Securing Policy Negotiation for Socio-Pervasive Business Microinteractions

Secure, Simple and Efficient Implementation of Policy Negotiation

Mitja Vardjan
Research department
SETCCE
Ljubljana, Slovenia
mitja.vardjan@setcce.si

Miroslav Pavleski
Research department
SETCCE
Ljubljana, Slovenia
miroslav.pavleski@setcce.si

Jan Porekar
Research department
SETCCE
Ljubljana, Slovenia
jan.porekar@setcce.si

Abstract— **In this paper, we study security of policy negotiation and policy-based agreements for emerging mobile based dynamic business environments that feature many previously unknown parties sharing services to each other in an ad hoc fashion. Signed agreements among parties are basic enablers of trust in such dynamic environments. Before a micro service or a 3rd party application can be consumed by an employee, a policy like Service Level Agreement (SLA) typically has to be agreed by the service consumer and service provider. Various types of policies have to be accepted also in other socially tailored use-cases, e.g., when an employee joins a community. To enable appropriate degree of trust, consumer privacy protection, as well as authenticity and non-repudiation of the final agreement, the policy negotiation process has to be secured. The security principles introduced in the paper are applicable to any kind of policy negotiation and selection where two entities are involved: the provider and the requester who need to establish trust between them. A simple but secure policy negotiation or selection is described, followed by description of its implementation on Android operating system. The interactions between the parties are minimized in order to boost usability and limit bandwidth usage in socio-pervasive environment.**

Keywords-security; trust; policy agreement; policy negotiation.

I. INTRODUCTION

Formal policies are used to legally define relationship between the parties and to specify conditions under which something is provided by one party and consumed by the other party. A prominent example is Service Level Agreement (SLA) that defines relations between service provider and service consumer. SLA represents the means through which the parties involved in service provision and consumption agree on different aspects of the terms of service in question. The business user acting as service consumer needs to be aware of what the service will be providing and is the starting point for definition of Quality of Service (QoS) metrics that the service will need to respect. This is especially important in dynamic environments placed in business oriented socio-pervasive scenarios (see [10]). Furthermore, the end-user needs to know how the service will treat his private and confidential information. If a fee is applicable, the service provider needs to be sure that the end-user will pay for the service and needs to provide information on how he is going to charge for the service. As SLA defines rights and obligations of both parties, all in relation to the service provided, it contains some private information about the parties, especially about the consumer, his preferences and possibly also some sort of his identity. The provider's identity is usually less critical and most providers disclose their identities voluntarily in advance in order to attract potential consumers to their services.

However, the scope of policy is not limited to collaboration between businesses via service sharing and to SLAs only. Same principles apply when an employee is trying to join a social group or a community that may have internal rules that are not compatible with his company's confidentiality policy. In more general terms, the parties will be referred to as the requester and the provider.

In any case, the requester and provider exchange some of their data during each step of policy negotiation. In each step they need to either limit the data to be sent to non-sensitive data only, which may result in other party terminating the negotiation due to incomplete or wrong data, or trust the other party would not abuse the received sensitive data. Obviously, the former option would severely limit the usefulness of such negotiation. In order to trust the other party, each side should at least be able to verify the other one's identity using digital certificates. They both also have to make sure that the final policy, which is a binding agreement, has not been modified during the process and they would not formally confirm such tainted agreement.

All verifications and data transmissions increase usage of network bandwidth and other resources. This is especially critical in a mobile environment with unreliable connectivity, limited battery power and possible costs of data transfer. Therefore, negotiation process and number of network transfers should be minimal.

Finally, after the policy has been negotiated, the agreement itself should be stored in a secure manner because it may contain sensitive private data and because it is an evidence of agreed terms between the business parties that can be used if one of the parties repudiates its involvement.

II. POLICY NEGOTIATION TYPES

Traditionally, a single policy is generated by the service provider or other authority and the requester (e.g., service consumer, user) can either accept or reject the policy. Even with this classic approach where there is only one option for the policy and the requester can either accept or reject it, the provider gets some information about the requester, usually his identity or point of contact and his interest or fondness of the particular type of service.

