Design of Graded Trusts by Using Dynamic Path Validation

Akira Kubo, Hiroyuki Sato

To cite this version:

Akira Kubo, Hiroyuki Sato. Design of Graded Trusts by Using Dynamic Path Validation. 4th IFIP WG 11.11 International on Trust Management (TM), Jun 2010, Morioka, Japan. pp.172-183, 10.1007/978-3-642-13446-3_12. hal-01061326

HAL Id: hal-01061326
https://hal.inria.fr/hal-01061326

Submitted on 24 Nov 2017

Design of Graded Trusts by Using Dynamic Path Validation

KUBO Akira[1] and SATO Hiroyuki[2]

[1] symmetriccipher@gmail.com

[2] Information Technology Center,
The University of Tokyo, Japan.
schuko@satolab.itc.u-tokyo.ac.jp

Abstract. In modern information service architectures, security is one of the most critical criteria. Almost every standard on information security is concerned with internal control of an organization, and particularly with authentication. If an RP (relying party) has valuable information assets, and requires a high level to authentication for accepting access to the valuable assets, then a strong mechanism is required. Here, we focus on a trust model of certificate authentication. Conventionally, a trust model of certificates is defined as a validation of chains of certificates. However, today, this trust model does not function well because of complexity of paths and of requirement of security levels. In this paper, we propose "dynamic path validation," together with another trust model of PKI for controlling this situation. First, we propose Policy Authority. Policy Authority assigns a level of compliance (LoC) to CAs in its domain. LoC is evaluated in terms of a common criteria of Policy Authority. Moreover, it controls the path building with considerations of LoC. Therefore, we can flexibly evaluate levels of CP/CPS's in one server. In a typical bridge model, we need as many bridge CAs as the number of required levels of CP/CPS's. In our framework, instead, we can do the same task in a single server, by which we can save the cost of maintaining lists of trust anchors of multiple levels.

1 Introduction

In modern information service architectures, security is one of the most critical criteria. Today, security is discussed in terms of computer security, network security, and information security. It is not long before information security is considered to be important. Information security is concerned with controls of behaviors of systems and humans for protecting information assets. As one of major differences of information security to others, we must consider organizations as major players of security. "Internal control" is discussed organization-wise. Security policies are also organization-wise defined and published.

Information security is closely related to the concept of information assets. An organization recognizes that information itself has and produces value, whenever it experiences information theft, leakage, and insider trading. Moreover, information security is related to legal issues such as privacy. Privacy related information is often accumulated in an organization. In such a situation, the organization is required by law to protect such information that is considered to be “owned” by individuals, not by the organization. Thus, protection of information assets is demanded. By controlling systems and humans that handle information assets, information security gives some guarantee to such protection.

There are defined several standards on information security. For example, ISMS, or ISO 27001, is commonly used as a criteria of system and information security. Actually, almost every standard such as ISMS is concerned with internal control of an organization. Among several issues of internal control, authentication is the most critical. Allowing access of critical information assets is guaranteed by how assured the used authentication is. A Level of assurance associated with an authentication differs in its mechanism. For example, certificate authentication certainly provides higher level mechanism than password authentication. Even in certificate authentication, its strength differs in CPs of certificates.

If an RP (relying party) has valuable information assets, and requires a high level to authentication for accepting access of the valuable assets, then a strong mechanism is required. Although it is of course that the strength of authentication is brought by its mechanism, initial setups and lifecycle management of credentials and IDs must also be considered. In certificate authentication, they are defined and published in CP/CPS’s of certificates. Matching of the strength of IdP (ID providers) and requirements by RPs are the source of trust in a federation.

In this paper, we focus on a trust model of certificate authentication. Conventionally, a trust model of certificates is defined as a chain of certificates. A certificate chain is constructed so that an issuing CA is endorsed, or given its digital signature by another CA. If the anchor of the chain is contained in a list of trusted CAs, then the target CA of the chain can be trusted. These chains are central in constructing the trust of PKI.

