Journal of Information Processing Vol. 19 263–273 (July 2011)

*Regular Paper*

# Graded Trust of Certificates and Its Management with Extended Path Validation

Hiroyuki Sato[†1] and Akira Kubo

In modern information service architectures, many servers are involved in service building, in which servers must rely on the information provided by other servers thereby creating a trust. This trust relation is central to building services in distributed environments, and is closely related to information security. Almost every standard on information security is concerned with the internal control of an organization, and particularly with authentication. In this paper, 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 the fragmentation problem caused by complexities of paths and by fine a requirement at 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 trust domain. LoC is evaluated in terms of the certificate 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 a single 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 costs of maintaining lists of trust anchors at multiple levels.

## 1. Introduction

In modern information service architectures, many servers are involved in service building. For example, in a service federation, it is common that a user agent interacts with an authentication server and several service providers. Moreover, the validation of a digital certificate generally includes intermediate certificates issued by multiple CAs (Certification Authority). In these scenarios, the servers must rely on the information provided by other servers. This *trust* relation is central to building services in distributed environments.

†1 The University of Tokyo

As distributed environments grow in size and complexity, controlling trust becomes complicated. Users require both easy-to-control and fine grained trust. Easy-to-control trust is required by the internal control within an organization. Because trust must be provided to all services in an organization, it must be easy for an organization to manage trust. On the other hand, fine grained trust is required for services at multiple levels of significance. If the level of trust in handling privacy and that public information are separately managed, we can save costs by concentrating on managing trust in handling privacy.

Therefore, we can say that trust is closely related to information security. Although security is discussed in terms of computer security, network security, and information security in a modern security framework, it is not long before information security is considered to be important. Information security is concerned with control in the behavior of systems and humans for protecting information assets. One of the major differences of information security compared to others is that organizations are major players of security. "Internal control" is discussed organization-wise. Security policies are also defined and published organization-wise.

The best way to evaluate the internal control of an organization and security policies are as follows: we (stakeholders) make an agreement on certain criteria for evaluation, we publish the results evaluated by using the criteria, and prove that the evaluation is correct. There are several standards defined on information security together with the security audit framework. For example, ISMS (Information Security Management System), or ISO 27001, is commonly used as a criterion of system and information security.

Among several issues of internal control, authentication is the most critical one. Allowing access to critical information assets is guaranteed by how assured the adopted authentication is. The level of assurance of a given 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 (Certificate Policy) of certificates.

Let us consider a trust circle in which there are some Id providers (IdP) and service providers (SP). If an SP has valuable information assets, and requires a high level authentication for accepting access to valuable assets, then a strong

mechanism is required. Although it must be stressed that the strength of authentication is generally explained by its mechanism, issues related to administrations and operations such as initial setups and lifecycle management of credentials and IDs must also be considered. In certificate authentication, there is an established framework to evaluate them: CP/CPS (Certificate Policy/Certificate Practice Statement)'s of certificates with audit. Matching of the strength of IdP and requirements by SPs are the source of trust in a service 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. By evaluating CP/CPS's of a target CA, a certificate is issued to guarantee that an issuing CA trusts the target CA. Thus, a certificate chain is constructed in such a way that the trust is chained. 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 the trust is 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 CAs as levels, which proliferates the number of CAs. Actually, major commercial PKI vendors operate as many CAs of different assurance levels 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 causes a long negotiation in building a bridge CA. To control such a fragmentation is strongly required.

In regard to this fragmentation problem, one realistic solution for an organization is to manage only CAs of higher level, and to leave CAs of lower levels uncontrolled. Because uncontorlled CAs are not in the trust circle, no trust problem is observed. But, in terms of organizational internal control, such CAs must be controlled because it is true that they are operated in the organization. The trust circle must be extended to cover them in terms of internal control.