A more advanced alternative is to include negotiation between both parties where the requester can reach an agreement more suitable to his needs, possibly by merging or combining various policies using dedicated algebra [11] which is out of scope of this paper. The negotiation can be done either by choosing a policy among two or more policies offered by the provider (Figure 1) or by negotiating individual parts of the overall policy with the provider. The latter alternative [12] is most complex and while it allows for complete customization of the final policy it is also harder for the provider to generate and maintain. Furthermore, whether the negotiation succeeds or fails, such a complex negotiation can be used to extract additional private data from the requester during the negotiation process [1]. This paper focuses on the former alternative, i.e., the simple degenerated policy negotiation or policy selection process (Figure 1).

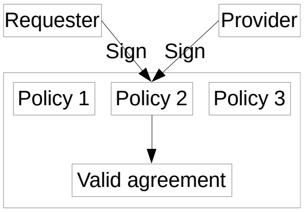


Figure 1: Policy Selection

In social and pervasive environments number of service advertisements and interactions may increase significantly. In such cases the SLA should be negotiated in a semi-automatic manner, bothering the user as little as possible and making some decisions automatically. The ability to automate policy negotiation depends on machine readability of the policy, while the security and trust aspects of policy negotiation described here are applicable for both semi-automatic and manual negotiations.

III. SECURITY PROPERTIES OF POLICY NEGOTIATION AND POLICY AGREEMENT

In order for the policy negotiation and its resulting agreement to be secure and to consequently enable trust between parties involved in service sharing or group collaboration the following security properties need to be provided:

- **Authentic evidence of non-repudiation of involvement in the negotiation process**: both parties involved in the negotiation process (the requester and provider) should not be able to deny that they were involved in the process of negotiation.
- **Confidentiality of negotiation process:** only parties involved in negotiation should be able to read and process messages passed in each step of negotiation.
- **Integrity of resulting policy or SLA agreement**: at the end of the negotiation process both parties should have identical documents that represent the agreement they have achieved with negotiation.

After the agreement is reached it will reside at each party's device for some time – at least until it is sent to the company's back end system to be archived. This may not happen instantly since mobile devices with limited connectivity or bandwidth are used in negotiation process. During this post-negotiation time an additional security property is therefore required:

- **The integrity of agreement storage:** the agreement document should not be modified before it is archived in company's back-end systems**.**

IV. SECURING NEGOTIATION

In this section, we describe how the required security properties of negotiation described in former section were satisfied. Security measures and technologies for satisfying the requirement of maintaining integrity of agreement storage are described in Section V. For each step of the negotiation process we describe the security measures, technologies, standards and implementations used to realize the required security properties. The steps or aspects of negotiation are the following:

- Provider's initial offering
- Negotiation phase
- Communication channel

A. *Provider's Initial Offer*

Regardless of the policy negotiation type, the requester has to be given some assurance that the provider's offer is real and the provider will not just collect the requests, possibly associate them with requester's data (e.g., his identity) and then not even provide the advertised service or advertised community membership.

The requester can put more trust into provider's offer if it is digitally signed by the provider using a verifiable certificate, especially if the provider is a known entity whose reputation can be affected by the requester. The trust in certificate authorities is important, but out of scope of this paper. X.509 [5] certificates are used in the described prototype.

Policy options can be prepared in advance, or generated on request. The latter alternative might not make sense because the requester does not provide any private data at the first step and the provider can tailor the policy only by vague data that can sometimes be gathered from the remote connection such as requester's IP address.

Technically, the policy options are encoded in an XML resource. The XML is in canonical form [7] and digitally signed with the provider's digital identity using XML-DSig [6]. Figure 7 shows the structure of the initial offer by example.

B. *Negotiation phase*

The whole process of policy negotiation is shown in Figure 3. At every step the parties verify the policy content has not changed from the previous step. This assures consistency of the policy during the whole negotiation process and prevents policy modifications by the other party.

At any time both parties can verify that the author of received message at each step of negotiation is the same as before. The author's identity can also be checked. A prerequisite is digital signing of policy by both parties at any step of negotiation and the usage of digital certificates.