However, today, this trust model does not function very well. Its reasons are classified in twofold: one is that there can be constructed an arbitrary complex chain, in which chains are hard to control. Although there are defined three trust models, hierarchical, mutual, and bridge for taming this complexity, they are only partially implemented to validate complex chains. The other reason is more critical: because there are provided several levels of certificate policies, CAs of the same levels are fragmented into small groups. This means that we need as many as CAs as levels, which proliferates the number of CAs. Actually, major commercial PKI vendors operate as many CAs of different assurance as required even for the same usage such as client authentication. Although the difference of levels can be inferred by checking CP/CPS’s, it must manually be done. This will cause a long negotiation in building a bridge CA. To control such fragmentation is strongly required.

This paper proposes dynamic path validation, together with another trust model of PKI for controlling this situation. First, we propose a policy management server. This server assigns a level to CP/CPS of a given CA. The assignment may mutually be done in an agreement of the policy management server with the CA. Or, some criteria approved by a group of CAs may be used. Second, we propose an extended path validation based on the levels provided by the policy evaluation server. In the path construction, levels are used together with certificate chains. The consistency of levels is also discussed.

Our framework assumes one policy management server, which plays as a pivot among policies of CAs. Instead of mutually agreeing or fighting on CP/CPS of a bridge CA, this policy management server accepts multiple levels of securities of CAs. Therefore, we can flexibly evaluate levels of CP/CPS's in one server. In a typical bridge model, we need as many bridge CAs as the number of required levels of CP/CPS's. In our framework, instead, we can do the same task in a single server.

The rest of this paper is organized as: Section 2 studies scenarios in which efficiently handling multiple levels is important. Section 3 proposes dynamic path validation as our solution. Policy Authority is introduced. Furthermore, path validation is extended in the way that levels of CP/CPS's are reflected. Section 4 surveys related work. Section 5 summarizes this paper.

2 Stratified Paths Depending on LoA

Recently, many of critical services have been implemented as Web applications. Accordingly, there are many services of various levels of significance. Today, even in a single organization, there are provided many services that have various levels of significance. The significance is evaluated in the information assets handled by the service. For example, if a service handles privacy, it must be treated with care. If a service handles medical information, it must be treated with the highest security.

Authentication is a key mechanism that implements the levels of significance. Its idea is to control the access by identifying end users with how assured the authentication can be. Generally, they are called *"level of assurance (LoA)."* There are defined some standards of LoAs such as NIST 800-63[4] for evaluating the levels. In such situations, an SP(service provider) requires an appropriate LoA to an IdP(ID provider) for accessing its information assets. In a fixed trust circle, it is common that its member IdPs and SPs are under some agreement as for keeping LoA of IdPs, which actually gives trust in the circle.

Among various authentication mechanisms, certificate authentication is usually given the highest LoA. Certificate authentication includes a process of path validation: a path between two CAs is constructed if a CA trusts the other CA. If the root of the path constructed in the validation process is in the domain of trusted CAs, then the validation, and therefore authentication succeeds. Thus, the trust of certificates is reduced to path construction whose root is trusted.

Trust brought by path validation causes a problem: the “LoA” of the trusted domains. In general, CAs are operated in various levels of CP/CPS. Some CP/CPS can be stricter than others. More strictly operated CAs can provide a higher LoA to SPs. SPs that require a higher LoA can trust only strictly operated CAs. A problem is that high level requirements result in inconvenience to users and high cost in operations. Today, a solution to such trade-offs is given in such a way that an organization operates CAs of multiple roots that correspond to multiple levels of operations. Looser certificates are used to access less important information assets, but with less cost than strictly issued certificates.

Typical examples of multiple roots can be seen in server certificates. Today, most major browsers classify server certificates as three: EV certificates[5], web trust[3], and others. In the path validation of server certificates, the three never intersect. Even in client certificates, major vendors such as Verisign provide multiple roots of different levels of trust. In a complex organization, the situation is very similar to the real world. There are many organizational units that have various levels of independence. There are many services that require various LoAs. The result would be many CAs of various LoAs to cope with the various requirements of services.

This kind of scenarios causes fragmentation and maintenance problems. In the real world, many CAs are established to provide required LoAs. In such a situation, the cost of maintenance of trust domain is high. The domain is fragmented according to LoAs. Moreover, if a requirement level of an SP changes, the list of trusted CAs must be accordingly modified, causing a problem in maintenance. Even if no change occurs to an SP, because the world of CAs continuously changes, the maintenance is still a problem.

