Security Personalization for Internet and Web Services

Yee, George; Korba, Larry

National Research Council Canada
Institute for Information Technology

Conseil national de recherches Canada
Institut de technologie de l'information

NRC·CNRC

*Security Personalization for Internet and Web Services**

Yee, G., and Korba, L.
2007

* International Journal of Web Services (IJWSR). 2007. NRC 49358.

Security Personalization for Internet and Web Services (Revision from P10172)

George Yee and Larry Korba
Institute for Information Technology
National Research Council Canada
{George.Yee, Larry.Korba}@nrc-cnrc.gc.ca

ABSTRACT:

The growth of the Internet has been accompanied by the growth of Internet services (e.g. e-commerce, e-health). This proliferation of services and the increasing attacks on them by malicious individuals have highlighted the need for service security. The security requirements of an Internet or Web service may be specified in a security policy. The provider of the service is then responsible for implementing the security measures contained in the policy. However, a service customer or consumer may have security preferences that are not reflected in the provider's security policy. In order for service providers to attract and retain customers, as well as reach a wider market, a way of personalizing a security policy to a particular customer is needed. We derive the content of an Internet or Web service security policy and propose a flexible security personalization approach that will allow an Internet or Web service provider and customer to negotiate to an agreed-upon personalized security policy. In addition, we present two application examples of security policy personalization, and overview the design of our security personalization prototype.

KEY WORDS:

Security, Security Policy, Personalization, Negotiation, Internet Services, Web Services

INTRODUCTION

The term "Internet service" is used here to mean any electronic service that is accessed using the Internet, e.g. electronic banking. In the following, we use "Internet service" to refer to all electronic services that are available through the Internet, including Web Services that are based on the Service Oriented Architecture. We use "Web Service" when we wish to indicate that we are treating Web Services in particular.

A large number of Internet services targeting consumers has accompanied the rapid growth of the Internet. Internet services are available for banking, shopping, learning, healthcare, and Government Online, to name a few. However, these services are subject to malicious attack in one form or another. This leads to concerns over their security (Joshi, Aref, Ghafoor, & Spafford, 2001).

In order for Internet services to be successful, they must be secured from malicious individuals who constantly try to compromise them. An effective and flexible way of managing security for these services is to make use of security policies. An Internet service security policy is a specification of what security measures will be used to protect the service from security attacks.

A security policy by itself does not guarantee that its stated security measures will be put in place or be complied with. That is an area of policy compliance that is outside the scope of this paper.

An Internet service provider makes use of a security policy to specify the security measures that it has put or will put in place to protect its services. However, this security policy may not match up with the security preferences of a customer or consumer (we use "user", "customer" and "consumer" interchangeably) of the services. For example, suppose the security measure is user authentication by the use of a password. This authentication approach is known to be insecure. A security-sensitive consumer such as, for example, a defense contractor, may wish to add biometric authentication. Unless the user authentication is changed to include biometrics, the defense contractor would not be able to make use of the service. As another example, suppose the security measure is access control. The provider's security policy may provide access to 5 features of a service, whereas a particular customer may need access to only 3 features. In this case, the customer may be reluctant to make use of this provider's service, especially if the customer can find another provider that only offers the features needed and at a lower price. One solution to these mismatches of a provider's security policy with a customer's security preferences is to allow the customer to personalize the security policy by negotiating with the provider regarding the security measures that are in the provider's security policy. We call this negotiation process *security policy personalization*, i.e. the provider's security policy becomes personalized to a particular customer through negotiation.

This paper extends Yee & Korba (2005b) by a) providing new details on the "scheme for online help in making offers" during the negotiation process, b) providing new details on how the approach can be implemented for Web Services, c) giving a more complete description of the prototype, d) adding example applications, e) enlarging the section on related works, f) including an evaluation of the proposed approach for security policy personalization, and g) improving the clarity of the writing in all sections.

The objectives and contributions of this paper are to a) introduce the need for personalization of provider service security policies, b) derive a security policy template suitable for use with Internet services, c) present an approach for consumer-provider negotiation that accomplishes this personalization, including a novel method of providing help during negotiation, d) show how security policy personalization can be implemented for Web Services, e) give example applications of security policy personalization, f) describe our prototype for security policy negotiation, and g) evaluate this work and discuss related works. Note that our security policy template is only an example template, since it may change depending on future security requirements as well as available security technology.

The remainder of this paper is organized as follows. The next section defines Internet services and derives requirements for security policies and their negotiation. Section "SECURITY POLICY NEGOTIATION" derives an Internet service security policy template, presents our approach for Internet services security policy personalization using negotiation, and shows how this approach can be implemented for Web Services. Section "APPLICATION EXAMPLES" describes two example applications of security policy personalization. Section "PROTOTYPE FOR SECURITY POLICY NEGOTIATION" gives an overview of our prototype. Section "RELATED WORK" examines the literature for related work. Section "EVALUATION" discusses the applicability and effectiveness of our personalization approach. We end with "CONCLUSIONS AND FUTURE RESEARCH".

INTERNET SERVICES, REQUIREMENTS FOR SECURITY POLICIES AND THEIR NEGOTIATION

In this section, we begin by defining an Internet service. We then describe requirements for security policies and security policy negotiation.

Internet Services

An Internet service for the purposes of this paper is characterized by the following attributes:

- The service is performed by application software (service software) that is owned by a provider (usually a company); the service is accessible across the Internet.
- The provider's service software can make use of the service software of other providers in order to perform its service.
- The provider has a security policy that specifies what security measures it will use to secure the service.
- The service may require the use of the consumer's private information, in which case it should also have a privacy policy that states what private information it requires and how it will make use of the private information.
- The service is consumed by a person or another application accessing the service across the Internet.
- The consumer has security and privacy preferences for the service that may not be reflected in the provider's security and privacy policies respectively.
- There is usually a fee that the consumer pays the provider for use of the service.

Thus, an Internet service includes all electronic services that are accessible via the Internet, including Web Services. These services may differ in the way they are implemented but our approach applies to all of them. Two classes of Internet services that differ in implementation are: a) client-server type services where consumers (clients) access a service website (server) with the service software running at the backend, and b) Web Services that are based on the Service Oriented Architecture (O'Neill, 2003) and use protocols based on XML (World Wide Web Consortium). Examples of current Internet services are Amazon.com (online retailer), optionsxpress.com (online stockbroker), and WebMD.com (health information and technology solutions provider).

Security Policy Requirements

Requirements for Internet services security policies address what security measures should be covered in the policy. Since Internet services fall under the category of open systems, we begin by looking at requirements prescribed by ISO 7498-2, the reference model for security architectures by the International Organization for Standardization (International Organization for Standardization).

This standard identifies 5 main categories of security services, as follows:

1. Authentication
2. Access Control
3. Data Confidentiality
4. Data Integrity
5. Non-repudiation

The International Telecommunication Union Telecommunication Standardization Sector (ITU-T) provides Recommendation X.800, Security Architecture for OSI (Open Systems Interconnection) (ITU-T) that lists the same 5 main categories of security services as above. We propose that these 5 categories of security services be covered in an Internet security policy since they are part of standards. We would add the following security services:

6. Secure Logging – of user transactions by the provider
7. Certification – user or provider would use some certifying authority to certify credentials
8. Malware Detection – user or provider would use some anti-malware software to detect and eliminate malware from their computing platforms
9. Application Monitoring – user platform monitoring for licensed, verified, and permitted applications (results only reported to the user)

Security services 6 to 9 are added to enhance security required in the Internet services environment. Secure logs assist in the verification of policy compliance. Certification is required for credentials such as certificates exchanged across the Internet. Malware detection is a definite requirement in today's Internet-connected computing platforms that are open to all sorts of virus and Trojan horse attacks. Application monitoring protects the user from the inadvertent use of unlicensed, illegal, or malicious programs, which may be loaded onto the user's platform via the Internet by attackers. We thus have 9 security services that should be specified in an Internet service security policy. Figure 1 identifies where these security services are typically applied using an Internet service network view. In Figure 1, the Certification Authority is typically a Certificate Authority as used for PKI (Public Key Infrastructure). The double arrows represent two-way communication channels, and the dashed line represents logical traversal of the Internet, i.e. the actual traversal, possibly using diverse physical links, is not shown. The storage of the consumer's private information is identified explicitly as requiring confidentiality. The provider may make use of other service providers in the provision of its service but this aspect is not shown.

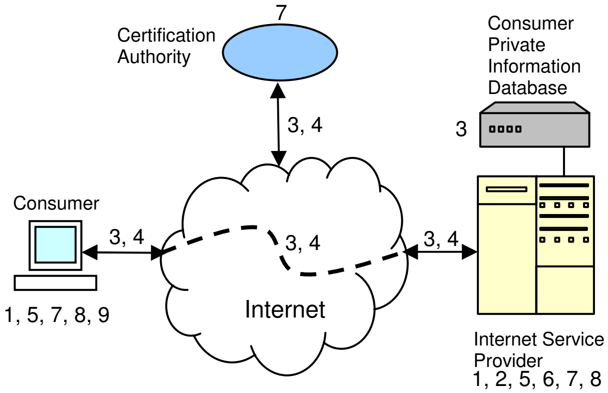


Figure 1. Application of security services (numbers correspond to security services listed above)

The above standards also list specific security services under the main security service categories. As an example, non-repudiation has the specific services (with the obvious meanings): "Non-repudiation, Origin" and "Non-repudiation, Destination". As well, security mechanisms (e.g. digital signature) are used to support or implement security services. We will employ specific services and security mechanisms to formulate our Internet services security policy template in the next section.