If at any step a fraud is suspected by either party, it can safely terminate the process of negotiation. For example, if the other party (or somebody else in case of a man-in-the-middle attack) changes the terms during negotiation, or if signature verification fails, or if identity of the other party is not verifiable, then the negotiation is terminated.

The number of steps is minimized and shown in Figure 3. Figure 7 shows the initial policy options in collapsed view and the provider's signature. After getting the initial policy options, the requester locally (without interactions with provider) performs verifications (digital identity check and cryptographic signature validity check), chooses a suitable policy and signs it. The requester's signature is highlighted in Figure 8. Verification of provider identity can be skipped if Transport Layer Security (TLS) or similar protocol is used to verify it. Requester then sends the provider his choice and signature. XML-DSig [6] provides a convenient way to add requester's signature into XML-based policy. The provider locally verifies policy consistency and requester's signature. Before final policy is sent back to the requester, the provider appends another signature to the XML document. This time, the signed reference is not the policy, but the requester's signature of the policy. These two signatures are shown in Figure 9. The last signature is a confirmation and proof from the provider's side that not only the requester has agreed to the policy but also that the policy and requester's acceptance of the policy were successfully received by the provider. When the requester receives this final policy with provider's second signature, the requester archives this proof. At this point, the provider can not dispute the policy validity on grounds that the requester did not sign it or did not send it back.

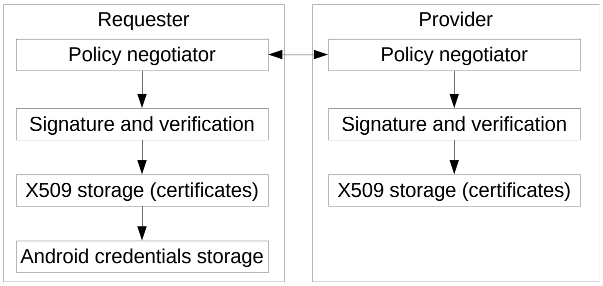

Requester
Policy negotiator
Signature and verification
X509 storage (certificates)
Android credentials storage
Provider
Policy negotiator
Signature and verification
X509 storage (certificates)

Figure 2: Requester and provider schematics

The prototype implementation is based on schematics in Figure 2. High-level negotiator components control the process and communicate between each other, while low-level components are used for signature management and storage of sensitive data.

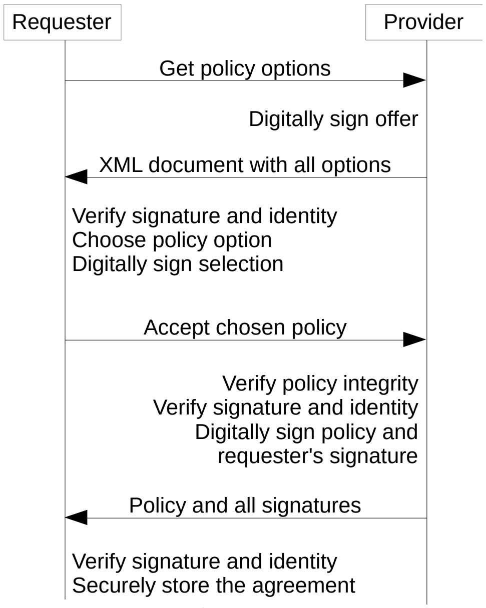

Requester
Provider
Get policy options
Digitally sign offer
XML document with all options
Verify signature and identity
Choose policy option
Digitally sign selection
Accept chosen policy
Verify policy integrity
Verify signature and identity
Digitally sign policy and
requester's signature
Policy and all signatures
Verify signature and identity
Securely store the agreement

Figure 3: Simplified sequence diagram

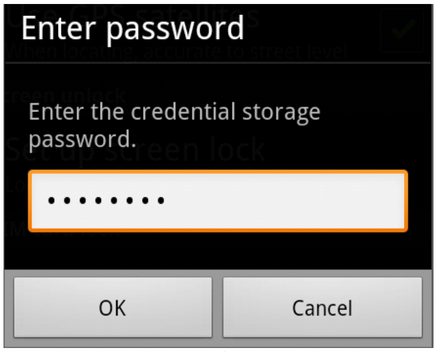
C. *Communication channel*

Typically, TCP/IP protocols are used and communication is secured using Transport Layer Security (TLS). Only server (in this case the provider) authentication is to be used because the identity of the requester can be revealed only if necessary at a later stage of the communication as opposite from the start of the communication mandatory by design of the TLS protocol. Technically, the client will reveal its digital identity only if and when it chooses to accept the policy.