3 Dynamic Path Validation

In this paper, we propose *“dynamic path validation.”* to tame the fragmentation and maintenance problems stated above. Dynamic path validation is a kind of delegated path validation in which Policy Authority plays as a key component.

3.1 Architecture of Dynamic Path Validation

In our scenario, there are three players: end entities, RP(or SP), and Policy Authority. An end entity requests authentication for a service with his/her certificates. An RP (Replying Party) or an SP (Service Provider) is a server that authenticates a user. In this scenario, an RP requires that a certificate of an end entity has a certain level of assurance. The RP delegates the path validation to Policy Authority. The Policy Authority checks whether a path provided by an end entity is valid. Policy Authority dynamically builds the path by using levels of CP/CPS’s. Specifically, it checks whether requirements of an RP given as a level of certificates is satisfied in the certificate chain. In other words, this framework checks conditions of path validation in a way not related to information statically embedded in certificates. This is the name of “dynamic” path validation We illustrate our architecture in Fig. 1.

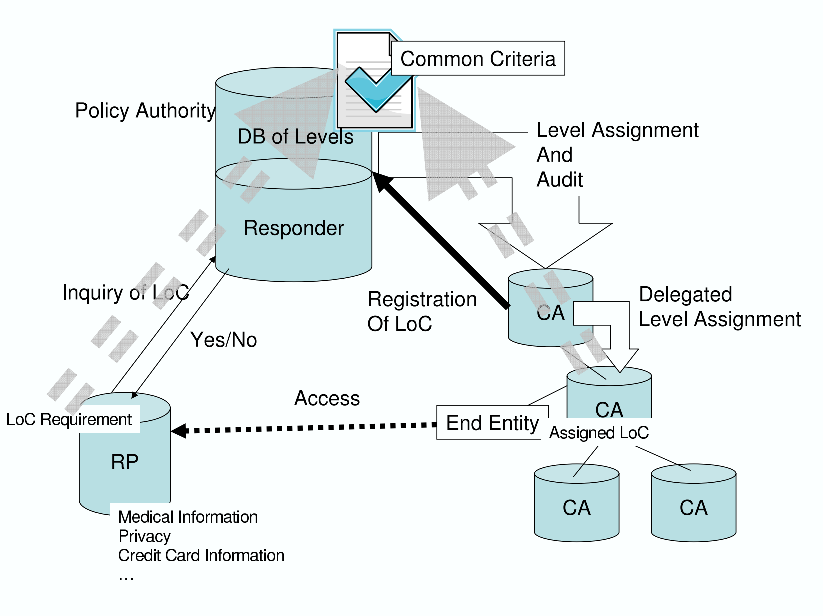

Fig. 1. Architecture of Dynamic Path Validation

Policy Authority The key component in our framework is Policy Authority. The functions provided by Policy Authority are:

1. to decide and publish the common criteria of CP/CPS,
2. to register CAs that comply with the published criteria of CP/CPS, and
3. to validate paths on behalf of RPs.

In Fig. 1, 1. corresponds to "Common Criteria," 2. to DB of Levels, and 3. to Responder, respectively.

Policy Authority must be operated under an agreement with participating CAs. It assumes that participating CAs agree on some predefined criteria. This at least includes those on audit and delegation of assignment of levels to subordinate CAs. Today, audit is considered to be a standard way to assure the quality of operations. Therefore, we demand audit to assure the compliance with the criteria. Moreover, delegation must be operated in an appropriate way.

In general, delegation is one of major solutions of distributed system management in the case that specific tasks are hard to control or to maintain. In this case, path validation is a heavy task, and hard to maintain in a single client.

In this way, with Policy Authority as the core, CAs and RPs participate in the circle, which simulates the circle of trust in Liberty-like federations. The difference is that in the latter (Liberty), IdPs and RPs mutually evaluate the quality of their services, while in the former (ours), they refer to the criteria via

Policy Authority. In this meaning, Policy Authority plays as a pivot in the circle. We illustrate our concept of circle in Fig. 2. Actually, building circles of trust is one of key issues in federations. by Policy Authority acting as the pivot of the circle, we can save cost of building multiple circles. This scenario resembles putting bridge CAs as a pivot in path building.

