Policy Formalization to combine separate systems into larger connected network of trust

V.Casola, A.Mazzeo, N.Mazzocca, V.Vittorini
Dipartimento di Ingegneria dell'Informazione, Seconda Università di Napoli, Naples, Italy
Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli "Federico II", Naples, Italy

Abstract: This work presents a certificate policy formalization approach that copes with the automated policy mapping problem. We briefly describe: a) the steps of the proposed method; b) an XML-based implementation of our approach and c) an example of comparison between two different policies.

Key words: Public Key Certificate, Certificate Policy Formalization, Policy Comparison

1. CERTIFICATE POLICY FORMALIZATION

In the recent years, the need of Policies to dynamically face security problems has grown in all computer system areas, especially in Security Infrastructures (PKI) on which our attention is focused.

The verification of Certificate Policy provisions is not yet an explicit procedure for the validation of Public Key Certificate (PKC), but certainly a formal Policy representation should help CAs to trust a certificate especially if it has been issued in an un-trusted domain.

In literature, some examples of formalization of general security policies and Certificate Policies are available [Klobucar et al.],[Grimm et al.],[Mendes et al],[Thompson et al]. Probably, the most detailed and relevant suggestion for the formal presentation of Certificate Policies was described in the RFC2527 [RFC2527].

Nevertheless, all policy frameworks, including RFC2527, present wide limits, they are not sufficiently structured to resolve ambiguity problems, neither in the formal verification stage nor when a CA wants to decide which

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35620-4_43
D. Gaïti et al. (eds.), *Network Control and Engineering for QoS, Security and Mobility*

certificate policies from other security domains can be considered equivalent [Polk et al].

2. THE PROPOSED METHOD

Usually the formal representation of the certificate policy defines a hierarchical structure of the provisions. We have refined the textual provisions in a more fine-grain and defined a grammar to automatically compare them.

The proposed method consists of the following steps:

Step 1) definition of macro-provisions;

Step 2) definition of second level provisions;

Step 3) grammar definition;

As mentioned above, the RFC2527 is a good example of policy formalization so we have decided to use its structure to implement the first two steps of our method. According to the RFC2527, the first level provisions includes:

a)Introduction, b)General Provisions, c) Identification and Authentication, d) Operational Requirements, e) Physical, Procedural and Personnel Security, f) Technical Security Control, g) Certificate and CRL Profiles, h) Specification Administration.

An example of second level provisions (Technical Security Control) includes:

f1) Key Pair Generation, f2) Private Key Protection, f3) Other Aspect of Key Pair Management, f4) Activation Data, f5) Computer Security Controls, f6) Life Cycle Technical Control, f7) Network Security Control, f8) Cryptographic module engineering controls

At this point we have defined the main provisions of our structure which are actually filled with textual descriptions.

The provisions defined in the first two steps are very complex objects and this is the most important reason of ambiguity. The hierarchical structure consists of a set of couples (element-type, value) representing provisions and sub-provisions, where the "value" itself may be a complex object. In the last step of our method, we have structured all critical fields until we have obtained "leaves" representing the definition of very simple types (enumerative sets or numeric types), thus creating a grammar based on such data-structure set that could be automatically processed.

We first illustrate two examples of the formalized logical structures in Fig.1 and Fig. 2, and then we define a way to implement them.

```
KeySizes= {Alghorithm, BitLenght};
PrivateKeyProtectionBOX = {NoProtection, OnLocalPC, OnFloppy,
OnSmartCard, OnSmartCardWithBiometrciSensor};
```

Figure 1 Logical Structure of the provisions: Key Length and Private Key Protection

```
PublicationRepository={PublicationCAInformation, FrequencyPublication,
AccessControls, Repositories};
PublicationCAInformation={PolicyIssuanceFrequency,
PublishedCertificateIssuanceFrequency, CRLIssuanceFrequency };
AccessControls={OnLineByAnyone,OnLineBySubscribedUsers,....}
```

Figure 2. Logical Structure of the provision: Publication Repository

In the above examples, the logical structure of the provisions include both very simple types that primarily correspond to quantifiable parameters (Bit length, CRL Issuance Frequency), and elements that have a more complex and structured representation. The former are usually used in available literature when comparing policies, but the applicability of our method to the latter is the true innovation of our job.

By XML we have implemented both the elements and the structured types of the grammar used to model the whole policy. Fig. 3 and Fig.4 show the XML structure of the examples in Fig. 1 and Fig.2;

```
<KeySizes>
   <Alghorithm>..... </ Algorithm >
   <BitLenght>..... </ BitLenght >
 </KeySizes>
.....
<PrivateKeyProtectionBOX>
    <PrivateKeyNoProtection>.....</PrivateKeyNoProtection>
   <PrivateKeyOnLocalPC>.....</PrivateKeyOnLocalPC>
   <PrivateKeyOnFloppy>.....</PrivateKeyOnFloppy>
   <PrivateKeyOnSmartCard>.....</PrivateKeyOnSmartCard>
   <PrivateKeyOnSmartCardWithBiometricSensor>
      .....
   </PrivateKeyOnSmartCardWithBiometricSensor>
</PrivateKeyProtectionBOX>
```

Figure 3. XML description of some Technical Security Provisions

```
<FrequencyPublication>
  < PolicyIssuanceFrequency>.....</ PolicyIssuanceFrequency>
  <PublishedCertificateIssuanceFrequency>.....
  </ PublishedCertificateIssuanceFrequency>
  < CRLIssuanceFrequency>.....</ CRLIssuanceFrequency>
  < FrequencyDay>.....</ FrequencyDay >
  < FrequencyHours>.....</ FrequencyHours >
  < FrequencyMinutes>.....</ FrequencyMinutes >
  < FrequencySeconds>.....</ FrequencySeconds >
</FrequencyPublication>
```

Figure 4. XML implementation of some Publication Repository Provisions

A limit of our approach is the number of data structures and that complex combinations of them are necessary to evaluate the security level associated to a certificate. Moreover some elements are hardly to formalize in a parametric way. Nevertheless we have been able to obtain very fine and useful information from a wide range of "critical" provisions .

