CA's root CA. Sup-CAs could be verified by the signature of their super CAs and so on. Notice that root CAs are not checked that way since they are self signed certificates that can be generated using any key generation tool such as OpenCA [http://www.openca.org]. Consequently, root CA's that are major part of the hierarchy are not verified in proper way since they don't have super CA's. That is why Root CA's in the real implementation in operating systems are pre-installed by the operating system or software itself. These Root CAs are verified and ensured by the vendor of the operating system. Users are not restricted to those preinstalled root CA's. In fact, users can install new certificates through a manual process which depends on the user to *allow* or *deny* the certificate to be installed on the system.

This manual process requires that users have a clear knowledge about the PKI Trust Model and what does certificate file mean. In practice, this certification hierarchy is not well-known to normal (naive) Internet users. Actually, users usually deny or allow certificates to be installed to their system influenced by the browser. For example, some very

assiduous users would deny any suspected contents because of the message shown by the system they use regarding new root CA certificate.

Blindly denying and allowing certificates to be installed are both improper because trust worthy root CAs that are new could be denied due to over assiduous user behavior. Similarly, an invalid root CA that fakes users could be installed due to user inattention. In this paper, a new distributed multi-agent approach is proposed to collaboratively evaluate certificates. This should help naïve users to install certificates according to the required level of security.

II. RELATED WORK

Cross Certificates are used to generate certificates between two root CA's to ensure the identity of each other [8]. This certification implementation is actually the same as normal trust model. Cross certification aims at signing CA certificate to new root CA through already trusted root CA. This approach is similar to creating sub CA from the older root CA because in both cases certificate is assured and verified according to *older* root CA certificate. Moreover, such certification mechanism makes the new root CA to be restricted by the rules of the older one as if it is a *sub CA* of the older one.

Another drawback is that if a *fake* and *none trusted* root CA certificate has been installed in the system of a given user, this root CA can allow more *none trusted* root CAs to be installed using cross certificates. The third drawback is that even if a root CA certificate has obtained a cross certificate from another trusted root CA, this doesn't ensure the validity of this new root CA in all systems. This is because of the different versions of the root CAs available in each system. Moreover, cross certification is not supported in SSL protocol so that an enhanced implementation is proposed in [9] to overcome this problem. This approach is used in very sensitive locations in order to restrict their users as to *who is trusted* as in [10].

Two mechanisms are proposed to check the status of certificates if they are revoked or not. These mechanisms are: (i) Online Certificate Status Protocol OCSP [11] and (ii) Certificate Revocation Lists (CRL) [15]. OCSP is a standard protocol that is used to check current status of certificate in order to check if it is revoked or not. In [12] an extension to this protocol is proposed to make some enhancements over Grid Security Infrastructures (GSI) [13]. This enhancement is proposed to increase the performance of the protocol in terms of speed. Another extension is proposed in [14] to support Grid CA cross certificates in the status determination process.

Certificate Revocation Lists, or CRLs, are periodic lists signed by CAs in order to publish revoked certificates [15]. This type of validation is done to decrease the load on CA servers that may result from OCSP requests. The main disadvantage of this method is the delay of revocation announcement because of the periodic announcements.

In order to get some optimal solution, OCSP and CRL are used together. This is done by classifying certificates according to their *importance* and *sensitivity*. This way, relatively important certificates are checked using OCSP. Less important certificates, however, are checked by CRLs [4].

In practice, OCSP and CRLs are usable in case of sub-certificates. However, in the case of root CA certificates, responses are signed by the same certificate that could be compromised and revoked. In this case OCSP responses and CRL files can be faked by compromising attacker [4].

Another protocol called Server-Based Certificate Validation Protocol (SCVP) [16] is released to enhance certificate validation process in case of missed CA certificates in the certification path. This protocol allows client to send validation requests to specialized server which is responsible of validating certificates.

Another enhancement is done in [17] to allow caching of expired or revoked certificates records. This enhancement is done to keep track of previous certificates to ensure none-repudiation of past CA certificates. This protocol is intended to provide help to clients by providing certification path validation service to them but it doesn't provide any mechanism on how to ensure that this new root CA is trusted or not.