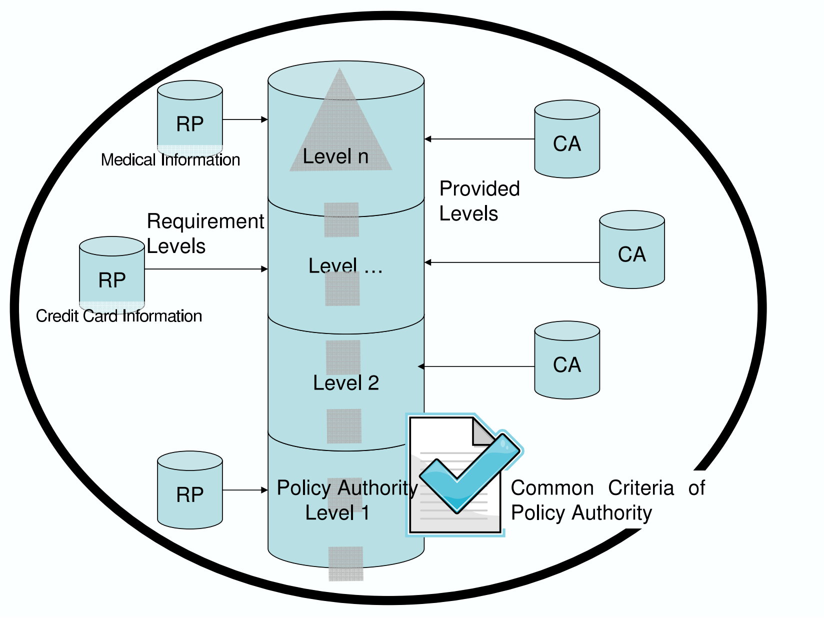

Fig. 2. Circle of Trust consisting of servers of various LoCs as Policy Authority as the Pivot

RP and CA An RP delegates path validation to Policy Authority. In the delegation, maintaining the list of trust anchors is a task of the delegated server. In our framework, an RP maintains its requirement to levels of certificates. Policy Authority, or the delegated server receives the requirement together with a path, then validates it. This means that it can house multiple path validation methods in one server. Policy Authority controls path validation by using required levels together with trust anchors (i.e. registered CAs) as illustrated in Fig. 3

We see that in the figure, instead of having as many bridge CAs as the number of levels, we can house multiple levels of path building in one Policy Authority.

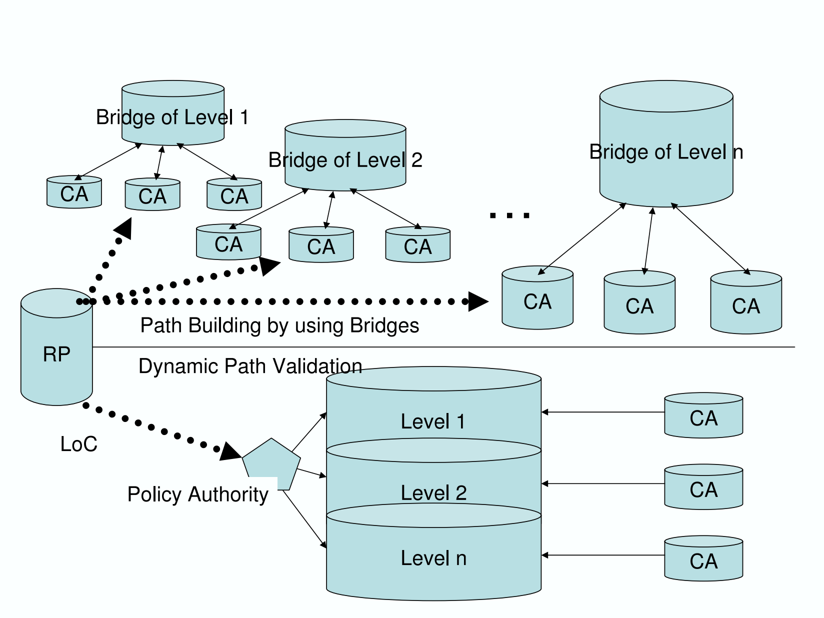

Fig. 3. Housing Multiple Path Validations in One Policy Authority

3.2 CP Certification

The relation that a CA trusts another CA is determined by some kind of evaluation of CP of the target CA. The evaluation must be based on a common criteria such as EV, Web Trust and RFC 3647.