In this paper, we propose *"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. It is possible that some criteria approved by a group of CAs are used to calculate the level. 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 on CP/CPS of a bridge CA, this policy management server accepts multiple levels of CAs. Therefore, we can flexibly evaluate levels of CP/CPS's in one single 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 follows: Section 2 studies some 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. Fragmented Trust Problem of Certificates

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

Authentication is a key mechanism that implements levels of significance. The idea is to control access by using the information to assure the authentication. Generally, they are called *"level of assurance (LoA)"*, which are defined by some standards of LoA such as NIST 800-63 [4] for evaluating levels. In such situations,

an SP requires an appropriate LoA to an IdP for accessing its information assets. In a fixed trust circle, it is common that its member IdPs and SPs are under some agreement to maintain a certain LoA of IdPs, which actually maintains trust within the circle.

Among various authentication mechanisms, certificate authentication is usually given the highest LoA. Certificate authentication includes a process of path validation: a path from one CA (A) to another CA (B) is constructed, if B trusts A. If the root of the path constructed in the validation process is within the domain of the trusted CAs, then the validation, and therefore the authentication succeeds. Thus, the trust of certificates is reduced to a path construction whose root is trusted.

Although certificate authentication can be given the highest LoA, it has a problem of flexibility in building trust. That is, trust built by path validation causes inflexible management of LoA in the trust domains. In general, CAs are operated at various levels of CP/CPS's. Some CP/CPS's can be stricter, which provides a higher LoA to SPs. Therefore, SPs that require a higher LoA will trust only strictly operated CAs. This results in an inconvenience to users and a high cost in operations. Today, a solution to such trade-offs is offered in a way that an organization operates CAs by multiple roots which correspond to multiple levels of operations. Looser certificates are used to access less important information assets with less cost than strictly issued certificates.

Typical examples of multiple roots can be seen in server certificates. Today, most major Web browsers classify server certificates by using two major criteria, EV certificates [5], WTCA (Web Trust CA) [3], and others which do not match the two. In the path validation of server certificates, these three criteria never intersect. Under the Web model of trust, an individual Web browser controls the list of trusted CAs. Even within WTCA, levels of assurance differ. Some CAs guarantee that an issued certificate is owned by an organization, whereas some only guarantee that an issued certificate belongs to a domain. We can understand this kind of differences only by inspecting CP/CPS's. Therefore, strictly, a verifier must always be aware of what is guaranteed by a WTCA certificate.

Also in client certificates, major vendors such as Verisign provide multiple roots at different levels of trust. In a complex organization, the situation is very similar to the real world. There are many organizational units with various levels of independence, and many services that require various LoAs. The result would be many CAs of various LoAs to handle the various requirements of services.

These two examples are not exceptional. When an (attractive) SP requires a specific level of assurance, it is natural that a part of IdPs make a separate trust circle to meet the level, resulting in the fragmentation of trust.

This kind of scenarios cause the fragmentation and furthermore a maintenance problem in the path validation. In the real world, many CAs are established to provide required LoAs, increasing the cost of maintenance in trust domains. The domain is fragmented according to LoAs.

Moreover, if a required level of SP changes, the list of trusted CAs must be modified accordingly, which causes a problem in maintenance. Even if no change occurs to an SP, the maintenance is still a problem because the world of CAs continuously changes.

Another problem occurs that once a new trust circle emerges, we must evaluate its level of assurance. Today, a number of local criteria of trust have been established. Comparing criteria is often difficult because in a usual case one criterion is specified on different axes from other criteria. From this view, we must manage evaluation axes of criteria to save the evaluation cost of the level of assurance. We can obtain a hierarchy of levels of trust by evaluating criteria based on a single meta-criteria (criteria on evaluation axes).

Thus, we see that both a fine control of LoA requirements and an uncontrolled establishment of trust circles with their own criteria cause fragmentation problems.

## 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 a key role.

### 3.1 Software Architecture of Dynamic Path Validation

The goals of dynamic path validation are to provide a framework of path building in which it is possible to house CAs with multiple LoAs, and to build a graded or flexible path depending on LoA. It solves the problem of fragmented trust by

housing many SPs and CAs, if these agree on the evaluation by the certificate common criteria [⋆1] proposed by Policy Authority, and by making the path validation flexible.

In our scenario, there are three players: end entities, SP, and Policy Authority. An end entity requests the authentication for a service with his/her certificates. An SP is a server that requests the authentication of a user. In this scenario, an SP requires that a certificate of an end entity has a certain level of assurance. The SP 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 SP given as the level of certificates are 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 explains the name of "dynamic" path validation. We illustrate our software architecture in **Fig. 1**.