In our proposed approach, expert users collaboratively participate in the process of validating root CAs. This should provide help to naïve or inexperienced users. Moreover, this proposal should help root CAs to publish their certificates according to collaborative human-based experience without any help of previously installed root CAs. In addition to the prevention of installation of faked root CAs certificates by determining the level of trust of new root CA.

III. THE PROPOSED CERTIFICATION MECHANISM

We propose using experienced users in order to evaluate root CAs using a distributed multi-agent approach. This is achieved by assigning real-experience-based trust-values scores to root CAs. Using these trust-values and based on predefined *automatic level of trust* (ALT) in the system, new root CAs can be automatically approved or rejected without user prompt. This ALT value is set according to the sensitivity of the work performed by the *current* client. For instance, if the client is a computer that is being used in a bank, a high ALT value is applied. Thus, only highly trusted root CA certificates are trusted in the system. If the client is a computer with less sensitive applications as home computers, low ALT value can be assigned.

Proposed algorithm involves using a user experience repository where users provide their feedback on new root CAs. We refer to this repository by the World Wide Trust Feedback Repository, or WWTFR for short. WWTFR is a centralized international repository that provides information regarding collaboratively evaluated certificates.

IV. THE PROPOSED APPROACH STEPS

Step 1: Expert users are registered with WWTFR.

Step 2: WWTFR assigns a dynamic impact factor (IF) to the account of each expert user. This IF is determined based on his/her experience.

Step 3: When a new root CA certificate is identified by a client, a request is sent to WWTFR in order to ask about that certificate. If the certificate is new to the WWTFR system, then WWTFR sends a *publish* message to each registered expert in order to collect their opinions. The evaluation outcome that is reported to the client is initially set to 0 as the certificate has never been evaluated before. Otherwise, if the certificate has already been reported and evaluated before, then the client is provided with the current trust-value of the certificate.

Step 4: WWTFR collects expert-users' opinions and computes average trust-value according to the impact factor of each expert-user.

Step 5: If the evaluation of the certificate is more than the ALT of trust for that client, then the system should be able to install the certificate automatically without prompting the user as it is considered trusted. Otherwise, the user is warned against installing the new root CA to his/her device. The final decision of allowing or disallowing the certificate to the system can be left to the current user.

A. The Impact Factors of Expert Users

Some users can be considered to be experts and their opinion or feedback can be considered to identify trust root CAs. For example, administrators of Proxy Servers are expected to be able to determine which certificate to trust. This is assumed due to the nature of their job. Such users are responsible of the security of the network located at the organizations with which they are affiliated. Consequently, they are expected to be aware and responsible of the protection of their network against installing non-trusted certificates. Such users are expected to choose to trust only very well known certificates to them.

In the proposed approach, an Impact Factor (IF) is calculated to each *registered* proxy administrator. The IF of a given user represents the *contribution* of the corresponding user to the overall root CA evaluation score. The IF is chosen such that it reflects the proficiency level of the user (who is an administrator of a proxy server). Another factor that decides the IF value is the size of *responsibility* he/she carries.

Based on the above discussion, the IF value of a given user depends on the following factors:

1) ***User's Proficiency***: This factor can be calculated or estimated by determining the percentage of tasks that he/she successfully accomplished on the proxy server that he/she administrates. Two major values affect this factor, namely; (i) the Average Number of Requests handled per day by the proxy server (ANR). And (ii) the Average Number of Successful requests per day (ANS) achieved. Based on that, we expect that as the ratio ANS/ANR increases, the user's proficiency increases.

2) ***The Size of the Organization***: relatively large organizations where the user (administrator) is working are expected to have more impact on the IF value of that user. Based on that, the IF value is expected to be directly proportional to size of the organization. One statistical measure can be used to represent the size of an organization. That is the Average Number of Clients the proxy of the organization handles per day (ANC). Given that the total number of clients in all the proxies is TNC, then the IF value is expected to be directly proportional to the ratio ANC/TNC.

3) **Type of organization** (TO): This factor reflects the sensitivity of the job done by the administrator with his/her organization. This is because organizations with relatively sensitive mission such as military agencies are most likely to be critical in evaluating root CAs than *normal* organizations such as educational organizations. Consequently, we propose that a coefficient to be to the formula that estimates the IF. This coefficient should reflect the sensitivity to security issues of the organization. This coefficient takes a value from the interval [0, 1].