Levels of Compliance of Certificates Conventionally, a level of assurance is given to a CA according to a specific criteria and certification based on the criteria and related audit. Typical criteria include WTCA [3], EV [5], and "Specified Certification Business" [22] in Japan. All of these criteria require audit to assure the quality of CA operations. However, the audit is done for checking compliance with CP/CPS's of given CAs. Compliance of a CP/CPS with a given criteria must be proved as another process.

In this paper, we define *"level of compliance (LoC)."*

Definition: We define a level of compliance (LoC) for a criteria as a numeric value that represents how strictly a server is operated in compliance with the given criteria.

As a criteria of LoC, we have some standard templates for CP/CPS's such as EV, WTCA, and RFC3647. Conventionally, because they define the minimum set of requirements, compliance levels are just 0/1 (yes or no). In our extended compliance, we may define optional criteria that enhances the level of security in the same template. Therefore, we have more than two levels as for compliance: enhanced compliance/minimum compliance/no. In this meaning, LoC can be considered as an extension of conventional certification.

Moreover, LoC can be defined as an extension of LoA. Usually, certificate authentication is given the highest level. In LoC, the level of operations of CA is in concern, which is the same framework as the assignment of LoA in which the level of ID providers is in concern.

In evaluating LoC, operations of a given CA are audited for a given criteria. In LoC, audit is done not only for CP/CPS, but also for a predefined criteria. The evaluation must be done by an authority for the criteria. Here, Policy Authority plays as an auditor.

Assigned Levels and Derived Levels In our framework, in addition to Policy Authority, which assigns a level to a CA, a CA can assign derived levels to its subordinate CAs. This delegation is essential in saving cost of Policy Authority operations. It is a fundamental assumption that a CA must control its subordinate CAs in path validation. This means that a level of a given CA is inherited to its subordinate CAs. In this paper, this control is extended to derived level assignment. This assignment is done under the restriction that the derived level must not be greater than the level of the parent CA.

Evaluation Axis of Criteria In NIST standard[4], we see four axis for evaluating authentications. In this paper, we borrow related two axes of evaluation: levels of initial identification (more generally, ID lifecycle management in [17, 18]), and levels of tokens. For example, Verisign defines three levels as for the levels of assurance depending on the methods of initial identification and the coverage of assurance. In this meaning, our proposal is already implemented in the real world.

Moreover, a CA must be operated under a certain security constraint. RFC 3647 also defines security constraint in operations. [17] proposes criteria for both IdPs and SPs. Here, we propose our evaluation axes of criteria in terms of [17]:

1. ID lifecycle management,
2. levels of tokens
3. Quality of management of the server:
   (a) Management of access control
   (b) Control of physical security.
   (c) Management of privileges in operation

In addition to these axes, we require audit as the mechanism that guarantees the quality in terms of published criteria. To control the quality of operations, audit is considered to be very effective. It is mandatory that Policy Authority audits participating CAs.

3.3 Extended Path Validation by using Dynamic Path Validation

Now we have two components: LoC and Policy Authority. We extend path validation so that a CA of a lower LoC can trust a CA of a higher LoC, even if there is no path between the two in conventional meaning. We call this extension as *"dynamic path validation (DyPV)."*

Our DyPV is processed as follows: first, all CAs registered at the given Policy Authority are considered to be in its domain. In other words, a CA in the domain is given an LoC under the common criteria. Second, the path validation is extended by using levels: if a certificate issued by CA_1 is presented at an RP that requires n_2 as LoC, and n_1 is given to CA_1, then the certificate is validated if $n_1 \geq n_2$. We extend this validation to a general certificate chain. If a path is built whose root is CA_1, and $\mathrm{CA}_1 \cdots \mathrm{CA}_n$, are registered, then we compare their LoCs with the required LoC.

The algorithm of DyPV is given in Fig. 4. A validating RP delegates the validation to Policy Authority. Policy Authority responds with true/false depending on whether DyPV succeeds or not. The given inputs are `CC[]`, a certificate chain given by a validatee, and `LoC`, a required LoC given as a policy of the validating RP. This algorithm partially extends `CC[]` so that the root of the path is in the domain. Here, Policy Authority builds a path in the conventional way so that its root is in the domain. Then, Policy Authority compares the required `LoC` with LoCs in the domain. In other words, registered CAs play as trust anchors in a conventional sense. Instead of maintaining the list of trust anchors, an RP just makes an inquiry of LoC as its requirement, and Policy Authority returns yes/no to the inquiry.