Security Policy Negotiation Requirements

Based on the Internet service environment (i.e. providers providing services to consumers across the Internet), and negotiation processes in general, we propose the following requirements for Internet services security policy negotiation:

1. The security services and mechanisms to be negotiated must be clear and understandable.
2. The consumer may negotiate any subset of security services and mechanisms in the policy.
3. There needs to be some form of trusted online help for the consumer in cases where it is difficult to know what choice to make in a particular step in the negotiation.
4. The consumer normally initiates negotiation after finding the service that she (note: we use “she” and “her” to stand for both sexes) wants to use. However, when a provider changes its service and requires new security levels, it may initiate a new security policy negotiation with the consumer.
5. Negotiation may be terminated by either the consumer or the provider, at any step in the negotiation prior to a successful outcome. If so terminated, the associated service may not proceed.
6. The user interface for the negotiation must be easy to use, intuitive, and trustable (i.e. give the user a sense of ease that everything is working as stated or planned).

Requirement 3 is needed in order that the negotiation is not blocked simply due to the fact that the consumer does not know what security choice to make. This can occur quite easily where the consumer is not knowledgeable about security resources. We will propose a way for achieving this requirement in the next section.

SECURITY POLICY NEGOTIATION

In this section, we first discuss the goals of security policy negotiation. We then define an Internet service security policy template according to the above security policy requirements. Finally, we present an approach for security policy negotiation that satisfies the above negotiation requirements.

Goals of Security Policy Negotiation

The consumer’s objectives or goals for security policy negotiation differ from the provider’s goals for such negotiation. In the absence of security policy personalization, the provider is concerned with protecting its systems (e.g. servers) and ensuring that there is just sufficient security to comply with laws and what is needed for the general working of the service (e.g. authentication, secure communication channel). Providing extra security beyond this level would take away from profitability, so the provider in all likelihood will not do so. However, in the presence of security policy negotiation, the provider must additionally comply with the customer’s security requirements or face losing the customer to another provider. This does not necessarily mean that the provider’s security costs will be higher since the consumer may require less security for special situations (e.g. consumer trades off security for performance where the service is supplied through a mobile device of limited power and where the lower security level is still adequate for the application). The consumer’s goals for negotiation are to have the provider put in place the security measures that the consumer requires for her *personal* use of the service.

These requirements may depend on the following aspects of the consumer's use of the service: service device (e.g. mobile or connected to physical lines), how the service is used (e.g. e-learning with highly sensitive information or e-learning with public information), and even environment (e.g. consumer's neighborhood has hackers that delight in breaking into Wi-Fi networks). Our application examples of security policy personalization (see below) will illustrate some of these consumer goals.

Internet Service Security Policy

Based on the requirements of Subsection "Security Policy Requirements", and using example values and security mechanisms, we propose the Internet service security policy template shown in Table 1.

Table 1. Internet Service security policy template

Policy Use: **E-learning**	***Owner:*** **Learners Online, Inc.**	***Valid:*** **unlimited**
CONSUMER PROVISIONS	**PROVIDER PROVISIONS**	**PROVIDER PROVISIONS**
Consumer Authentication *Implement:* yes (default) *Mechanism:* password *Mechanism:* V+F biometrics	**Provider Authentication** *Implement:* yes (default) *Mechanism:* security token *Mechanism:* digital signature	**Communication Confidentiality** *Implement:* yes (default) *Mechanism:* SSL
Consumer Non-Repudiation *Implement:* yes (default) *Mechanism:* digital signature	**Provider Non-Repudiation** *Implement:* yes (default) *Mechanism:* digital signature	**Communication Integrity** *Implement:* yes (default) *Mechanism:* MD5 Hash
Consumer Certification *Implement:* yes (default) *Mechanism:* certificate	**Provider Certification** *Implement:* yes (default) *Mechanism:* certificate	**Secure Logging** *What:* order transactions *Mechanism:* 3DES encrypt *What:* user input *Mechanism:* 3DES encrypt
Consumer Malware Detect *Implement:* yes (default) *Mechanism:* Norton	**Provider Malware Detect** *Implement:* yes (default) *Mechanism:* Norton	**Access Control** *User Role:* Secretary *Resource:* scheduling module *Resource:* admin module *User Role:* President *Resource:* admin module *Resource:* salary module
Application Monitoring *Implement:* yes (default) *Mechanism:* IIT-ISG	**Data Store Confidentiality** *Implement:* yes (default) *Mechanism:* 3DES encrypt	

In Table 1, the top shaded portion is the policy header. The header contains the following administrative fields: *policy use* identifies for which service the policy is provided, *owner* identifies the name of the provider of the service, and *valid* specifies the end date after which the policy is no longer valid, or "initial/continuing" which indicates whether or not the security policy is enforced only initially or continuously. The figure also shows that some security services can have multiple mechanisms (e.g. consumer authentication using password and biometrics). In such cases, the additional mechanisms can simply be listed under the security service. Similarly, secure logging and access control can have additional items (e.g. access control can have additional resources under each role). Note that for most services, security policy negotiation would involve the selection of a particular mechanism. However, consumers can also select a set of mechanisms where the consumer either cannot decide or it does not matter to her which mechanism from the set will be implemented. In that case, the provider chooses which mechanism from the set to implement. The security policy outcome of a negotiation that chooses sets of security mechanisms instead of single mechanisms would look like Table 1.

Security Policy Negotiation

We propose that security policy negotiation be the first of two stages of negotiation, the second stage being privacy policy negotiation. Privacy policy negotiation is fully described in Yee & Korba (2003a) and Yee & Korba (2003b); it is outside the scope of this paper. Security policy negotiation is entered once the consumer has determined which service she wants to use. Privacy policy negotiation is entered only if security policy negotiation is successful. The service can only be activated if both stages of negotiation are successful. Where negotiation is not needed due to a match found between the provider's policy and the consumer's preferences, this result still signals a successful negotiation. Where a negotiation is unsuccessful, the consumer needs to look for another Internet service to try (or find ways to match the security requirements of the service but it is probably easier to just find another service). Figure 2 gives a flowchart of this process, where each box is only carried out if all the boxes above it are successful. Otherwise, the control flow returns to "start".

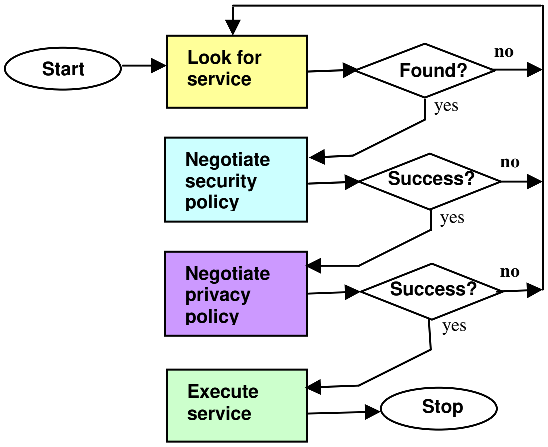


Figure 2. Negotiation Stages prior to service execution

In this work, (see Figure 3), a non-autonomous software agent acts on behalf of the consumer to receive/send negotiation messages from/to the provider. Another non-autonomous agent serves the provider in the same way. These agents also perform validation checks on the information to be sent. It is probably feasible to use autonomous agents to automate our form of security policy negotiation, but this is for future work.

Once the consumer has determined the service she wants to use, the security policy negotiation proceeds as follows (assuming a consumer-initiated negotiation):

1. The consumer requests the provider's security policy from the PA.
2. The consumer compares the provider's SP with her own security preferences to see if there is a "match". A "match" can occur for either a single security mechanism or for a set of security mechanisms and means that the consumer's preferred security mechanism(s) is (are) identical to the mechanism(s) in the provider's SP. If there is a match, the CA signals a "successful negotiation" and the processing proceeds to privacy

negotiation. If there is no match, consumer and provider begin security policy negotiation (step 3).

3. The consumer changes the provider's SP according to her preferences (i.e. formulate or make an offer) and sends it back (via the CA) to the provider. The provider either accepts the new SP or changes it according to what it can accept. The provider then sends it back (via the PA) to the consumer. The consumer looks at it again and makes further changes (i.e. formulate or make a new offer) and sends it back (via the CA) to the provider. This negotiation process continues back and forth until a) both sides agree and the negotiation is successful or b) one side terminates the negotiation (after concluding that no progress can be made) and the negotiation is unsuccessful. If the negotiation is unsuccessful, the consumer searches for another service to try (or tries to satisfy the provider's security requirements).

Figure 4 illustrates these steps using a message sequence chart for a consumer initiated negotiation (a provider initiated one would replace the top two arrows with one arrow from provider to consumer representing a request for negotiation together with the provider's SP). In Figure 4, SP1 is the consumer's first offer, SP2 is the provider's counter-offer, SP3 is the consumer's counter-counter offer and so on. After n steps the negotiation is successful, since the provider returns SPn, the consumer's last offer, unchanged.