### 3.1.1 Policy Authority

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

(1) to decide and publish the certificate 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 SPs.

In Fig. 1, (1) corresponds to "Certificate Common Criteria," (2) to DB of Levels, and (3) to the 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, the delegation must be operated in an appropriate way.

In general, the 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.

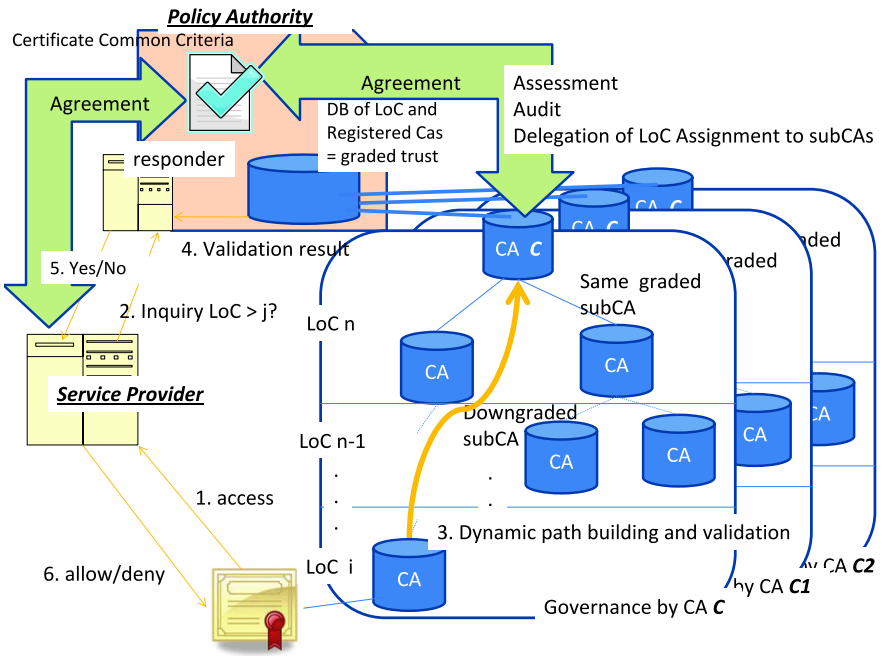



**Fig. 1** Softeware architecture of dynamic path validation.

In this way, with Policy Authority as the core, CAs and SPs participate in a circle. This simulates the circle of trust in Liberty [25]-like federations. The difference is that in the latter (Liberty), IdPs and SPs 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 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. With Policy Authority acting as the pivot of the circle, we can save the cost of building multiple circles. This scenario resembles putting bridge CAs as a pivot in path building.

⋆1 We use the term "certificate common criteria" to distinguish it with Common Criteteria (ISO/IEC 15048) of information security.