A problem arises in the algorithm: the path extension. There can be a case that there are two or more possibilities of extension, and in one extension, the extended validation succeeds, and in another extension, it fails. Our algorithm requires that the extension must be done so that the extended certificate chain $DD[]$ satisfies the condition $DD[1] \geq LoC$. By this restriction, we can eliminate false cases. If validation fails in a path extended in this way, we restart the path building.

If we can validate a path, then we must guarantee that a validation of any extension of the path also succeeds. Therefore, we require that if there is a path from $\mathrm{CA}_1 \longrightarrow CA_2$, meaning that CA_1 issues a certificate to CA_2, then their LoCs must satisfy $\mathrm{L}oC(CA_1) \geq \mathrm{L}oC(CA_2)$. This consistency must be maintained by Policy Authority.

3.4 Comparison with RFC 5280

Conventionally, path validation is defined as RFC 5280 [8]. In RFC5280, there is defined control of path building via policy extension fields in certificates. In our framework, for representing policies of CAs, we use LoC under a common criteria of Policy Authority. Our idea is that lifecycles of CAs and of their policies are not the same. Policies and operations can continuously be enhanced even in the same CA. We separate the two lifecycles, and manage policies by using Policy Authority on-line.

```
Policy Authority:

Boolean validate(cert chains CC[], int LoC)
{
start:
 if (CC[1] is in the domain of Policy Authority) {
   DD[] = CC[]; // guarantees an LoC is assigned to DD[1].
 } else {
  if (CC[1] can be extended by using information in Policy Authority) {
     select a chain CC1[] such that LoC(CC1[1]) >= LoC;
     DD = CC1 + CC;  //Extend CC[] with CC1[];
  } else
     return false;
 }

 validate DD[]; // RFC 5280 compliant path validation

 for (C = tail of DD[]; C != DD[1]; C = parent(C)) {
   if (C is in the domain of Policy Authority) {
     if (LoC(C) < LoC) goto start;
     // Validation fails for DD[]. Reset and Restart.
     if (LoC(C) is undefined) continue;
     // if undefined, LoC of C inherits its parent's.
   } else {
     continue;
   }
 }
 // check if all of LoC's of certificates in CC are
 // higher than the requirement.
 return true; // validated.
}
```

Fig. 4. Algorithm of Dynamic Path Validation

3.5 Control of Subordinate CAs

In our framework, Policy Authority allows registered CAs to assign levels to subordinate CAs. This delegation is a key to save the cost of operations of Policy Authority. Assignment of LoC assumes that subordinate CAs have lower or equal LoCs than those of their parents. Policy Authority must enforce this restriction on every participating CAs Audit must also be effective for this enforcement.

4 Related Work

Path building[7], and validation[8] have been central issues in PKI domain extensions. There have been proposed three major methods of path construction models: hierarchical, mutual, and bridge models. Although the bridge model has been considered to scale, and has been implemented on some major domains, there are found some problems other than technical ones to hinder its growth. Furthermore, delegation of tasks related to them is studied because they are too heavy for general RPs. The discussions are summarized as RFC 3379[16]. OCSP[11] and SCVP[9] are also classified as protocols partly delegating validation. Our framework is also classified as delegation. Ours considers LoCs in path validation.

It is commonly understood that operations of CAs can differ in their CPs. They include usage, profile, and security. There are some standard templates of CPs such as RFC 3647[6], and PKI lite[21].

Evaluating IdPs and assigning specific LoA is required by some security-sensitive SPs[2]. There have been proposed several systems that use LoA. As major federated identity systems, both Liberty and OpenID provide a mechanism of sending LoA of IdPs to SPs [13, 15].

Moreover, in Grid, there are established policy management authorities[23] to enforce the policies of Grid on participants.

Although discussions of LoA [12] have been limited to ID and authentication, they are very fruitful in assuring security level in building federations. In particular, they are essential in the framework that ID information is provided to an SP by IdPs in multiple organizations via SSO. OMB guidance[14] and NIST standard[4] are milestones in the discussion. They are also the driving force to define LoA to large federations. Today, LoA is widely discussed in many organizations, grids, federations [10], and inter-federations [1].