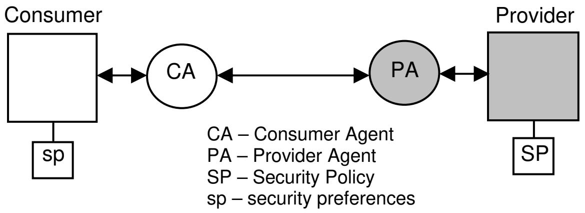


Figure 3. Security policy negotiation entities

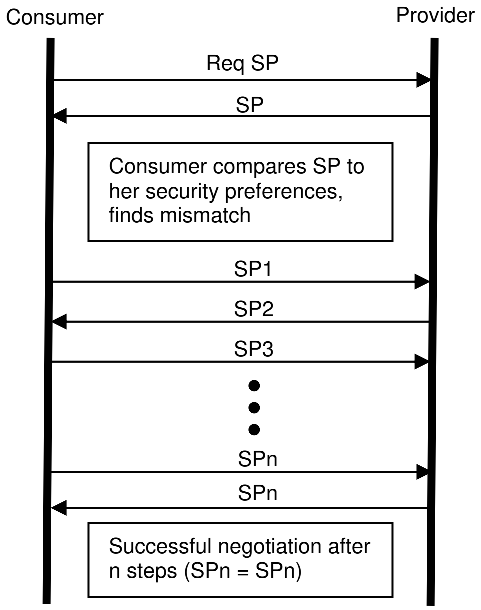

Consumer
Provider
Req SP
SP
Consumer compares SP to her security preferences, finds mismatch
SP1
SP2
SP3
SPn
SPn
Successful negotiation after n steps (SPn = SPn)

Figure 4. Security policy negotiation steps

Satisfying the Negotiation Requirements

We now examine the negotiation requirements of Subsection "Security Policy Negotiation Requirements" to see how they can be fulfilled. Requirement 1 will be fulfilled in our prototype using online help in the form of pop-up windows that explain the particular security service for which help was requested. Requirement 2 is fulfilled by the consumer's ability to change any subset of security measures in the policy. Requirement 3 is addressed below. Requirements 4 and 5 are already part of our negotiation procedure. Requirement 6 will be fulfilled in our prototype by an appropriate interface design. We will describe this interface in Section "PROTOTYPE FOR SECURITY POLICY NEGOTIATION".

Scheme for Online Help in Making Offers

Negotiation requirement 3, the provision of trusted online help for the consumer to formulate a particular offer (i.e. change the provider's policy to reflect her security preferences) is fulfilled using the knowledge of what others selected under the same circumstances. This knowledge is acquired through the following steps:

1. Each provider stores the security policies that have been used with its services, identifying the services for which they were used and the dates they were used but not identifying the consumers with whom they were used (to preserve privacy). A services authority (SA) periodically collects these security policies from all providers, along with the types of Internet services to which they were applied, the dates they were applied, and the name of the provider that applied them.

2. Over a moving period *P* of the last *p* months (e.g. *p=12*), the SA constructs the following Security Score Table (Table 2), using the security services and mechanisms from each security policy within *P*, together with security violation and impact data (from providers, discussed below) corresponding to these security services and mechanisms:

   Table 2. Security score table

Pro-vider	Internet Service (S)	Security Policy (SP)	Security Service (SS)	Security Mecha-nism (SM)	No. of Security Violation s *(k)*	Average Impact per Violation *(v)*	Security Score *(s = kv)*
P_1	S_1	SP_1	SS_1	SM_1	2	3	6
P_1	S_1	SP_1	SS_1	SM_2	3	1	3
P_1	S_1	SP_1	SS_1	SM_3	2	2	4
P_1	S_1	SP_2	SS_1	SM_1	1	4	4
P_1	S_1	SP_2	SS_1	SM_4	8	2	16
P_1	S_1	SP_2	SS_1	SM_5	5	2	10

   Providers are asked or required (through legislation, e.g. in some jurisdictions, hospitals are required to report patients that have been shot to police) to report the number and nature of security violations over period *P,* along with the number of consumers affected by each violation to the SA. The latter then assigns an impact number to each violation using a scale of 1 to 5, corresponding to lowest and highest impact respectively, based on the number of consumers affected and the nature of the violation. For example, a consumer failure to use Norton software to detect malware (see Table 1) involving 3 consumers may be assigned a low impact score (e.g. 2). On the other hand, a consumer authentication violation arising from password abuse involving 200 consumers may be assigned a high impact score (e.g. 4).

3. In the course of a security policy negotiation, a consumer who needs help in making a security choice (e.g. a security service or a security mechanism) can request from the SA the security services and mechanisms matching the service, the provider, and a security

score below a certain threshold (clearly, the lower the security score, the more effective the corresponding security mechanism and security service). This information can be displayed as the number of security policies making use of the qualifying security mechanisms and associated security services. For example, from the above table, for P_1, S_1, threshold 7, the qualifying mechanisms are SM_1, SM_2, and SM_3 for SS_1 in SP_1 as well as SM_1 for SS_1 in SP_2. This information can be displayed as $SS_1(2)$, $SM_1(2)$, $SM_2(1)$, $SM_3(1)$ signifying that for P_1, S_1, threshold 7: a) SS_1 has mechanisms below the threshold and was used in 2 security policies, b) SM_1 of SS_1 is below the threshold and was used in 2 security policies, c) SM_2 of SS_1 is below the threshold and was used in 1 security policy, and finally d) SM_3 of SS_1 is below the threshold and was used in 1 security policy. The consumer would then use this frequency of utilization information to guide her choice of security services and mechanisms during negotiation. We will illustrate this step further in describing our prototype below.

Figure 5 illustrates this scheme. In Figure 5, providers provide the SA with the inputs mentioned above. The SA then computes the security score tables and stores them in a database. The consumer can then use standard database queries to retrieve the assistance needed.

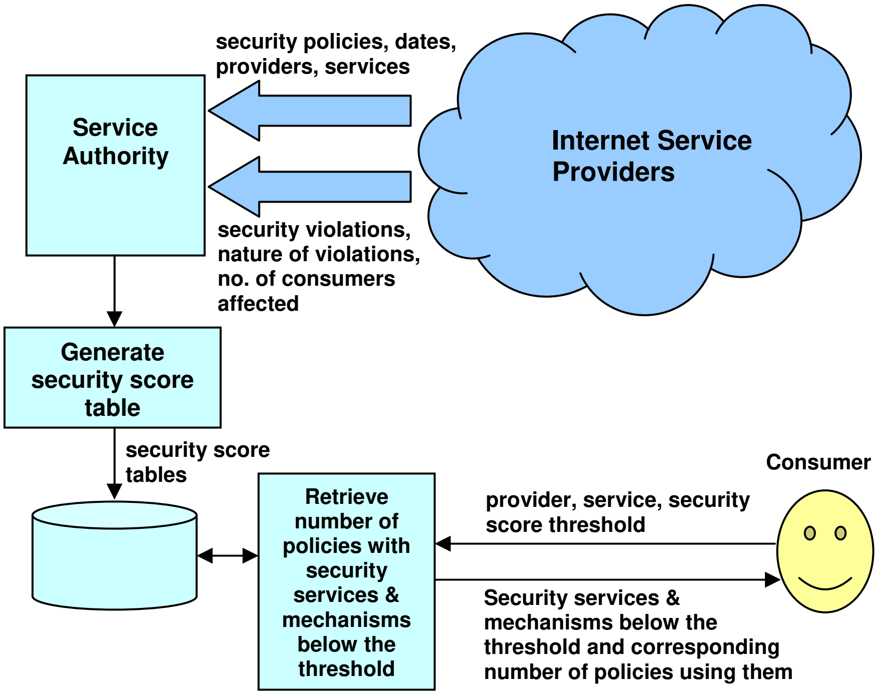

security policies, dates, providers, services
Service Authority
Internet Service Providers
security violations, nature of violations, no. of consumers affected
Generate security score table
security score tables
Consumer
Retrieve number of policies with security services & mechanisms below the threshold
provider, service, security score threshold
Security services & mechanisms below the threshold and corresponding number of policies using them

Figure 5. Scheme for online help in making offers in security policy negotiation

Implementation for Web Services

Web Services operate within a Service-Oriented Architecture (SOA) which uses XML, UDDI, SOAP, and WSDL to publish a service, find a service, and bind to a service (O'Neill, 2003). In this scenario, a consumer wishing to execute a particular service would first find details of the provider and the services offered by the provider in the UDDI Web Services directory. (Providers would have previously populated the UDDI directory with their names and details of the services they offer.) Once the consumer has sufficient information about the service, including service key and binding information, the consumer formulates a SOAP message to send to the provider to execute the service. It is here where our negotiation stages can be inserted. The initial SOAP message to the provider would not be to execute the service but to request the provider's security policy to begin the negotiation sequence. Only after the privacy policy negotiation is successful (with the negotiation stages described above in Subsection "Security Policy Negotiation") would the SOAP message to execute the service be sent. Where a negotiation fails, the consumer could access the UDDI directory again to find another provider and start the negotiation stages all over again (or find ways to satisfy the first provider's security policy). Figure 6 illustrates the implementation of security policy negotiation for Web Services using three state machines representing the consumer's Web Services client (running on the consumer's computer), the Web Service provider, and the UDDI directory. The transition arrows in Figure 6 are labeled using the convention "condition / action" where "?" means "received" and "!" means "send". For example, "? service request / ! provider query to UDDI", has the condition "received service request" and the action "send provider query to UDDI".