4) ***The Activity of the Administrator***: This parameter represents the overall activity of the administrator in the evaluation process. *Active* administrators are trusted more than *inactive* ones. We propose to use (i) the *Number of Evaluated Certificates* (NEC) which is the number of certificates that have already been scanned and evaluated by the user, and (ii) the *Number of Explored Certificates* (NXC) is the number of certificates that the administrator have received and asked to evaluate them. In our approach, the IF is assumed to be directly proportional to the ratio NEC/NXC.

5) The ***Geographical Location of the Administrator Relative to Root CA*** (GL): This factor is used to give more weight to the evaluation of the root CA in hand that comes from the administrators that are co-located with that root CA. This is motivated by the fact that the administrator from a given geographical location are expected to be more capable of evaluating the root CAs from that location as they are expected to be more aware of it. In other words, organizations that are located in a given country or city have more authentic knowledge about the new root CAs belonging to that country or location. Again, this value is assigned a number between 0 and 1 according to the closeness of the new root CA and source of its evaluation. Notice that the CA location is a part of the standard certificate file [15].

Initial Impact factor is calculated as the following:

$$IF = ANS/ANR + ANC/TNC + TO + NEC/NXC + GL \quad (1)$$

This information are collected and maintained by the WWTFR system. Moreover, averages are calculated by the log of the proxy through an automated process frequently run on the WWTFR servers to keep the trust-values up-to-date.

B. Trust Level Computation

Root CA certificate trust level is calculated by taking the average IF values of all participated proxy servers in the trust process. Actions taken by a proxy administrator when facing a new root CA certificate may vary as follows:

1) The proxy administrator may add the certificate manually without requesting to update the WWTFR database. In this case certificate is assumed to be 100% trusted by that proxy admin.
2) The proxy administrator may add the certificate from the WWTFR update and trusts it manually without enquiring WWTFR for its trust-value. In this case certificate is assumed to be 100% trusted by that proxy.

3) The proxy administrator may check the current trust-value of the certificate before installing it. In this case certificate is assumed to be trusted by 50%.
4) The proxy administrator may install the certificate automatically from the WWTFR update according to a certain level of trust. In this case, certificate is assumed to be trusted with 0% by that proxy. This is because of blind process of trusting new certificate.
5) The proxy administrator may deny the installation of the new root CA certificate. In this case certificate is not trusted from that proxy with 0% percentage of trust.
6) The proxy administrator may deny the installation of the certificate due to automatic rejection mechanism. In this case, certificate is assumed to be none-trusted by 0%.
7) The proxy administrator may deny the installation of the new certificate after checking its evaluation. In this case certificate is assumed to be none-trusted by 50%.
8) Finally, the proxy administrator may manually deny the installation of the new certificate. In this case certificate is assumed to be none-trusted by 100%.

Assume that we have 10 proxy servers registered in WWTFR with IF values as presented in Table 1.

Table 1. Sample Proxies Impact Factors.

Proxy ID	Impact Factor (IF)
1	3.75
2	3.57
3	2.52
4	2.45
5	2.54
6	2.66
7	2.40
8	3.03
9	1.92
10	2.45

IF values should be computed using equation (1). The above numbers are assumed for demonstration purpose only.

Notice that greater IF value indicates greater impact of the corresponding proxy in the overall root CA certificate evaluation. Assume that certificate X is installed manually by the proxies 4, 5, 7, and 9 and is installed automatically in proxies 1, 2, and 3. Certificate is blindly rejected in 6 manually without considering its evaluation. Finally, assume that the administrator of proxies 8 and 10 did not do any evaluation to X.

The total IF of the certificate is calculated as follows:

4,5,7,9 IF values are computed by multiplying their corresponding IF values by 1 due to the total trust of these parties. So total trust IF is:

2.45 X 1 + 2.54 X 1 + 2.40 X 1 + 1.92 X 1 = 9.31

1, 2, and 3 trust values are neglected from the average because they have blindly trusted X. Proxy 6 IF value must be multiplied by 1 due to the manual rejection of the certificate. Consequently, proxy's 6 share to the Total IF is 2.66 X 1 = 2.66. Proxy 8 and 10 IF values should be multiplied by 0 as they did not provide any feedback regarding X.