LoA can be generalized to SPs. [17, 18] propose a consistent assignment of LoA to SPs in terms of security policies of organizations.

Note that all of these must be done as a part of risk management. [19, 20] discuss authentication in terms of risk management.

5 Concluding Remarks

In this paper, we have proposed dynamic path validation (DyPV). In DyPV, CAs in a domain are registered in Policy Authority, which plays as a pivot. Moreover, according to a common criteria, LoC is assigned to each CA. In this way, Policy Authority houses multiple levels of compliance in one server. Furthermore, path validation has been extended so that an LoC, or a level of CP/CPS's is reflected.

Our framework uses LoC instead of a list of trust anchors. CAs are not required to issue unnecessary certificates for path building, but Policy Authority checks whether the validation in terms of LoC requirement succeeds. Operations under a common criteria of Policy Authority is easier than maintaining lists of trust anchors of multiple levels in multiple bridges.

References

1. Alterman, P.: Interfederation Initiatives for Identity Authentication. Federal Demonstration Partnership, January meeting (2008)
2. Alterman, P., Keltner, J., Morgan, R.: InCommon Federation: Progress, Partnerships, Opportunities. Internet2 2007 Fall Meeting (2007)
3. American Institute of Certified Public Accountants and Canadian Institute of Chartered Accountants: Trust Services Principles, Criteria and Illustrations for Security, Availability, Processing Integrity, Confidentiality, and Privacy (2006)
4. Burr, W., Dodson, W., Polk, W.: Electronic Authentication Guidelines. NIST SP800-63 (2006)
5. CA/Browser Forum: Guidelines for the Issuance and Management of Extended Validation Certificates. (2007)
6. Chokbani, S., Ford, W., Sabett, R., Merrill, C., Wu, S.: Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework. RFC 3647 (2003)
7. Cooper, M., Dzambasow, Y., Joseph, S., Nicholas, R., Internet X.509 Public Key Infrastructure: Certification Path Building. RFC 4158 (2005)
8. Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., Polk, W.: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280 (2008)
9. Freeman, T., Housley, R., Malpani, A., Cooper, D., Polk, W.: Server-Based Certificate Validation Protocol. RFC 5055 (2007)
10. InCommon Federation: Identity Assurance Profiles Bronze and Silver. http://www.incommonfederation.org/docs/assurance/InC_Bronze-Silver_IAP_1.0_Final.pdf (2008)
11. Myers, M., Ankney, R., Malpani, A., Galperin, S., Adams, C.: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. RFC 2560 (1999)
12. Nedanic, A., Zhang, N., Yao, L., Morrow, T.: Levels of Authentication Assurance: an Investigation. Proc. 3rd Int'l Symposium on Information Assurance and Security, 155–158 (2007)
13. OASIS: Level of Assurance Authentication Context Profiles for SAML 2.0. (2009)
14. Office of Management and Budget (U.S.): E-Authentication Guidance for Federal Agencies. M-04-04 (2003)
15. OpenID: OpenID Provider Authentication Policy Extension 1.0 (2008)
16. Pinkas, D., Housley, R.: Delegated Path Validation and Delegated Path Discovery Protocol Requirements. RFC 3379 (2002)
17. Sato, H.: A Service Framework based on Grades of IdPs and SPs. Proc. Securiy and Management 2009, 379–385 (2009)

18. Sato, H.: $N \pm \varepsilon$: Reflecting Local Risk Assessment in LoA. Information Secruity 2009 (OTM 2009, LNCS 5871), 833–847 (2009)
19. Stoneburner, G., Goguen, A., Feringa, A.: Risk Management Guide for Information Technology Systems. NIST 800-30 (2002)
20. Yan, J., Blackwell, A., Anderson, R., Grant, A.: Password Memorability and Security: Empirical Results. IEEE Security and Privacy, September/October, 25–31 (2004)
21. http://middleware.internet2.edu/hepki-tag/pki-lite/pki-lite-policy-practices-current.html
22. http://www.meti.go.jp/policy/netsecurity/digitalsign-law.htm
23. http://www.tagpma.org/