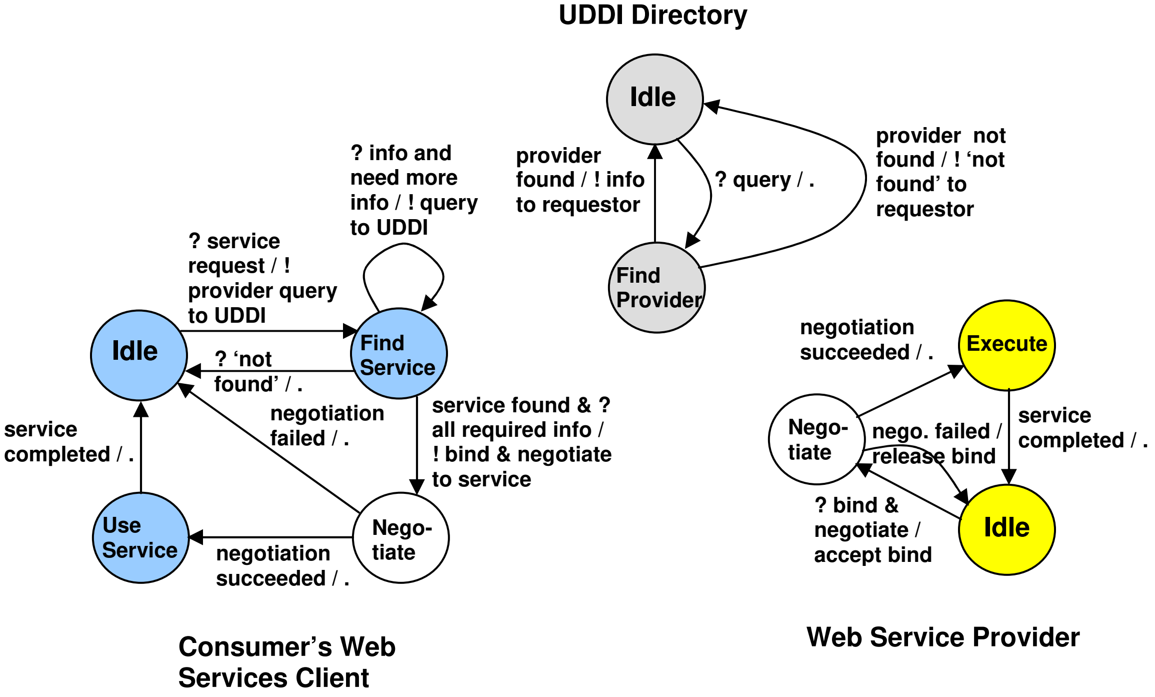

UDDI Directory
Idle
provider found / ! info to requestor
? query / .
provider not found / ! 'not found' to requestor
Find Provider
? info and need more info / ! query to UDDI
? service request / ! provider query to UDDI
Idle
Find Service
? 'not found' / .
negotiation failed / .
service found & ? all required info / ! bind & negotiate to service
service completed / .
Use Service
negotiation succeeded / .
Nego-tiate
Consumer's Web Services Client
negotiation succeeded / .
Execute
Nego-tiate
nego. failed / release bind
service completed / .
? bind & negotiate / accept bind
Idle
Web Service Provider

Figure 6. Implementation of security policy negotiation for Web Services

We explain Figure 6 by describing the execution flow, starting with the Consumer's Web Services Client (CWSC), the UDDI Directory (UD), and the Web Service Provider (WSP) all in the IDLE state. The CWSC moves from the IDLE state to the FIND SERVICE state after receiving a service request from the consumer and sending a query (using XML) to the UD for services offered by a particular service provider. The consumer may have learned about the provider from the Internet, prior to requesting the service. Upon receiving the query from the CWSC, the UD moves from the IDLE state to the FIND PROVIDER state, in which the UD searches its database to find the provider and its service offerings. Once found, the UD sends this information (using XML) to the CWSC and transitions back to the IDLE state. If the provider is not found, the UD sends a "not found" message to the CWSC and moves back to the IDLE state. While in the FIND SERVICE state, if the CWSC receives a "not found" message from the UD, it transitions back to the IDLE state where the consumer would need to request the service from another provider and the CWSC would start again. If the CWSC receives the requested provider's service offerings, the available services are presented to the consumer who then chooses a desired service. The CWSC then queries the UD for information related to the specific service chosen, such as service key and binding information (modeled by the transition back into itself). Once the CWSC receives this information it sends another query to the UD to get the WSDL description of the service (again modeled by the transition back into itself). Upon receiving this WSDL description, the CWSC formulates and sends a SOAP message to the provider and moves to the NEGOTIATE state. The SOAP message binds to the desired service and requests security policy negotiation. Once this SOAP message is received by the WSP, it accepts the binding and also moves to the NEGOTIATE state. Within the NEGOTIATE state, the CWSC and the WSP carry out security policy negotiation as described above (see Figure 4). If this negotiation is successful, the CWSC transitions to the USE SERVICE state and the WSP moves to the EXECUTE state where the consumer uses the service (any service specific parameters not shown). From these states, both the CWSC and the WSP move back to the IDLE state once the service is completed. If the negotiation is unsuccessful, the CWSC and the WSP both move back to the IDLE state. In so doing, the WSP releases the bind. In the IDLE state after an unsuccessful negotiation, the consumer can request the same service again but from a different provider (we have not modeled the case where the consumer tries to satisfy the provider's security policy after an unsuccessful negotiation).

Given the above, the implementation of security policy negotiation for Web Services would involve writing Web Services software to implement the NEGOTIATE state in Figure 6 (i.e. implementing Figure 4) with appropriate user interfaces as well as interfaces to the adjacent states, since Web Services software for the rest of the states and state machines already exist. In addition, XML-based policy languages would be needed to express security policies so that they may be machine processed, for policy creation, editing, and compliance checking. We examine this aspect next.

Web Services already possess XML-based language specifications to implement security policies. These specifications are generally worked on by a consortium of companies and then submitted to OASIS (Organization for the Advancement of Structured Information Standards) for standardization. We may use WS-Policy (Bajaj et al., 2006a) and WS-SecurityPolicy (Della-Libera et al., 2005) to express web service security policies (example given below). WS-Policy may be applied to express security requirements for web services in general. WS-SecurityPolicy contains the policy elements (security assertions) applicable to WS-Security (OASIS, 2006). WS-Security provides security enhancements for SOAP messaging to ensure message integrity and confidentiality. In addition, we would need WS-PolicyAttachment (Bajaj et al., 2006b) to define how policies are discovered or attached to a Web Service. WS-PolicyAttachment specifies mechanisms for associating a policy with arbitrary XML elements, WSDL artifacts, and UDDI elements. At the time of this writing, WS-Policy, WS-SecurityPolicy, and WS-PolicyAttachment are all draft specifications from a number of companies, including IBM and Microsoft, that are waiting to be standardized. WS-Security became an OASIS standard in February 2006.

WS-Policy, also known as the Web Services Policy Framework, provides a general purpose model with corresponding syntax to specify the policies of a Web Service. It does this by defining a basic set of constructs that can be used and extended by other Web Services specifications to specify a broad range of requirements and capabilities for services. Web Services specifications (WS*) are in fact designed to be inter-composable. WS-Policy is often composed with WS-SecurityPolicy (see example below). In addition, WS-Policy should be regarded as a building block that can be used together with other Web Services and application specific protocols (such as the one defined by Figure 4) to provide a negotiation solution for Web Services. An example of this building block aspect is the use of WS-Policy in conjunction with WS-PolicyAttachment as mentioned above.

WS-Policy specifies a policy as a collection of policy alternatives, where each policy alternative is a set of policy assertions. Listing 1 gives an example WS-Policy specification of the authentication portion of the security policy template in Table 1. This example uses WS-SecurityPolicy to define consumer authentication (lines 04-11) and provider authentication (lines 12-19) where it is assumed that the alternatives (lines 07-08 and lines 15-16) are assertions defined in WS-SecurityPolicy (assumed for illustrative purposes only, this is not currently the case). Note that the "ExactlyOne" operator requires that only one of the encapsulated alternatives (e.g. lines 07-08) can be implemented. A valid interpretation of Listing 1 is that an invocation of the Web Service to which this policy corresponds requires that one and only one of the alternatives in lines 07-08 be implemented for consumer authentication, and one and only one of the alternatives in lines 15-16 be implemented for provider authentication.