Alternatively, instead of using wireless network, the whole negotiation could be done through Near Field Communication (NFC). Dodson and Lam [8] and Dodson et al. [9] describe the concept of micro-interactions through NFC which is applicable to policy negotiation as well. Although the close proximity required by NFC may increase requester's trust in the provider identity, the connection itself is insecure. To amend for this and encrypt the connection, a common encryption protocol like Transport Layer Security (TLS) can be used over NFC.

V. STORING THE FINAL AGREEMENT

The Android operating system facilitates the concept of Secure Credentials Storage which is used by system services to manage sensitive information such as passwords and keys. This sensitive information is stored in system protected files encrypted with a "Credential Storage Password" (Figure 4). The password is entered by the user during the first use of the negotiation and is not required for subsequent accesses to the protected data by same process. The data is still not decrypted and made available for other services in the system. Our prototype uses this system to store Policy Agreements in the phone before they are sent to company back-end storage systems.


Enter password

Enter the credential storage password.

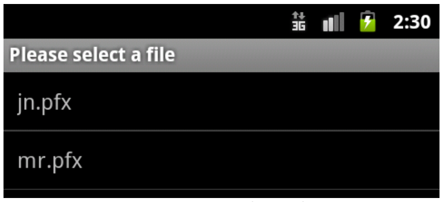
••••••••

Figure 4: Android Secure Storage. The first time credential storage is used user has to provide the password to unlock the storage. The window does not reappear for further requests.

Although our prototype can access the agreement at any time, it is problematic to export the agreement to the phone's memory card, which is the only area directly and easily accessible by the user, e.g., when the phone is connected to a computer. Any installed app with permission to read from external storage can access any file on the memory card. If future improvements of our prototype implement such export of the agreements, they should appropriately protect the exported data by encrypting and signing the data. Therefore, it may be more convenient to send the agreement using TLS protocol where the data is transparently decrypted at the recipient side.

VI. Installation of Certificates

Android Secure Storage is not designed for third party applications. However, since Android is free open source project, it is possible to use the Secure Storage for a custom purpose. A special application was developed to install certificates and private keys into Android's secure storage (Figure 5). Each certificate has to be unlocked by the user before it is put into Android Secure Storage in decrypted form (Figure 6). After the initial installation, certificates are conveniently – without entering password other than that for Android Secure Storage – picked by the user when he chooses the identity to present himself with during policy negotiation.

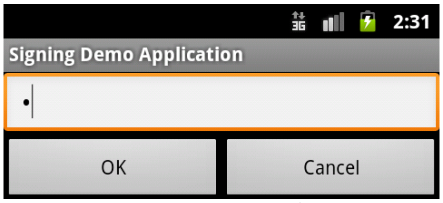


Please select a file

jn.pfx

mr.pfx

Figure 5: Installation of certificates



Signing Demo Application

•

Figure 6: Unlocking a certificate

Conclusion and Future Work

A secure and trustworthy policy negotiation was presented. The negotiation itself is simplified and policy is actually selected in a single step and not seriously negotiated. Although limited in negotiation possibilities, this fast and efficient approach still addresses the needs of most providers in the real world and offers great amount of security for both requester and provider sides by means of digital certificates. Digital signatures associated with those certificates and policy integrity shall be verified in each step of negotiation, regardless of any secure network connection. The final result is an XML-based document that contains the policy, digital identities of the signing parties, evidence of the negotiation process and provides non-repudiation of the negotiation process. A working Android prototype was described as an example of secure policy negotiation in mobile pervasive environment.

In future, the prototype is planned to support also semi-automatic policy selection based on multiple policies or micro-agreements that are in place at the time of negotiation. With this upgrade the prototype is planned to be integrated into a service platform.