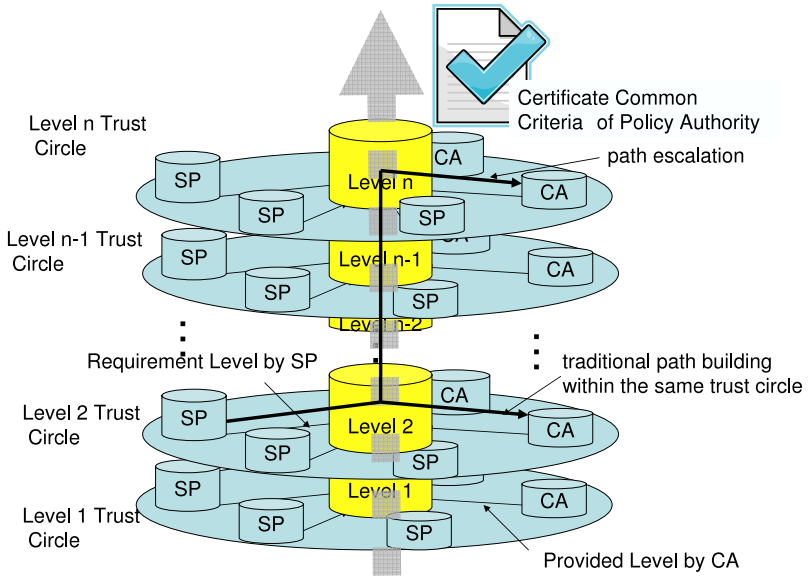


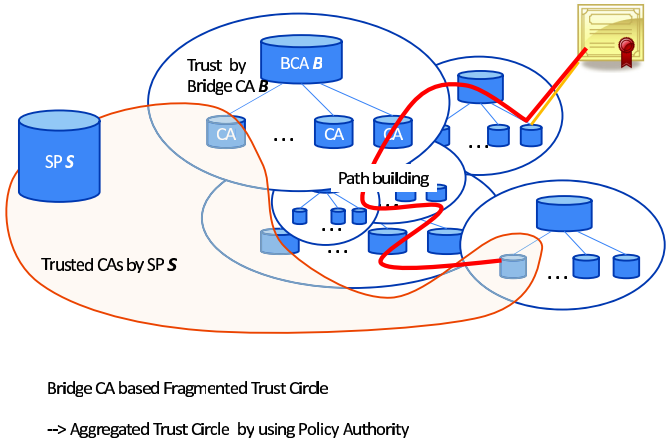

**Fig. 2** Graded circle of trust consisting of servers of various LoCs, with Policy Authority playing the pivot.

### 3.1.2 Graded Trust Circle

The trust model of our framework is naturally graded. In Fig. 2, there are multiple circles according to levels provided by Policy Authority. When a CA is given a level $n$, it is in circle $n$. When an SP requires a level $m$, it is in circle $m$. The path validation requested by an SP of level $m$ succeeds, if the root of the constructed path is in circle $m$ (the same as the conventional trust), or in circle $n$ ($n > m$). The path escalation from $m$ to $n$ is guaranteed by the certificate common criteria of Policy Authority. This trust model is essential for solving the fragmentation problems. We discuss the solution in Section 3.4.

### 3.1.3 SP and CA

An SP 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 SP maintains its requirement to levels of certificates. Policy Authority, or the delegated server receives the requirement together with a path, then validates



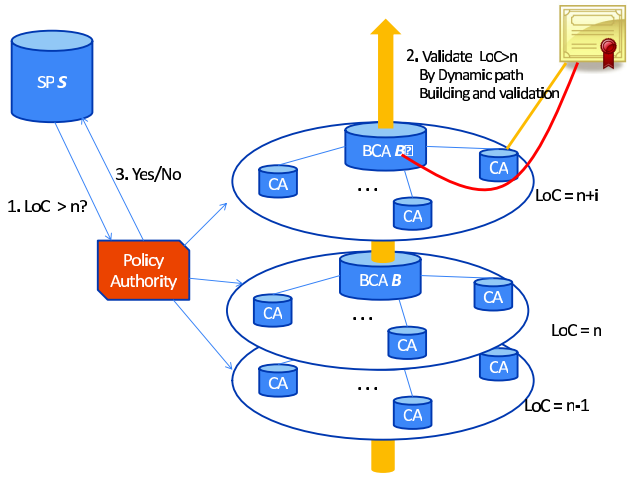




**Fig. 3** Housing multiple path validations in one Policy Authority.

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.

### 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 some certificate common criteria such as EV, Web Trust and RFC 3647.

#### 3.2.1 Levels of Compliance of Certificates

Conventionally, a level of assurance is given to a CA according to a specific criterion and certification based on the criteria and the related audit. Typical criteria include WTCA [3], EV [5], and "Specified Certification Business" [27] in Japan. All of these criteria require the 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 criterion 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 criterion as a numeric value that represents how strictly a server is operated in compliance with the given criterion.

As a criterion of LoC, we have some standard templates for CP/CPS's such as EV, WTCA, and RFC 3647. Conventionally, because they define the minimum set of requirements, compliance levels are just 0/1 (yes or no). In our extended compliance, we may specify optional criteria that enhance the level of security in the same template. Therefore, we have more than two levels for compliance: enhanced compliance/minimum compliance/no. Here, LoC can be considered as an extension of a 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 of concern. This is the same framework as the assignment of LoA in which the level of ID providers is of concern.