Listing 1. Example WS-Policy specification of authentication

```
(01) <wsp:Policy
(02)  xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy"
(03)  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy" >
(04)        <sp:ConsumerAuthentication>
(05)              <wsp:Policy>
(06)                    <wsp:ExactlyOne>
(07)                          <sp:password />
(08)                          <sp:V+F_biometrics />
(09)                    </wsp:ExactlyOne>
(10)              </wsp:Policy>
(11)        </sp:ConsumerAuthentication>
(12)        <sp:ProviderAuthentication>
(13)              <wsp:Policy>
(14)                    <wsp:ExactlyOne>
(15)                          <sp:securitytoken />
(16)                          <sp:digital_signature />
(17)                    </wsp:ExactlyOne>
(18)              </wsp:Policy>
(19)        </sp:ProviderAuthentication>
(20) </wsp:Policy>
```

APPLICATION EXAMPLES

In this section, we present two example applications of security policy personalization. The first example concerns a customer using a mobile device to access a stock quotation and order entry service called Stocks Unlimited. The second example describes an e-learning service called Easy Learn that targets a wide range of clients with different security preferences. The first example applies security personalization to accommodate personal preferences and the operational environment. The second example looks at personalization to accommodate personal security preferences. These two examples show that security personalization is a good solution to meeting diverse security needs that can arise from today's technological society.

Stocks Unlimited

Stocks Unlimited is an Internet service accessible using a mobile device such as a cell phone or wireless PDA. Figure 7 shows a network view of this service. In this figure, the mobile ISP (Internet Service Provider) provides mobile wireless access to the Internet. Stocks Unlimited provides the actual service.

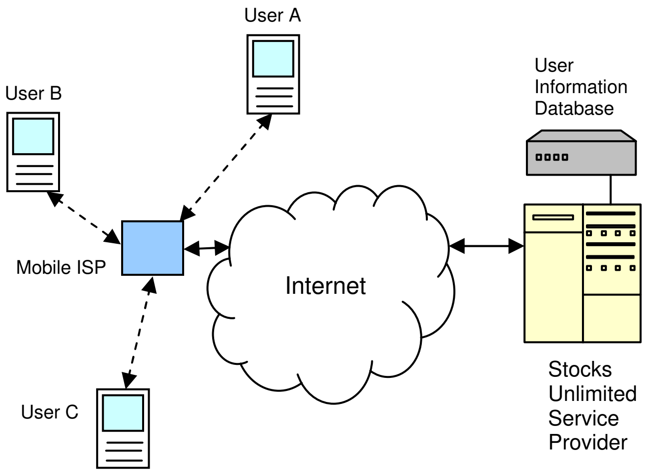


Figure 7. Network view of the Stocks Unlimited service

Stocks Unlimited makes use of a security policy to specify the security measures that it will use to protect its service. However, this security policy may not match up with the security preferences of the service user as we have seen above in the Introduction section. In addition, this security policy may not match up with the computational power of the user's mobile platform, or with security requirements imposed by the location of the mobile platform. To illustrate, suppose the security policy of Stocks Unlimited calls for encrypting the communication channel using AES (Advanced Encryption Standard). However, the user's cell phone has insufficient computing power to compute AES with reasonable performance, possibly leading to a security breach. Again, suppose there is an area of a large city that is notorious for man-in-the-middle attacks against mobile Internet services. Service users try to avoid this area but occasionally they have to traverse it in order to get to their destination, resulting again in a possible security breach. Our approach of security policy personalization can be applied to remedy these situations by having the consumer negotiate a suitable security policy with Stocks Unlimited, containing the desired sets of security mechanisms, and then using a software agent to select particular mechanisms on-the-fly. The agent would initiate the best available security mechanisms from among the negotiated choices, depending on the user's security preferences for the service, the computational power of the user's mobile platform, and the location of the user's mobile platform. We refer to this combination of user preferences, power, and location as "UPL".

Let us assume that the user negotiates with the provider to personalize the security policy following the procedure described above for negotiating sets of mechanisms, starting from the security policy template in Table 1. She then obtains, for example, the security policy in Table 3, where alternative mechanisms have been labeled with "P1" and "P2" for selection by the software agent.

Table 3. Example negotiated security policy for Stocks Unlimited

***Policy Use:* stock trading**	***Owner:* Stocks Unlimited**	***Valid:* unlimited**
CONSUMER PROVISIONS	**PROVIDER PROVISIONS**	**PROVIDER PROVISIONS**
Consumer Authentication *Implement:* yes (default) *P1: Mechanism:* password *P2: Mechanism:* V+F biometrics	**Provider Authentication** *Implement:* yes (default) *P1: Mechanism:* security token *P2: Mechanism:* digital signature	**Data *S*tore Confidentiality** *Implement:* yes (default) *Mechanism:* 3DES encrypt
Consumer Non-Repudiation *Implement:* yes (default) *Mechanism:* digital signature	**Provider Non-Repudiation** *Implement:* yes (default) *Mechanism:* digital signature	**Communication Confidentiality** *Implement:* yes (default) *P1: Mechanism:* SSL *P2: Mechanism:* VPN
Consumer Certification *Implement:* yes (default) *Mechanism:* certificate	**Provider Certification** *Implement:* yes (default) *Mechanism:* certificate	**Communication Integrity** *Implement:* yes (default) *Mechanism:* MD5 Hash
Consumer Malware Detect *Implement:* yes (default) *Mechanism:* Norton	**Provider Malware Detect** *Implement:* yes (default) *Mechanism:* Norton	**Secure Logging** *What:* order transactions *Mechanism:* 3DES encrypt *What:* user input *Mechanism:* 3DES encrypt

We call the software agent in this example a *context-aware security policy agent (CASPA)*. A CASPA is an intelligent software agent that resides in a mobile device and is responsible for selecting security services and mechanisms from the provider's security policy for a particular service, according to the values of UPL. The behaviour of a CASPA is described by the state machine in Figure 8, where the arrow labels are in the form "condition / action".

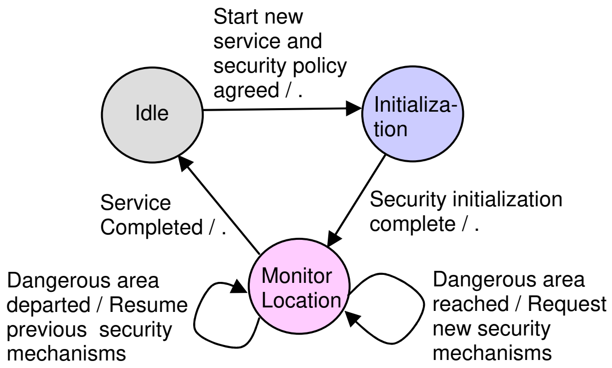

Start new service and security policy agreed / .
Idle
Initializa-tion
Service Completed / .
Security initialization complete / .
Dangerous area departed / Resume previous security mechanisms
Monitor Location
Dangerous area reached / Request new security mechanisms

Figure 8. Behaviour of CASPA

In Figure 8, the *Idle* state is exited once the service is ready to begin (i.e. the service has been found and the security policy agreed to between consumer and provider). In the *Initialization* state, the CASPA accounts for the U and P of UPL (i.e. reflects the user's security preferences and the computational power of the device) by setting the options in the negotiated provider's security policy to implement appropriate security services and mechanisms (see Table 3). For example, suppose the user has several mobile devices that she uses with the same security policy, including a PDA and a less powerful cell phone. CASPA would set security services and mechanisms that both reflect the consumer's security preferences and be appropriate to the computing power of each device. It would be straight forward to program a CASPA to perform this task. In the *Monitor Location* state, the agent is monitoring the device's location using GPS. Note that this location is only used by the CASPA and is not reported to either the mobile ISP or the provider of the service so that there should be no privacy concerns. An alternative way of determining the consumer's location is the use of signaling analysis by the mobile ISP. However, the latter would then learn the consumer's location leading to privacy concerns. When a dangerous area (i.e. an area with a high number of attackers) is entered, the agent messages the service provider to initiate a more powerful security mechanism for communication to defend against the attackers. Of course, this more powerful mechanism consumes more computing resources and should only be used when necessary. When the dangerous area is exited, the agent messages the provider that the normal security mechanism for communication may be resumed. The CASPA executes concurrently with the service. However, the service does not begin until the CASPA has completed the initialization. This example has been adapted from Yee & Korba (2005c), which may be consulted for further details including how the above dangerous area can be known, the secure communication protocols needed between the CASPA and the provider, certain operational requirements, and a discussion on location privacy.

This example application of security policy personalization calls for the use of a context-aware security policy agent to further personalize the security services according to UPL. This can be termed double personalization. The first personalization using security policy negotiation determines the sets of security mechanisms and corresponding security services required by the consumer. The second agent-based personalization dynamically accounts for the user's specific security preferences, the mobile device's available computing power, and the user's movement into a dangerous area with a higher number of attackers, where more powerful security mechanisms are needed. Security policy personalization is a form of service personalization that studies have shown is attractive to consumers (Ho & Kwok, 2003).