Acknowledgment

Authors would like to thank the SOCIETIES project [4] consortium and EC for sponsoring the project.

References

[1] J. Porekar, K. Dolinar, A. Jerman-Blažič, and T. Klobučar, Pervasive Systems: Enhancing Trust Negotiation with Privacy Support. Boston, MA: Springer US, 2007, ch. 2, pp. 23–38. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-71058-7_2 [retrieved: June, 2012].

[2] K. E. Seamons, M. Winslett, and T. Yu, "Limiting the Disclosure of Access Control Policies during Automated Trust Negotiation," Proc. symposium on network and distributed systems security, NDSS, pp. 1-11, 2001.

[3] W. Chen, L. Clarke, J. Kurose, and D. Towsley, "Optimizing cost-sensitive trust-negotiation protocols", Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Vol. 2, pp. 1431-1442, 2005.

[4] Self Orchestrating CommunIty ambiEnT IntelligEnce Spaces (SOCIETIES), EU FP7 project, Information and Communication Technologies, Grant Agreement Number 257493.

[5] X.509 standard recommendation, http://www.itu.int/rec/T-REC-X.509/en [retrieved April, 2012].

[6] XML-DSig, XML Signature Syntax and Processing, 2nd Edition http://www.w3.org/TR/xmldsig-core/, [retrieved April, 2012].

[7] Canonical XML 1.1, W3C recommendation, http://www.w3.org/TR/xml-c14n11/, [retrieved April, 2012].

[8] B. Dodson and M. S. Lam, "Micro-Interactions with NFC-Enabled Mobile Phones",

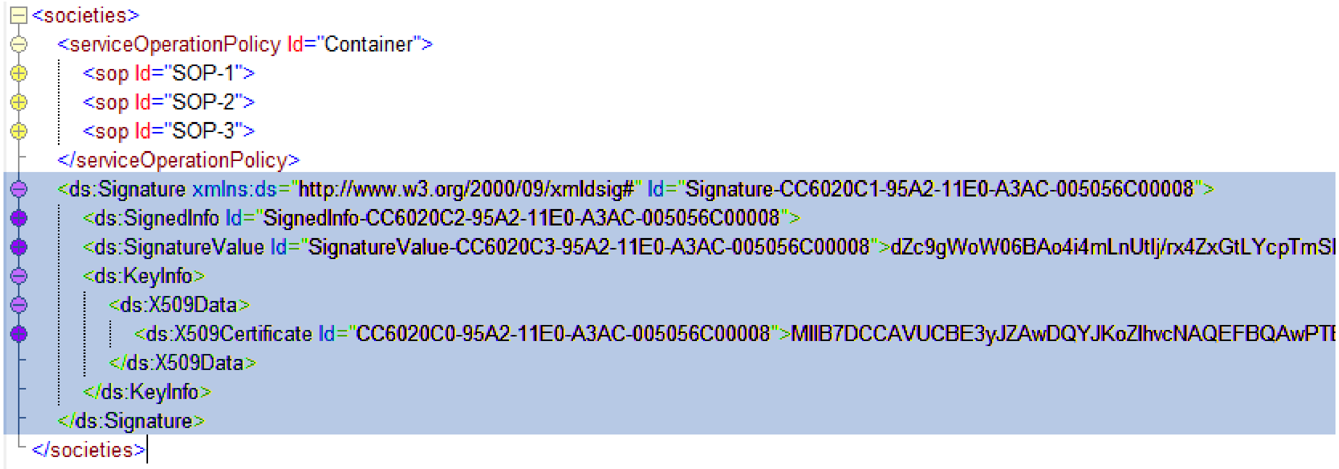
http://mobisocial.stanford.edu/papers/mobicase11.pdf, [retrieved April, 2012].

[9] B. Dodson, H. Bojinov, and M. S. Lam, "Touch and Run with Near Field Communication (NFC)", http://mobisocial.stanford.edu/papers/nfc.pdf, [retrieved April, 2012].

[10] S. Gallacher, E. Papadopoulou, N. K. Taylor, I. Roussaki, N. Kalatzis, N. Liampotis, F. R. Blackmun, M. H. Williams, and D. Zhang, "Personalisation in a system combining pervasiveness and social networking," in 2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN). IEEE, Jul. 2011, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/ICCCN.2011.6005900 [retrieved April, 2012].