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

#### 3.2.2 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. Delegation is essential in saving the 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 a 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. Policy Authority must therefore enforce this restriction on every participating CAs. Audit must also be effective for this enforcement.

#### 3.2.3 Evaluation Axes of Criteria

In the NIST standard [4], we see four axes for evaluating authentications. In this paper, we adopt related two axes of evaluation: the levels of the initial identification (ID lifecycle management in general [21),22)]), and the levels of tokens. For example, Verisign defines three levels as for levels of assurance depending on the methods of the 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 its security constraints in operations. There are the proposal [21] that criteria for SPs are necessary in addition to criteria for IdPs on which LoAs are based. Here, we propose our evaluation axes of criteria in terms of criteria for SP [21]:

(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 an *"dynamic path validation (DyPV)."*

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

An extended certificate chain is implemented as a list of pairs $(n_i, C_i)$ $(i \geq 0)$ where

(1) $C_0$ is the certificate of a root CA in an organization, and $n_0$ is the assigned level given to the CA.
(2) $C_i$ $(i > 0)$ is the certificate of a CA, signed by the CA whose certificate is $C_{i-1}$. Moreover $n_{i-1} \geq n_i$. $n_i$ represents the LoC of CA of $C_i$ given by the CA of $C_{i-1}$.

Note that for a given CA, there can be two certificates $C_1$ and $C_2$ that comprise different paths from the root CA. Accordingly, LoCs attached to $C_1$ and $C_2$ are different in general. Although the situation that a CA has multiple LoCs is not standard in terms of the internal control of an organization, Policy Authority allows this situation, which is represented as storing pairs $(n, C)$ of certificate $C$ signed by a parent CA, and LoC $n$ given by the parent.

This representation is an actual extension of conventional certificate chains where a list of certificates only matters. However, an extended chain has a weaker guarantee than the conventional chain, meaning that in the extended chain, a CA signs a less managed CA. We need an extension field in order to distinguish conventional signing chains from extended chains.

#### 3.3.1 Management of LoCs in Policy Authority

Policy Authority has a list of participating organizations that have one or multiple root CAs. Note that they are the target of level assignment and audit by Policy Authority. It is essential that Policy Authority delegates level assignment and audit to subordinate CAs in an organization in view of saving the management cost.

First, Policy Authority accepts the participation of an organization. A level is assigned to the root CA(s) of the organization by Policy Authority. This grade cannot be modified without audit and negotiation between Policy Authority and participating organizations.

Derived (delegated) level assignment is implemented as unconditional acceptance of extended certificate chains of CAs submitted by the root CAs of participating organizations. Concretely,

(1) the root CA builds an extended certificate chain whose root is themselves.
(2) they submit the chain to Policy Authority.
(3) Policy Authority receives the chain, and checks its validity. The validity is examined as:
  (a) the level of the root CA of the chain is the preassigned one.
  (b) if $\mathrm{CA}_1$ of LoC $n_1$ gives a signature to $\mathrm{CA}_2$ of LoC $n_2$, $n_1 \geq n_2$.
(4) if the chain is validated, Policy Authority registers CAs in the chain together with their LoCs.
(5) at the registration, there can be updates of levels of subordinate CAs. Policy Authority manages the consistency related to the registration.

The consistency problem arises when there is a case that the registered LoC $n_{\mathrm{new}}$ of a given CA of $C$ is different from a previously registered LoC $n_{\mathrm{old}}$. If $n_{\mathrm{new}} \geq n_{\mathrm{old}}$, there is no inconsistency. If $n_{\mathrm{new}} < n_{\mathrm{old}}$, there can be subordinate CAs that are children of $C$, and its LoC is larger than $n_{\mathrm{new}}$. In this case, Policy Authority must invalidate such chains, and notify the result to the organization.

Updates of LoC must be a target of audit by Policy Authority. Because it can remember updates of LoCs at the registration of chains as above, we can say that there is enough data of this audit.