Easy Learn

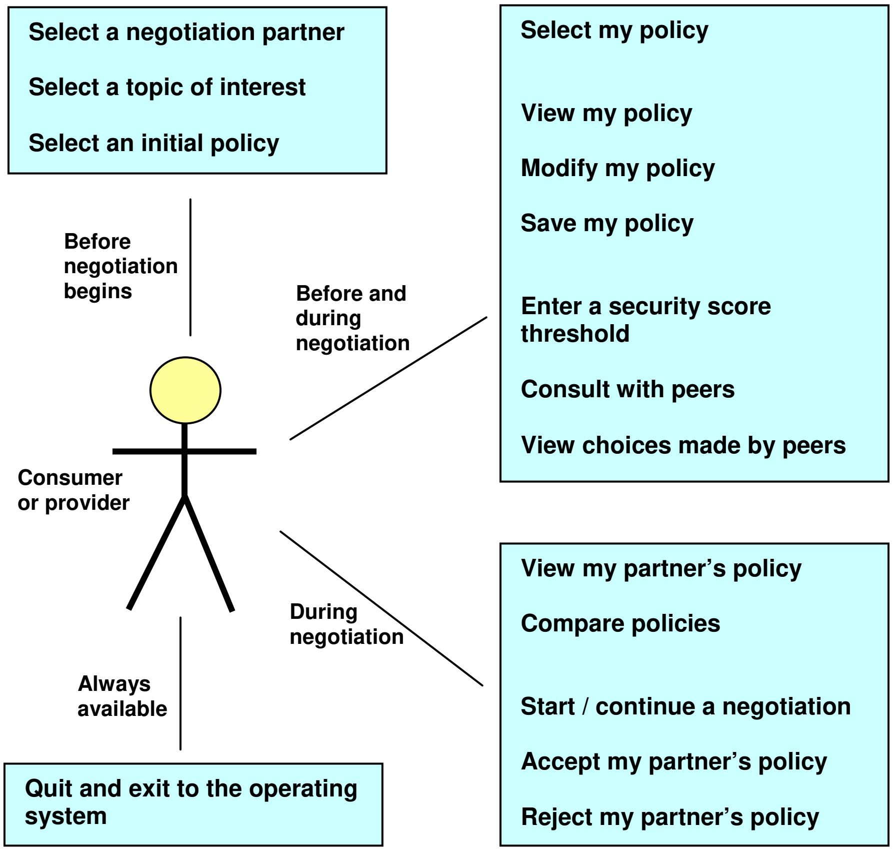
Easy Learn is an Internet-based e-learning service provider whose services consist of the delivery of online courses on numerous subjects, ranging from courses for the general public (e.g. "Finding Reliable Information on the Internet") to courses that are highly technical and of interest only to specific groups (e.g. "Maintenance Requirements for Next Generation M-5000 Tanks" for the military). Easy Learn has a security policy that it uses for all its e-learning courses, mainly focusing on user authentication and communications security. Suppose that user authentication is implemented using the familiar USERID/ password combination and secure communications is achieved using SSL. This security policy may suffice for someone taking the finding information course since the course content is probably not of a secretive or classified nature. However, the tank maintenance course is quite the opposite, since maintenance information could reveal vulnerabilities that can be taken advantaged of by adversaries. Thus, the military would want to negotiate the security policy, perhaps requiring two-factor biometrics authentication in addition to USERID/password, and the use of AES encryption for the course content while in transit.

PROTOTYPE FOR SECURITY POLICY NEGOTIATION

We have extended a prototype that we had developed for privacy policy negotiation (Yee & Korba, 2003a; Yee & Korba, 2003b) so that it can be used for security policy negotiation. The prototype is based on a peer-to-peer architecture programmed in JADE (Java Agent Development Framework) (Telecom Italia Lab). The prototype allows a consumer and a provider to contact each other across the Internet, initiate, and carry on a negotiation session.

The high-level functionality of the prototype is described by Figure 9. In Figure 9, a consumer-provider negotiation is considered a partner-partner negotiation, so for such a negotiation, the consumer's partner is the service provider and vice versa. Also, "topic of interest" refers to the type of service, and "consult with peers" means obtain help for negotiation using the scheme for online help in making offers described above.

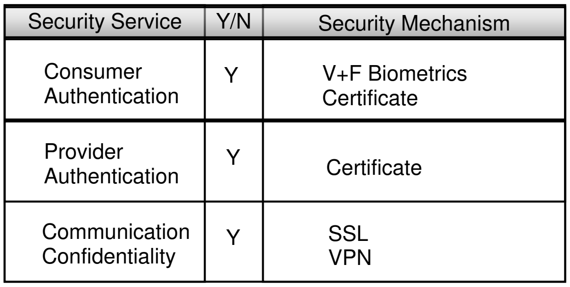
Only minor changes were needed to the prototype for security policy negotiation. The changes primarily involved a) provision of a pop-up window help facility for consumers who need to learn about a particular security service or mechanism (to satisfy requirement 1 of Subsection "Security Policy Negotiation Requirements"), and b) enhancing the user selection mechanism to allow for selection of multiple choices needed for some security services such as authentication and for negotiating sets of security mechanisms.



Select a negotiation partner
Select a topic of interest
Select an initial policy
Before negotiation begins
Select my policy
View my policy
Modify my policy
Save my policy
Enter a security score threshold
Consult with peers
View choices made by peers
Before and during negotiation
Consumer or provider
During negotiation
View my partner's policy
Compare policies
Start / continue a negotiation
Accept my partner's policy
Reject my partner's policy
Always available
Quit and exit to the operating system

Figure 9. High-level functionality of the security policy negotiation prototype

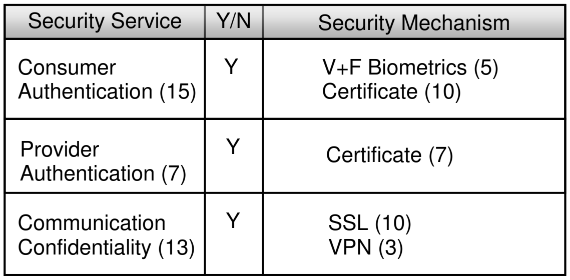
For security policy negotiation, the main component of the user interface consists of a table (see Figure 10) that has columns for security service, implement (Y/N), and security mechanism. Figure 10 only shows 3 security services for ease of explanation. A consumer can change the "Y" (default) to "N" to delete the associated security service. If the "Y" is left alone, the consumer can then select one or more of the corresponding security mechanisms.



Security Service	Y/N	Security Mechanism
Consumer Authentication	Y	V+F Biometrics Certificate
Provider Authentication	Y	Certificate
Communication Confidentiality	Y	SSL VPN

Figure 10. Tabular interface of security policy negotiation prototype

For consumers who need help regarding what security choice to make during a negotiation session, we implemented the help scheme described above. The user interface provides this help by showing the number of previously used security policies (corresponding to the same service and provider, with security scores below the threshold) that implemented each previously used choice by appending the number next to the choice (see Figure 11). Of course, the consumer must have previously requested help (via a button) and entered a security score threshold. For example, Figure 11 shows that "Consumer Authentication" having mechanisms below the threshold was used in 15 policies. In addition, mechanism "V+F Biometrics" of Consumer Authentication is below the threshold and used in 5 policies. Mechanism "Certificate" of Consumer Authentication is also below the threshold and used in 10 policies. This advises the consumer that Consumer Authentication is "a good security service to have" relative to the other two services. This does not mean that the consumer will only select Consumer Authentication, as the choices also depend on the needs of the consumer and the service. However, this information does guide the consumer in making a selection by letting her know what security measures have been used previously and how often they were used. Within Consumer Authentication, the certificate mechanism is more popular than the biometrics mechanism. This may guide the consumer into choosing certificate over biometrics, assuming the negotiation is for single mechanisms. Although some of these choices (e.g. consumer authentication) may seem obvious to security knowledgeable people, we point out that we are targeting the general public with our approach and there are people in this group who are not familiar with the choices. We have not yet trialed this prototype on the public to evaluate validity and usability. We plan to do this next and report the results in a subsequent paper.



Security Service	Y/N	Security Mechanism
Consumer Authentication (15)	Y	V+F Biometrics (5) Certificate (10)
Provider Authentication (7)	Y	Certificate (7)
Communication Confidentiality (13)	Y	SSL (10) VPN (3)

Figure 11. Frequency of security services and mechanisms

RELATED WORK

Work related to the topic of this paper generally fall under three categories: i) the specification and use of security policies, ii) the negotiation of trust, and iii) the personalization of security policies. One work (Ryutov et al., 2005) straddles ii) and iii). Of these three, category i) has the most number of papers, followed by category ii) with fewer but still many papers, followed by category iii) with only a handful of papers. The latter category not only has the smallest number of papers, but the works they describe do not deal with personalization using *personal* negotiation between a service consumer and a service provider. Rather, they concern automatic negotiation for networking resources or other forms of automatic adaptation of privacy or security policies. Thus our work on personal security policy negotiation is unique as far as we can tell. We provide a summary of each related work below.

Category i): The Specification and Use of Security Policies. Security policies have traditionally been used to specify security requirements for networks and distributed systems (Varadharajan, 1990). Bertino et al. (2001) present a XML-based language for specifying credentials and security policies for Web documents. More recently, security policies have been applied to manage security for distributed multimedia services (Duflos, 2002) and for very large, dynamically changing groups of participants in, for example, joint command of armed forces for some time period (Dinsmore et al., 2000). Ventuneac, Coffey, & Salomie (2003) describe a policy-based security framework for web-enabled applications, focusing on role-based security policies and mechanisms. Scott & Sharp (2003) present a structuring technique for abstracting security policies from large Web applications. The abstracted policy is expressed in a machine processable policy language and used to program an application level firewall. This firewall then dynamically analyzes and transforms HTTP requests/responses to enforce the policy. They claim that such a high-level technique is needed to overcome the problem of too many security holes in Web applications to fix individually. More recently, Bhargavan et al. (2005) describe a rule-based advisor tool that detects typical errors in Web Services configuration and security policy files. The tool generates a security report after checking for over thirty syntactic conditions corresponding to errors found during security reviews. Faheem (2005) presents a multi-agent based system for managing security policies, with the goal of making it easier to configure and implement a given security policy under dynamically changing threat conditions. Tan et al. (2004) describe the use of meta-level architectures for managing and discerning policy-based security specifics. They discuss how such use can detect and resolve policy conflicts as well as lead to a security reconfiguration if warranted by a change in the environment. Finally, Yau et al. (2005) present an adaptable security framework for large scale service-based systems. Their framework includes a core ontology together with a security specification language for specifying dynamic security policies, policy conflict detection and resolution, and tools for deploying agents to enforce security policies. They claim that their framework allows security policies of large scale service-based systems to be rapidly specified, updated, verified, and enforced for various threat situations.