3. CASE STUDY: CROSS-CERTIFICATION

In this section we briefly illustrate how the formal structure presented in the last section could help during a cross certification, for example when the relying parties want to state the conditions which trustworthy policy provisions have to match with, or to find specific provisions of the policy to trust, for a more accurate analysis.

In the examples we refer to two significant certificate policies:

1. The Government of Canada Certificate Policy (digital Signature basic assurance, GofC for short) [Digital Signature and Confidentiality];
2. Manuale operativo per il servizio di certificazione di chiavi pubbliche per la RUPA (registered name, CT for short)[Centro Tecnico].

GofC is formalized according to the RFC2527; CT is an Italian policy, written in Italian and structured according to a different template.

Table 1 summarizes two examples of comparison between provisions of GofC and CT whose logical structure has been presented in Fig3. The first raw of each example contains the textual form of the related policy Section, the second raw contains its formalization according to the defined criteria.

In Example 1 we compare the provision *BitLenght.* This is a simple case because the set value is a numerical one, so it already exists a total order relation among the elements.

In Example 2 we compare the provision PrivateKeyProtectionBOX. This is a more interesting example because it widely shows the advantage of our approach. In this case, GofC has a lower security level compared with CT because GofC allows two different ways to protect the private key (*PrivateKey OnLocalPC, PrivateKeyOnSmartCard*).

Table 1: Examples of Policy Comparison

		Policy 1: GofC	Policy 2: CT
Example 1	Plain text	Asymmetric Key Size = A CA must ensure that the key pairs for all PKI entities must be 2048 bit RSA . (From Section 6.1.5)	Asymmetric Key Size = A CA must ensure that the keys for digital signature operations must be 1024 bit Algorithm = for digital signature generation and verification, the following algorithm must be used: RSA (From Section 10.2 and 10.3)
	XML formalization	<KeyPairGenerationInstallation> ..... <KeySizes> <Algorithm> RSA </ Alghorithm > **<BitLenght>**2048 </ BitLenght > </KeySizes> ..... </KeyPairGenerationInstallation>	<KeyPairGenerationInstallation> ..... <KeySizes> <Algorithm> RSA </ Algorithm > **<BitLenght>**1024 </ BitLenght > </KeySizes> ..... </KeyPairGenerationInstallation>
Example 2	Plain text	Hardware\Software key generation= Key pairs for all entities may be generated in a software or hardware cryptographic module (From Section 6.1.8)	Digital Signature Token for key generation = Key pairs may be generated and protected in the digital signature token (Smart Card); private key escrow is denied. (From Section 10.2)
	XML formalization	<PrivateKeyProtectionBOX> **<PrivateKeyOnLocalPC>** software cryptographic module </PrivateKeyOnLocalPC> **<PrivateKeyOnSmartCard>** hardware cryptographic module </PrivateKeyOnSmartCard> </PrivateKeyProtectionBOX>	<PrivateKeyProtectionBOX> **<PrivateKeyOnSmartCard>** hardware cryptographic module </PrivateKeyOnSmartCard> </PrivateKeyProtectionBOX>

From the above examples it is plain that also a simple parser could be used to extract from the proposed structure the information needed to automatically perform a policy mapping, once that the proper security metrics have been defined.

4. CONCLUSIONS AND FUTURE WORKS

A formalized certification Policy could represent a very interesting solution to face the lack of automatic process to verify and accept a certificate, especially if it is issued in an untrusted domain.

In this work we have proposed an approach to policy formalization which is based on the definition of a grammar of elements and types.

We are currently working to integrate our approach with the definition of proper security metrics to evaluate a PKI trust level and to implement automatic techniques to efficiently manage a certificate policy in its all life cycle.

5. REFERENCES

RFC: 2527 S. Chokhani, W. Ford Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework, March 1999

Centro Tecnico per la Rete Unitaria, Sezione Sicurezza: Manuale operativo per il servizio di certificazione di chiavi pubbliche per RUPA. Versione 1.1 , 4 Maggio 2001

Digital Signature and Confidentiality, *Certificate Policies* for the Government of Canada Public Key Infrastructure, version 3.02 , April 1999

T. Klobucar, B. Jerman-Blazic, A Formalization and evaluation of certificate policies, Computer Communication 22 (1999), 1104-1110

R. Grimm, T.Hetschold, Security Policies in OSI management: experiences from the DeTeBerkom Project BMSec, in: Proceedings of the JENC6,1995

Mendes, S.; Huitema, C., A new approach to the X.509 framework: allowing a global authentication infrastructure without a global trust model, Proceedings of the 1995 Symposium on Network and Distributed System Security (SNDSS'95)

M. R. Thompson, S. Mudumbai, A. Essiari, W. Chin Authorization Policy in a PKI Environment, Proceedings of the 1st Annual NIST workshop on PKI

W.Polk, N. Hastings, Bridge Certification Authorities: Connecting B2B Public Key Infrastractures, September 2000