#### 3.3.2 Path Validation

The algorithm of DyPV validation is given in **Fig. 4**. A validating SP delegates

```
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.
   } 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.

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 SP. 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 SP 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. This case can occur when there is temporal inconsistency in LoC of CAs caused by LoC updates independently done by suborganizations. (Such inconsistency of two different LoCs must eventually be resolved in view of the internal control in the organization.) In our algorithm, we allow that 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 the validation of any extension of the path also succeeds. Therefore, we require that if there is a path from $\mathrm{CA}_1 \longrightarrow \mathrm{CA}_2$, meaning that $\mathrm{CA}_1$ issues a certificate to $\mathrm{CA}_2$, then their LoCs must satisfy $LoC(CA_1) \geq LoC(CA_2)$. This consistency must be maintained by Policy Authority offline as explained in this section.

### 3.4 Solution of Fragmented Trust Problem

Under the umbrella of Policy Authority, CAs of multiple levels are accommodated. Conventionally, trust must be maintained according to the level of participating entities, which has caused fragmentation of trust circles and cumbersome negotiations in accepting new participants. In our framework, instead, we define multiple LoCs for participation, because the evaluation is done in reference to a published single criterion for which participants can select its appropriate LoC depending on the cost, the significance and the easiness of entry into the trust circle.

The path building is extended so that LoC is considered. Policy Authority validates a path so that LoC of every CA on the path is larger than the required LoC. CAs of different LoCs supply different constructions of paths. Conventionally, the path building must be done within trusted CAs. Different trust requires different path building, which causes fragmentation in path building. Instead, we provide an aggregated framework of trust in terms of certificate common criteria. This saves the cost of trust maintenance of a single certificate common criterion under the umbrella of Policy Authority.

Our framework resembles establishing another trusted third party. However,

in addition to 0/1 (yes/no) information of trusted or not, trust levels can be controlled, which extends the functions of trusted third parties. There are some approaches that aims at controlling levels. For example, TFPAP (Trust Framework Providor Adoption Process) [10] has started giving its "common" LoAs to its participating trust providers, and SPs can control access by judging the LoA of IdP and referring to the grade given by TFPAP. Although no validation method of levels other than manual validation is provided, we can say that the idea of this approach is similar to our framework.

### 3.5 Comparison with RFC 5280

Conventionally, path validation is defined as RFC 5280 [9]. In RFC 5280, 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 the certificate 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 online.

## 4. Related Work

This paper is a refinement of the basic idea of dynamic path validation [13] in terms of extended path validation and analysis of the fragmentation problem.

Path building [8], and validation [9] 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 some problems other than technical ones which hinder its growth. Furthermore, delegation of tasks related to them is studied because they are too heavy for general SPs. The discussions are summarized as RFC 3379 [20]. OCSP [14] and SCVP [11] are also classified as protocols partly delegating validation. Our framework is also classified as delegation. Ours considers LoCs in path validation.

Another approach is to lessen the maintenance of trust by using online rating of trust network. For example, instead of formal procedures of trust evaluation, trust values are calculated by antecedents [6] or by local trust [16].

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 [7], and PKI lite [26].

Evaluating IdPs and assigning a 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 [17,19].

Moreover, in Grid, there are established policy management authorities [28] to enforce the policies of Grid on participants.

Although discussions of LoA [15] have been limited to ID and authentication, they are very fruitful in assuring the 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 [18] and the NIST standard [4] are milestones in the discussion. They are also driving forces to define LoA to large federations. Today, LoA is widely discussed in many organizations, grids, federations [12], and inter-federations [1]. In the U.S., some frameworks such as TFPAP [10] have been initiated for establishing common criteria for evaluating LoAs. We can say that they share the goal with our framework.