Category ii): The Negotiation of Trust. Trust negotiation is applied to situations where peers need to interact across a network (such as the Internet) and the peers are complete strangers to one another. Trust negotiation is used to establish trust between such peers by iteratively exchanging certified digital credentials. Trust negotiation differs from security policy negotiation as described in this work in that the purpose of trust negotiation is to establish trust between interacting parties who do not trust one another, so that further online processing can proceed. It does not usually negotiate all the security mechanisms to be used for an electronic service where there is already some trust for the service provider. Examples of typical papers on trust negotiation are Bertino, Ferrari, & Squicciarini (2004), Winslett et al. (2002), and Winsborough, & Li (2004). More recently, Lee et al. (2006) present a third party authorization service that leverages the power of existing prototype trust negotiation systems by acting as an authorization broker. Their system issues resource access tokens in an open system after the interacting parties use trust negotiation to satisfy the appropriate resource access policies. For this work we view trust negotiation as complementary but not needed in most cases of provider-consumer relationship. This is because providers of Internet services have ways of making themselves known to consumers (e.g. advertising) and readily conduct business with strangers (with appropriate safeguards).

Category iii): The Personalization of Security Policies. The available papers largely describe security policy negotiation across Internet domains needed to manage cross domain network security (e.g. Barrere, Benzekri, Grasset, Laborde, & Nasser (2003), Yang, Fu, & Wu (2003), and Park, & Chung (2003)), negotiated resource sharing agreements between members of coalitions (Khurana, Gavrilal, Bobba, Koleva, Sonalker, Dinu, Gligor, & Baras, 2003), security policy mediation between heterogeneous information systems (Hale, Galiasso, Papa, & Shenoi, 1999) for secure interoperation, and negotiation of security parameters within protocols such as SSL (Chou, 2002). Additional examples of security mediation or adaptation follow. Rannenberg (2001) discusses multilateral security in which security policies of different parties may conflict and gives some examples and solutions to resolve the conflicts. In considering that different interests must be respected, Rannenberg (2001) confirms our ideas that a) different parties may have different (or personal) security goals, b) these parties can specify their own interests or security goals, and c) conflicts may be negotiated. Torrellas et al. (2003) propose the use of multi-agent security systems to react to a changing threat environment due to new virus attacks, active intrusions, and new attack technologies. They emphasize the need for flexible security, which again supports our contention that “one size does not fit all”. Finally, Ryutov et al. (2005) (can also be classified under Category ii)) propose a framework for adaptive trust negotiation that targets security attacks where the participants interact across security domains. Their framework adapts the associated security policies according to the sensitivity of the access request and a suspicion level assigned to the information requester.

It is interesting to note that some of the above works refer to changes in security policy necessitated by changes in the environment. This idea is also at the heart of this work. For us, a user is part of the environment and changes in security preferences among users mean changes to the environment that necessitate personalization of (changes to) security policies. Note that we only provide brief summaries of the above works, believing that more detail is not justified since these works relate to this work only in a minor way.

EVALUATION

We have presented a negotiations approach for Internet services security policy personalization, including a scheme for providing online help to consumers who are not sure of what security choices to make. Some strengths of our personalization approach are: a) straightforward and easy to use with appropriate interfaces as described above for our prototype, b) achieves its principal goal of meeting the personal security requirements of each user, and c) provides online help for the user in making security choices during negotiation. Some potential weaknesses of our approach are: a) in order for the approach to be used, users need to be somewhat Internet and security literate, b) the scheme for online help may be vulnerable to malicious biasing of security services and mechanisms through purposeful selection of weak security for a particular Internet service over some time period, and c) the scheme for online help may not be scalable with the number of users.

Regarding the potential weakness that users need to be Internet and security literate, we can say that users will become more literate in these areas over time. We have only to point to the large number of people using e-commerce and banking services over the Internet today, something that required higher levels of online literacy just a few years ago. In addition, the frequent lapses in electronic security that are headlined in the media also result in improved knowledge of security in the general population. In terms of the malicious biasing vulnerability mentioned above, one simple solution would be for the SA to reject security policies that employ outlandishly weak security so that such policies would not enter into the Security Score Table. Another solution would be to have the provider simply not accept a security choice if it is deemed too weak. As for the potential scalability weakness, we need to research this possibility further.

The scheme for online help clearly increases the workload of providers, but perhaps they would not mind the extra work if they can advertise that they are doing this to help consumers, and thereby gain more business. The scheme works for the negotiation of both single security mechanisms or sets of security mechanisms. For sets, the consumer would simply select a set from the better mechanisms (those having lower security scores) for a selected security service during negotiation. There would be a need for the SA to make sure the security score tables are kept up-to-date. In addition, the consumer's database retrievals need to occur in real-time so that the provider is not kept waiting unduly for the consumer to respond during negotiation. Both of these requirements can be easily fulfilled. The SA may be a government department or an extended role for a Certification Authority currently part of Public Key Infrastructure. The authority can recover its costs by charging consumers a small subscription rate for the use of its security consultation service (step 3 in the scheme for online help described above).

The use of the UDDI web services directory brings up an interesting possibility. Providers could store their security policies, in addition to details of their service offerings, in this directory. Consumers could then use the UDDI directory to select only those services with security policies that match (or come close to matching) their security preferences. This could lessen the need for negotiation (but not get rid of it entirely, as there may not be services with security policies that match completely). This can result in faster service invocation. However, the UDDI directory would need appropriate security protection, since successful attacks on this directory would be disastrous.

CONCLUSIONS AND FUTURE RESEARCH

Users of Internet services, including web services, have differing security requirements when invoking the services. The approach presented in this work for security policy personalization is a good way to fulfill these requirements that can lead to more widespread use of such services, with accompanying economic benefits. While some challenges remain before the approach can be widely adopted (see above Evaluation section), we feel that these challenges can be overcome and the full benefits of the approach realized.

The novel contributions of this work include: a) a security policy personalization approach for consumers of Internet services, b) a scheme for online help in making security policy offers during negotiation, and c) an interface for b) that easily and intuitively conveys the help needed. In addition, we have purposely kept our approach for a) simple, primarily so that the average consumer who is not a computer expert can understand how to use it.

Future research includes the following areas: a) performance and scalability of the scheme for providing online help to consumers for making security choices, b) alternative methods for ranking the security mechanisms in this scheme, c) the use of autonomous agents to automate the security policy negotiation process, d) the use of the UDDI directory to store provider security policies, e) further details on implementing security policy negotiation for Web Services – we have only indicated how it can be done at a high level, and f) security mechanisms required for the security policies and the negotiations themselves.

ACKNOWLEGMENT

The authors gratefully acknowledge the support of the National Research Council Canada for this work. In addition, the authors would like to thank colleagues Chris McDonald and Andrew Patrick for contributing to the design of the prototype's user interface.

REFERENCES

Bajaj, S., Box, D., Chappell, D., Curbera, F., Daniels, G., Hallam-Baker, P., Hondo, M., Kaler, C., Langworthy, D., Nadalin, A., Nagaratnam, N., Prafullchandra, H., von Riegen, C., Roth, D., Schlimmer, J., Sharp, C., Shewchuk, J., Vedamuthu, A., Yalcinalp, U., Orchard, D. (2006a). Web Services Policy Framework (WS-Policy). Version, 1.2, March. Available as of March 28, 2006 from: http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-polfram/ws-policy-2006-03-01.pdf

Bajaj, S., Box, D., Chappell, D., Curbera, F., Daniels, G., Hallam-Baker, P., Hondo, M., Kaler, C., Maruyama, H., Nadalin, A., Orchard, D., Prafullchandra, H., von Riegen, C., Roth, D., Schlimmer, J., Sharp, C., Shewchuk, J., Vedamuthu, A., Yalcinalp, U. (2006b). Web Services Policy Attachment (WS-PolicyAttachment). Version, 1.2, March. Available as of March 28, 2006 from: http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-polatt/ws-polat-2006-03-01.pdf

Barrere, F., Benzekri, A., Grasset, F., Laborde, R., & Nasser, B. (2003). Inter-Domains Policy Negotiation. Proceedings, 4th International Workshop on Policies for Distributed Systems and Networks (POLICY'03), pp. 239-242, Lake Como, Italy, June.

Bertino, E., Castano, S., & Ferrari, E. (2001). On Specifying Security Policies for Web Documents with an XML-Based Language. Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies, pp. 57-65, Chantilly, Virginia, USA, May.