Finally, ***Total IF*** = (9.31 - 2.66 + 0)/(2.45 + 2.54 + 2.66 + 2.40 + 3.03 + 1.92 + 2.45) = 6.65 / 17.45 = 0.38 = 38%

This result indicates that this certificate is trusted with a level of 38%. If the number is negative, this certificate is assumed to be none-trusted.

Including Certificate in Proxy Update

There are two cases in which certificates are assumed to be included for evaluation. These cases are: (i) One of the participating proxies has manually installed the certificate in the proxy server. (ii) Significant number of clients has explored the certificate while navigating through the internet.

In the first case, the proxy administrator has accepted the certificate, thus, this certificate is considered a trusted root CA. In this case, root CA certificate is initially assigned a trust level that is equal in value to the impact factor of the proxy server that accepted it as in Figure 1.

In the second case, if a client has explored this root CA certificate while navigating through the internet, then certificate id is sent to the WWTFR server in order to check for its evaluation. If the certificate is new, server asks the client to upload certificate file and store it and initiate its counter to 1. If another client has explored the same certificate again, it sends certificate id for evaluation and certificate counter is incremented by 1. If certificate counter reaches a predefined WWTFR threshold value, the certificate is assumed to need an evaluation. In this case the initial impact factor of the certificate is set to 0. clients that send initial request of evaluation are replied with 0% trust level which represents that this certificate is unknown certificate and that WWTFR cannot yet decide whether this particular certificate is trusted or not as in Figure 1.

C. Request for Evaluation

Request for evaluation is represented to proxy administrator as a new certificate update and ask him to evaluate this certificate and install it to his proxy store. This is done by an automatic service that downloads periodic updates of new root CA certificates from the WWTFR server as in Figure 2.

D. Security

Client communications with WWTFR server are handled through HTTPS protocol in order to prevent attacks or faked response by attackers. Moreover, HTTPS protocol can navigate through proxies within which the client resides. This involves installing WWTFR certificate in every client in order to be able to communicate with. This process can be done by including WWTFR server certificate in systems pre-installed root CA certificates.

Proxy servers are issued certificates from WWTFR in order to ensure their identities. Moreover, proxies ensure the identity of WWTFR server through its self-signed certificate. This scenario can be implemented using the standard SSL protocol by activating certificates handshaking step which is used to ensure the identity of both sides. This is required to deny faked evaluation responses and identity spoofing attacks for both WWTFR and proxy server. Proxy server certificate is generated and signed in the initial proxy registration step.

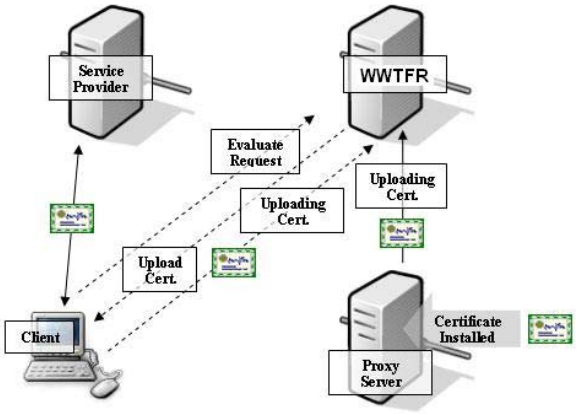

Figure 1: Updating WWTFR with new certificates.

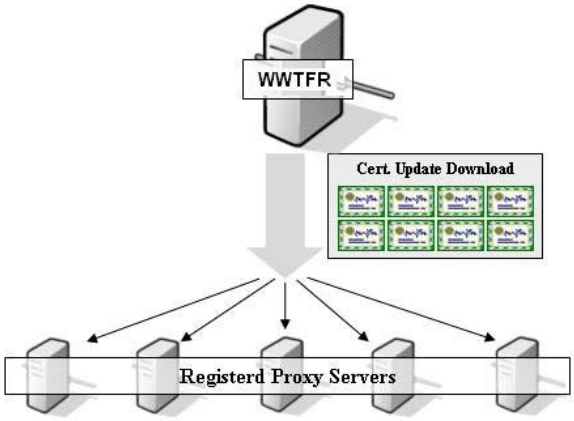

Figure 2: Downloading new certificates by proxy servers for evaluation.