LoA can be generalized to SPs [21,22]. A consistent assignment of LoA to SPs is proposed in terms of security policies of organizations.

Note that all of these must be done as part of risk management [23]. Authentication can be discussed in terms of risk management [24]. Moreover, there is defined Common Criteria (ISO/IEC 15408) for evaluating information security.

## 5. Concluding Remarks

In this paper, we have proposed dynamic path validation (DyPV) to tame the fragmentation problem. First, we have studied the fragmentation problem of trust in certificate path building. The fragmentation causes complicated path building together with complicated organization of CAs. Next, 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 certificate common criterion, LoC is assigned to each CA. In this way, Policy Authority accommodates 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.

Moreover, we have shown that DyPV can be a solution to the fragmentation problem.

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 certificate common criteria of Policy Authority are 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. and 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. and 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) Chua, F. and Lim, E.: Trust network inference for online rating data using generative models, *Proc. 16th Int'l Conf. Knowledge Discovery and Data Mining*, pp.889–897 (2010).
7) Chokbani, S., Ford, W., Sabett, R., Merrill, C. and Wu, S.: Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework, RFC 3647 (2003).
8) Cooper, M., Dzambasow, Y., Joseph, S. and Nicholas, R., Internet X.509 Public Key Infrastructure: Certification Path Building, RFC 4158 (2005).
9) Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R. and Polk, W.: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, RFC 5280 (2008).
10) Federal Identity, Credentialing and Access Management: Trust Framework Provider Adoption Process (TFPAP) For Levels of Assurance 1, 2, and Non-PKI 3 (2009).
11) Freeman, T., Housley, R., Malpani, A., Cooper, D. and Polk, W.: Server-Based Certificate Validation Protocol, RFC 5055 (2007).
12) InCommon Federation: Identity Assurance Profiles Bronze and Silver, http://www.incommonfederation.org/docs/assurance/InC_Bronze-Silver_IAP_1.0_Final.pdf (2008).
13) Kubo, A. and Sato, H.: Design of Graded Trusts by Using Dynamic Path Validation, *IFIP Trust Management 2010*, pp.172–183 (2010).
14) Myers, M., Ankney, R., Malpani, A., Galperin, S. and Adams, C.: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP, RFC 2560 (1999).
15) Nedanic, A., Zhang, N., Yao, L. and Morrow, T.: Levels of Authentication Assurance: An Investigation, *Proc. 3rd Int'l Symposium on Information Assurance and Security*, pp.155–158 (2007).
16) Nordheimer, K., Schulze, T. and Veit, D.: Trustworthiness in Networks: A Simulation Approach for Approximating Local Trust and Distrust Values, *IFIP Trust Management 2010*, pp.157–171 (2010).
17) OASIS: Level of Assurance Authentication Context Profiles for SAML 2.0 (2009).
18) Office of Management and Budget (U.S.): E-Authentication Guidance for Federal Agencies, M-04-04 (2003).
19) OpenID: OpenID Provider Authentication Policy Extension 1.0 (2008).
20) Pinkas, D. and Housley, R.: Delegated Path Validation and Delegated Path Discovery Protocol Requirements, RFC 3379 (2002).
21) Sato, H.: A Service Framework based on Grades of IdPs and SPs, *Proc. Security and Management 2009*, pp.379–385 (2009).
22) Sato, H.: $N \pm \varepsilon$: Reflecting Local Risk Assessment in LoA, *Information Security 2009* (OTM 2009, LNCS 5871), pp.833–847 (2009).
23) Stoneburner, G., Goguen, A. and Feringa, A.: Risk Management Guide for Information Technology Systems, NIST 800-30 (2002).
24) Yan, J., Blackwell, A., Anderson, R. and Grant, A.: Password Memorability and Security: Empirical Results, *IEEE Security and Privacy*, September/October, pp.25–31 (2004).
25) http://kantarainitiative.org/
26) http://middleware.internet2.edu/hepki-tag/pki-lite/pki-lite-policy-practices-current.html
27) http://www.meti.go.jp/policy/netsecurity/digitalsign-law.htm
28) http://www.tagpma.org/


(Received October 29, 2010)
(Accepted April 8, 2011)
(Released July 6, 2011)

**Hiroyuki Sato** is Associate Professor in the University of Tokyo. He received B.Sc., M.Sc. and Ph.D. from the University of Tokyo in 1985, 1987, 1990, respectively. He is majoring Computer Science and Information Security.

**Akira Kubo** worked in Baltimore, Japan, engaged in PKI from 1996 to 2003. From 2003 to 2006 he worked in NTT Communications, engaged in PKI. From 2006 he is working in CSE Ltd.