Bertino, E., Ferrari, E., & Squicciarini, A. (2004). Trust Negotiation: Concepts, Systems, and Languages. *Computing in Science and Engineering*, pp. 27-34, July/August.

Bhargavan, K., Fournet, C., Gordon, A., & O'Shea, G. (2005). An Advisor for Web Services Security Policies. Proceedings of the 2005 Workshop on Secure Web Services (SWS '05), pp. 1-9, Fairfax, Virginia, USA, November.

Chou, W. (2002). Inside SSL : The Secure Sockets Layer Protocol. In *IT Pro*, pp. 47-52, July-August.

Della-Libera, G., Gudgin, M., Hallam-Baker, P., Hondo, M., Granqvist, H., Kaler, C., Maruyama, H., McIntosh, M., Nadalin, A., Nagaratnam, N., Philpott, R., Prafullchandra, H., Shewchuk, J., Walter, D., Zolfonoon, R. (2005). Web Services Security Policy Language (WS-SecurityPolicy). Version 1.1, July. Available as of July 28, 2006 from: ftp://www6.software.ibm.com/software/developer/library/ws-secpol.pdf

Dinsmore, P., Balenson, D., Heyman, M., Kruus, P., Scace, C., & Sherman, A. (2000). Policy-Based Security Management for Large Dynamic Groups: An Overview of the DCCM Project. Proceedings, DARPA Information Survivability Conference and Exposition (DISCEX'00), Vol. 1, pp. 64-73, Hilton Head, South Carolina, USA, January.

Duflos, S. (2002). An Architecture for Policy-Based Security Management for Distributed Multimedia Services. Proceedings of the Tenth ACM International Conference on Multimedia, pp. 653-655, Juan-les-Pins, France, December.

Faheem, H. (2005). A Multiagent-Based Approach for Managing Security Policy. Proceedings, Second IFIP International Conference on Wireless and Optical Communications Networks (WOCN 2005), pp. 351-356, Dubai, UAE, March.

Hale, J., Galiasso, P., Papa, M., & Shenoi, S. (1999). Security Policy Coordination for Heterogeneous Information Systems. Proceedings, 15th Annual Computer Security Applications Conference (ACSAC '99), pp. 219-228, Scottsdale, Arizona, USA, December.

Ho, S., & Kwok, S. (2003). The Attraction of Personalized Service for Users in Mobile Commerce: An Empirical Study. *ACM SIGecom Exchanges,* 3(4), pp. 10-18, January.

International Organization for Standardization. (n.d.). IS0 7498-2, Information Processing Systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture. Available as of Feb. 11, 2004 from: http://www.iso.org/

ITU-T. (n.d.). Recommendation X.800, Security Architecture for OSI. Available as of Feb. 11, 2004 from: http://www.itu.int/rec/recommendation.asp?type=items&lang=e&parent=T-REC-X.800-199103-I

Joshi, J., Aref, W., Ghafoor, A., & Spafford, E. (2001). Security Models for Web-Based Applications. *Communications of the ACM*, 44(2), pp. 38-44, February.

Khurana, H., Gavrila1, S., Bobba, R., Koleva, R., Sonalker, A., Dinu, E., Gligor, V., & Baras, J. (2003). Integrated Security Services for Dynamic Coalitions. Proceedings, DARPA Information Survivability Conference and Exposition (DISCEX'03), Vol. 2, pp. 38-40, Washington D.C., USA, April.

Lee, A.J., Winslett, M., Basney, J., & Welch, V. (2006). Traust: A Trust Negotiation-Based Authorization Service for Open Systems. Proceedings, Eleventh ACM Symposium on Access Control Models and Technologies, pp. 39-48, Lake Tahoe, California, USA, June.

OASIS (2006). Web Services Security: SOAP Message Security 1.1 (WS-Security 2004). OASIS Standard Specification, February 1. Available as of March 28, 2006 from: http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf

O'Neill, M. (2003). *Web Services Security*. McGraw-Hill / Osborne.

Park, J., & Chung, J. (2003). Design of SPS Model Using Mobile Agent System. Proceedings, IEEE 37th Annual 2003 International Carnahan Conference on Security Technology, pp. 38-42, Taipei, Taiwan, October.

Rannenberg, K. (2001). Multilateral Security a Concept and Examples for Balanced Security. Proceedings of the 2000 Workshop on New Security Paradigms, pp. 151-162, Ballycotton, County Cork, Ireland, September.

Ryutov, T., Zhou, L., Neuman, C., Leithead, T., & Seamons, K.E. (2005). Adaptive Trust Negotiation and Access Control. Proceedings, Tenth ACM Symposium on Access Control Models and Technologies, pp. 139-146, Stockholm, Sweden, June.

Scott, D. & Sharp, R. (2003). Specifying and Enforcing Application-Level Web Security Policies. *IEEE Transactions on Knowledge and Data Engineering*, 15(4), pp. 771-783, July/August.

Tan, J.J., Poslad, S., & Xi, Y. (2004). Policy Driven Systems for Dynamic Security Reconfiguration. Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, Vol. 3, pp. 1274-1275, New York, New York, USA, August.

Telecom Italia Lab. (n.d.). JADE (Java Agent Development Framework). Available as of Feb. 14, 2005 from: http://jade.tilab.com/

Torrellas, G.A.S. & Vargas, L.A.V. (2003). Modelling a Flexible Network Security Systems Using Multi-Agents Systems: Security Assessment Considerations. Proceedings of the 1st International Symposium on Information and Communication Technologies (ISICT 03), pp. 365-371, Dublin, Ireland, September.

Varadharajan, V. (1990). A Multilevel Security Policy Model for Networks. Proceedings, Ninth Annual Joint Conference of the IEEE Computer and Communication Societies (INFOCOM 90), Vol. 2, pp. 710-718, San Francisco, California, USA, June.

Ventuneac, M., Coffey, T., & Salomie, I. (2003). A Policy-Based Security Framework for Web-Enabled Applications. Proceedings, 1st International Symposium on Information and Communication Technologies, pp. 487-492, Dublin, Ireland, September.

Winsborough, W., & Li, N. (2004). Safety in Automated Trust Negotiation. Proceedings, 2004 IEEE Symposium on Security and Privacy (S&P'04), pp. 147-160, Oakland, California, USA, May.

Winslett, M., Yu, T., Seamons, K., Hess, A., Jacobson, J., Jarvis, R., Smith, B., & Yu, L. (2002). Negotiating Trust on the Web. *IEEE Internet Computing*, pp. 30-37, November/December.

World Wide Web Consortium (W3C) (n.d.). Extensible Markup Language (XML). Available as of March 18, 2006 at: http://www.w3.org/XML/

Yang, Y., Fu, Z., & Wu, S. (2003). Bands: An Inter-Domain Internet Security Policy Management System for IPSEC/VPN. Proceedings, IFIP/IEEE Eighth International Symposium on Integrated Network Management, pp. 231-244, Colorado Springs, Colorado, USA, March.

Yau, S., Yao, Y., Chen, Z., & Zhu, L. (2005). An Adaptable Security Framework for Service-based Systems. Proceedings, Tenth IEEE International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS 2005), pp. 28-35, Sedona, Arizona, USA, February.

Yee, G., & Korba, L. (2003a). Bilateral E-services Negotiation Under Uncertainty. Proceedings, The 2003 International Symposium on Applications and the Internet (SAINT2003), pp. 352-355, Orlando, Florida, USA, January.

Yee, G., & Korba, L. (2003b). The Negotiation of Privacy Policies in Distance Education. Proceedings, 14th IRMA International Conference, pp. 702-705, Philadelphia, Pennsylvania, USA, May.

Yee, G., & Korba, L. (2005a). Semi-Automatic Derivation and Use of Personal Privacy Policies in E-Business. *International Journal of E-Business Research*, 1(1), pp. 54-69, January - March.

Yee, G., & Korba, L. (2005b). Negotiated Security Policies for E-Services and Web Services. Proceedings, 2005 IEEE International Conference on Web Services (ICWS 2005), Volume 2, pp. 605-612, Orlando, Florida, USA, July.

Yee, G., & Korba, L. (2005c). Context-Aware Security Policy Agent for Mobile Internet Services. Proceedings, The 2005 IFIP International Conference on Intelligence in Communication Systems (INTELLCOMM 2005), pp. 249-259, Montreal, Quebec, Canada, October.

ABOUT THE AUTHORS

George Yee (www.georgeyee.ca) is a Senior Research Officer in the Information Security Group, Institute for Information Technology, National Research Council Canada (NRC). Prior to joining the NRC in late 2001, he spent over 20 years at Bell-Northern Research and Nortel Networks. George received his PhD (Electrical Engineering) from Carleton University, Ottawa, Canada, where he is currently an Adjunct Research Professor. He is a Senior Member of IEEE, and member of ACM and Professional Engineers Ontario. His research interests include security and privacy for e-services, using software agents to enhance reliability, security, and privacy, and engineering software for reliability, security, and performance.

Larry Korba is a Principal Research Officer with the National Research Council Canada. He is the Group Leader of the Information Security Group in the Institute for Information Technology (http://www.iit-iti.nrc-cnrc.gc.ca/) and involved in the research and development of security and privacy enhancing technologies for applications ranging from gaming to ad hoc wireless systems.