E. Proxy Server Statistics

Proxy server statistics are sent using scheduled task that collects data and sends them to the WWTFR server in order to maintain an up-to-date impact factor of proxy server. These data is sent, as mentioned before, using the normal SSL protocol with handshaking capability.

F. Implementation Modules

The implementation of the proposed approach considers the implementations of three modules as follows:

1) **The Client Module**: This module is included in the client software in order to check new certificates by communicating with WWTFR server and automatically install or reject the installation of the certificate in client store.
2) **The Proxy Module**: This module is included in the proxy in order to periodically check new root CA certificates updates from the WWTFR server. This module also handles administrator requests and feedbacks to the WWTFR server.
3) **The WWTFR server module**: This module is the coordinator module that is responsible of handling requests from clients and proxies in addition to administrators' feedbacks.

H. Analysis

Our proposal is distributed in nature. Computing its complexity should consider one main factor; that is computing complexity in terms of the maximum number of messages exchanged in order to complete any process.

For this purpose, processes are classified as the followings:

1) ***Evaluation Query of New root CA certificate explored by normal client***. In this case three messages are exchanged. These messages are (i) "*Request to Evaluate Root CA certificate"* message, "*Send Certificate Response from WWTFR"* message, and the "*Root CA Certificate Upload"* message. In this case no WWTFR evaluation response is sent to the client because that client considers its evaluation as 0 because of the "*upload certificate response"* message sent by WWTFR server to it as in Figure 3.

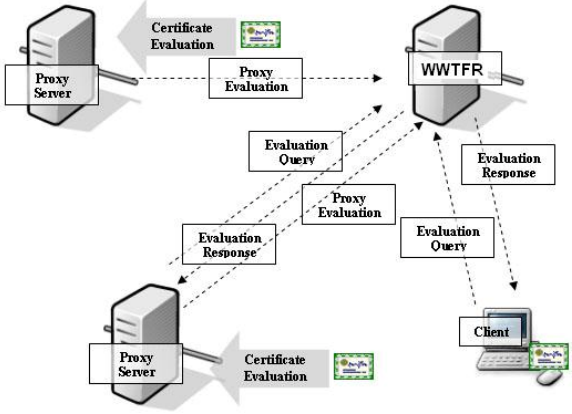

Figure 3: Evaluation process mechanism

2) ***Certificate Evaluation Process***: in this case a maximum 3****n*** messages are exchanged in order to complete certificate evaluation process. Where ***n*** is the number of registered proxies in the WWTFR server. This is because of our assumption that each proxy server queries certificate evaluation before that proxy's contribution of evaluation. This means that there are two messages to be exchanged which are "*request for evaluation"* and "*the WWTFR response"*. The third message is the evaluation message sent by proxy to WWTFR telling of its evaluation decision.
3) ***Manual installation of new root CA in proxy server***. In this case the proxy admin installs a certificate in the proxy server and an "*automatic upload of certificate"* message is sent to the WWTFR server.
4) ***Broadcast for certificate evaluation***. This is the process of broadcasting new root CA certificates to registered proxies in order to check their evaluation. This process is important because new root CA certificates are broadcasted in a periodic basis. In this case, in the worst case every new root CA certificate is broadcasted alone in a single update, which means that 2****m*** messages are sent to broadcast each new root CA certificate. Where ***m*** is the number of registered proxies in the WWTFR server. An extra message representing proxy server

request for update is sent by proxy server in order to download new root CA certificates.

5) ***Periodic (probably Daily) proxy statistics update***. In this case, statistics are sent to the WWTFR server using a single update message. So, the total number of messages exchanged is ***m***. Where ***m*** is the number of registered proxy servers in the WWTFR server.

The proposed mechanism is sensitive and must be protected from different types of attacks:

1) ***Identity spoofing attacks***: An attacker can spoof the identity of the WWTFR server in order to send faked evaluation results to clients or proxies. This attack can be prevented using digital signature that is used in HTTPS and SSL protocols used to communicate with the WWTFR server. This digital signature is used by client to check the identity of the sender in addition to the data integrity of the response. This signature is attached to the proxy evaluation or the query evaluation message sent by the proxy server in order to deny any identity spoofing of the proxy server.

2) ***WWTFR private key compromise***. The Private Key of WWTFR is very sensitive factor in the whole process. So, it must be properly secured. If it is compromised by any party, attackers can fake any response or evaluation and propagates illegal certificates. In order to prevent such situation, OCSP and revocation lists are enabled to allow clients and proxies to check the status of WWTFR certificate.