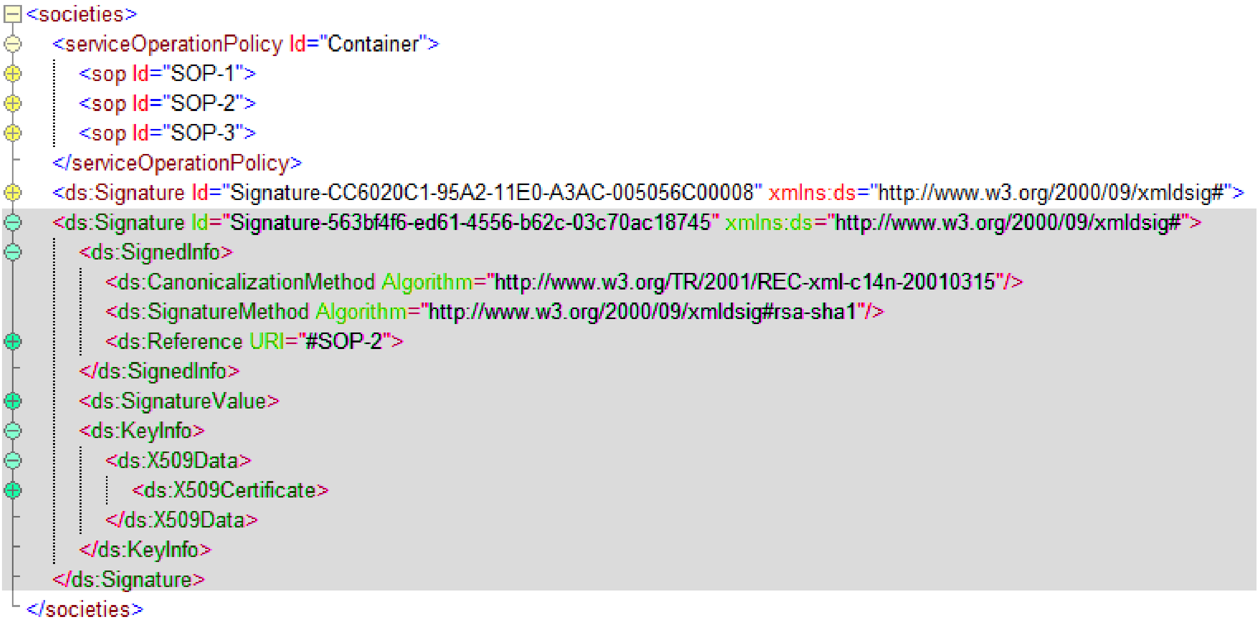
[11] P. Bonatti, S. De Capitani di Vimercati, and P. Samarati, "An Algebra for Composing Access Control Policies," in ACM Transactions on Information and System Security (TISSEC), vol. 5, no. 1, pp. 1-35, February, 2002.

[12] C. Sierra, P. Faratin, and N. R. Jennings, "A service-oriented negotiation model between autonomous agents," in Lecture Notes in Computer Science, Volume 1237/1997, pp. 17-35, 1997.