3) ***Denial of Service (DoS) Attack***. Attackers can overload the WWTFR server in order to deny its services to clients and proxies. This attack doesn't affect certificate evaluation results but it hangs the overall system as if the WWTFR server is down. In this case, several approaches can be applied such as election algorithms in addition to DoS prevention or avoidance algorithms.

V. RESULTS & CONCLUSIONS

The proposed approach is intended to provide intelligent and secure web service that provides a *trust-value* to relatively new root CAs certificates. As described before, this approach utilizes the knowledge of *expert users* to collaboratively evaluate new root CAs. Our proposal is secure against illegal access or spoofing through the usage of standard digital signature.

As a future work to our proposal, we are working on improving it to be more secure against DoS attackers. Another future research direction is to motivate expert users to provide their feedback to evaluate root CAs. One possible solution is to motivate organizations by providing such feedback as a requirement for gaining standard certificates such as the ISO.

REFERENCES

[1]. Blake-Wilson, S., M. Nystrom, D. Hopwood, J. Mikkelsen, T. Wright (2006). “Transport Layer Security (TLS) Extensions”, The Internet Engineering Task Force (IETF) RFC4366, April 2006.

[2]. Rescorla, E. (2000). HTTP Over TLS, The Internet Engineering Task Force (IETF) RFC2818, May 2000.

[3]. Rivest, R, A. Shamir, L. Adleman. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM 1978; 21 (2): 120–126.

[4]. Lyons-Burke, K., (2000). "Computer Security", National Institute of Standards and Technology (NIST), 2000.

[5]. Housley, R., W. Ford, W. Polk, D. Solo. (1999). “Internet X.509 Public Key Infrastructure Certificate and CRL Profile”, The Internet Engineering Task Force (IETF) RFC2459, January 1999.

[6]. Rivest, R.. (1992). “The MD5 Message-Digest Algorithm”, The Internet Engineering Task Force (IETF) RFC1321 April 1992.

[7]. Eastlake, D., P. Jones. (2001). US Secure Hash Algorithm 1 (SHA1), The Internet Engineering Task Force (IETF) RFC3174, September 2001.

[8]. Adams, C., S. Farrell. (1999). “Internet X.509 Public Key Infrastructure Certificate Management Protocols”, The Internet Engineering Task Force (IETF) RFC2510, March 1999.

[9]. “PKE Cross-Certificate Chaining Issue”, USA Department of Defense, May 2010.

[10]. Kaji, T., T. Fujishiro, S. Tezuka. (2008). “A Proposal of TLS Implementation for Cross Certification Model”. EICE - Transactions on Information and Systems 2008; E91-D Issue 5: 1311-1318.

[11]. Myers, M., R. Ankney, A. Malpani, S. Galperin, C. Adams. (1999). “X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP”, The Internet Engineering Task Force (IETF) RFC2560, June 1999.

[12]. Zhang, S., B. Wang. (2008). “Research on an extended OCSP protocol for grid”. Proceedings of 7th World Congress on Intelligent Control and Automation (WCICA 2008). Chongqing, China, 2008.

[13]. Tuecke, S., V. Welch, D. Engert, L. Pearlman, M. Thompson. (2004). “Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile”, The Internet Engineering Task Force (IETF) RFC3820, June 2004.

[14]. Zhang, S., H. Gong, B. Wang. (2006). “An Extended OCSP Protocol for Grid CA Cross-certification”. Proceedings of Second International Conference on Semantics, Knowledge and Grid (SKG '06). Guilin, Guangxi, China, 2006.

[15]. Cooper, D., S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk. (2008). “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile”, The Internet Engineering Task Force (IETF) RFC5280, May 2008.

[16]. Freeman, T., R. Housley, A. Malpani, D. Cooper, W. Polk. (2007). “Server-Based Certificate Validation Protocol (SCVP)”, The Internet Engineering Task Force (IETF) RFC5055, December 2007.

[17]. Wallace, C. (2008). “Using the Server-Based Certificate Validation Protocol (SCVP) to Convey Long-Term Evidence Records”, The Internet Engineering Task Force (IETF) RFC5276, August 2008.