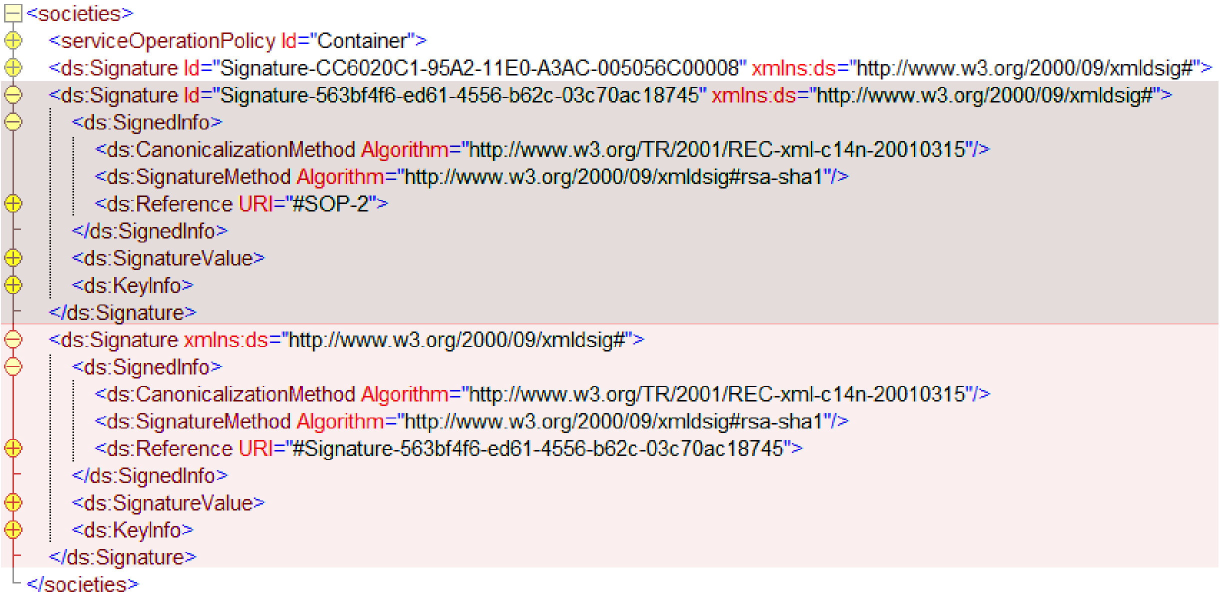
```
<societies>
  <serviceOperationPolicy Id="Container">
    <sop Id="SOP-1">
    <sop Id="SOP-2">
    <sop Id="SOP-3">
  </serviceOperationPolicy>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="Signature-CC6020C1-95A2-11E0-A3AC-005056C00008">
    <ds:SignedInfo Id="SignedInfo-CC6020C2-95A2-11E0-A3AC-005056C00008">
    <ds:SignatureValue Id="SignatureValue-CC6020C3-95A2-11E0-A3AC-005056C00008">dZc9gWoW06BAo4i4mLnUtIj/rx4ZxGtLYcpTmSI
    <ds:KeyInfo>
      <ds:X509Data>
        <ds:X509Certificate Id="CC6020C0-95A2-11E0-A3AC-005056C00008">MIIB7DCCAVUCBE3yJZAwDQYJKoZIhvcNAQEFBQAwPTE
      </ds:X509Data>
    </ds:KeyInfo>
  </ds:Signature>
</societies>
```

Figure 7: Signature (highlighted) of the initial policy offer. XML nodes for policy options are collapsed.



```
<societies>
  <serviceOperationPolicy Id="Container">
    <sop Id="SOP-1">
    <sop Id="SOP-2">
    <sop Id="SOP-3">
  </serviceOperationPolicy>
  <ds:Signature Id="Signature-CC6020C1-95A2-11E0-A3AC-005056C00008" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:Signature Id="Signature-563bf4f6-ed61-4556-b62c-03c70ac18745" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <ds:Reference URI="#SOP-2">
    </ds:SignedInfo>
    <ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data>
        <ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
  </ds:Signature>
</societies>
```

Figure 8: Policy during negotiation after the requester has chosen option "SOP-2" and signed it



```
<societies>
  <serviceOperationPolicy Id="Container">
  <ds:Signature Id="Signature-CC6020C1-95A2-11E0-A3AC-005056C00008" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:Signature Id="Signature-563bf4f6-ed61-4556-b62c-03c70ac18745" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <ds:Reference URI="#SOP-2">
    </ds:SignedInfo>
    <ds:SignatureValue>
    <ds:KeyInfo>
  </ds:Signature>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <ds:Reference URI="#Signature-563bf4f6-ed61-4556-b62c-03c70ac18745">
    </ds:SignedInfo>
    <ds:SignatureValue>
    <ds:KeyInfo>
  </ds:Signature>
</societies>
```

Figure 9: Final policy with all three signatures. The first signature is made by provider and is shown collapsed. The second signature is made by the requester. They both reference the policy. The last signature is the provider's one and references the requester's signature.