

# What Can Identity-Based Cryptography Offer to Web Services?

Jason Crampton    Hoon Wei Lim    Kenneth G. Paterson  
Information Security Group  
Royal Holloway, University of London  
Egham, Surrey TW20 0EX, UK  
{jason.crampton,h.lim,kenny.paterson}@rhul.ac.uk

## ABSTRACT

Web services are seen as the enabler of service-oriented computing, a promising next generation distributed computing technology. Independently, identity-based cryptography is emerging as a serious contender to more conventional certificate-based public key cryptography. However, the application of identity-based cryptography in web services appears largely unexplored. This paper sets out to examine how identity-based cryptography might be used to secure web services. We show that identity-based cryptography has some attractive properties which naturally suit the message-level security needed by web services.

## Categories and Subject Descriptors

K.6 [Management of Computing and Information Systems]: Security and Protection

## General Terms

Design, security

## Keywords

Identity-based cryptography, web services, message-level security

## 1. INTRODUCTION

The emergence of web services [2, 29, 44] is largely due to the result of attempts to integrate heterogeneous applications across the Internet. Achieving such inter-operability between cross-organisational applications has proved to be challenging because of the difficulty in passing remote service invocation calls through a firewall, the lack of standardised protocols, the need for loosely-coupled interactions and so forth. Previous attempts, using technologies such as the Distributed Computing Environment (DCE) [53], the Common Object Resource Broker Architecture (CORBA) [39]

and the Enterprise Application Integration (EAI) [35], have failed to fully address these issues [2]. On the other hand, web services, evolved from previous cross-domain integration technologies, are based on XML, message-oriented and supported by HTTP. Hence, web services seem to provide convincing answers to these challenges.

Adopting a message-oriented approach to enable interoperability is a key idea that underlies web services. It enables more dynamic, loosely-coupled and asynchronous interactions between inter-domain applications, compared to traditional approaches. Moreover, it may be desirable to protect only parts of a message that is being sent from one entity to another, so that intermediate nodes between the two entities can process the message appropriately. For example, a purchase order service may contain a customer's credit card details and it seems sensible to allow only an authorised party to view such information. However, other information, such as the product description and the purchase quantity, can be displayed in clear so that it is accessible by all parties that need to (partially) process the purchase order. This demonstrates that transport-level security provided by the SSL/TLS protocol [18], for example, may not necessarily be appropriate for securing web services messages. Thus, message-level security is an important aspect of web services.

Orthogonal to the development of web services, identity-based cryptography (IBC) [9, 58] is also a fast emerging research area in cryptology. Standardisation of IBC is well underway and commercial IBC products are now available. In an identity-based cryptosystem, an entity's public key is generated based on his identification information (identifier), such as username, email address, IP address or telephone number. The corresponding private key is produced by a trusted authority (TA) using a master secret. It is assumed that system users trust the TA to issue private keys only to the correct users. Since public keys can be constructed on-the-fly based on identifiers, certificates are not required to bind entities' identities to public keys. This seems to greatly simplify key management, particularly public key distribution. In addition, the use of identifiers can be extended to include information such as roles, validity periods and meaningful hierarchical namespaces. These attractive properties of IBC seem to fit naturally and align well with the notion of message-level security needed for web services.

Given the emergence of IBC as a serious contender to conventional certificate-based cryptosystems and the importance of web services, it is important and timely to examine

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SWS'07, November 2, 2007, Fairfax, Virginia, USA.

Copyright 2007 ACM 978-1-59593-892-3/07/0011 ...\$5.00.

the suitability of IBC for securing web services, seemingly an unexplored area to date. We discuss how various identity-based cryptographic techniques can be used to provide web services security in a natural way, with some reasonable assumptions and acceptable trade-offs. We also show how these identity-based techniques can be advantageous compared to existing certificate-based approaches. Additionally, we propose an identity-based key management framework for web services using hierarchical identity-based cryptography (HIBC) [23, 27].

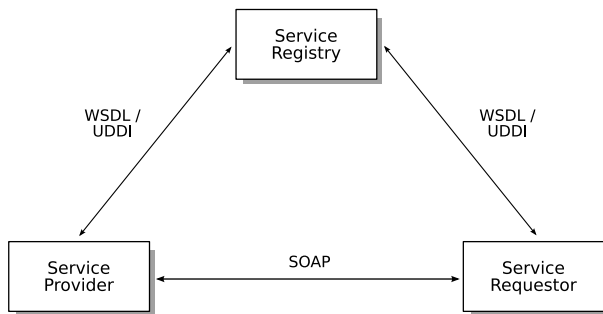
In the next section, we provide an overview of web services, and relevant XML security standards and technologies developed for web services. In Section 3, we introduce the concept of and describe the properties of IBC. In Section 4, we discuss how properties of IBC can play various roles in securing web services messages and provide improvements on certificate-based web services security. Section 5 presents an identity-based key management framework for web services environments, and Section 6 concludes the paper.

## 2. WEB SERVICES

The concepts, design and architectures of web services are heavily influenced by previous cross-organisational application integration technologies. In what follows, we briefly describe a basic web services architecture and its underlying building blocks.

### 2.1 Architecture

Many current web services architectures are based on the interaction of three types of entities: the service requestor, the service provider and the service registry [2, 29]. Generally speaking, the service provider advertises its services in a service registry. The service requestor finds a suitable service from the service registry, and subsequently interacts with the associated service provider. This architectural model is shown in Figure 1.



**Figure 1: The basic Web services architectural model.**

In order to implement web services based on this architecture, three core building blocks are used: SOAP [25], WSDL [13] and UDDI [15]. SOAP was designed as a uniform, unidirectional messaging mechanism to transport XML messages from one node to another. It is a protocol that underlies all interactions between web services. WSDL is an XML-based interface description language used to describe services in a standardised way. On the other hand, it is essential to have a standard way of publishing and locating services; UDDI is used as a service registry to accommodate these requirements.

## 2.2 Web Services Security

We now give a summary of key security XML security standards adopted for securing web services.

### 2.2.1 XML Encryption and Signature

Two fundamental building blocks for message-level security in web services are XML Encryption [20] and XML Signature [19]. XML Encryption specifies how standard cryptographic algorithms are used to encrypt XML data and the presentation of the encrypted data in XML format. It is designed to keep all or part of an XML document secret. On the other hand, XML Signature defines XML signature processing rules and syntax. An XML signature can be applied to some or all contents of one or more resources. Moreover, an XML document can contain multiple XML signatures.

### 2.2.2 WS-Security

The WS-Security specification [42] defines a common format for securing SOAP messages based on existing technologies. It makes use of XML Encryption and XML Signatures to protect message confidentiality and integrity. It also provides a way of passing security tokens, such as X.509 certificates [28] or Kerberos tickets [43], through SOAP headers.

Offering message-level security, WS-Security is instrumental in providing end-to-end web services security as each message can be encrypted or signed independently, and thus self-protected. This is in contrast to transport-level security which provides point-to-point security through, for example, a secure channel established using the SSL/TLS protocol. Although message-level security provides finer granularity than transport-level security in terms of selective message protection, this granularity potentially causes significant performance issues. This is because each message needs to be processed separately and different security tokens may be used within the same message or from message to message.

### 2.2.3 SAML

This standard defines methods for specifying trust assertions in XML [31]. These methods enable “portable trust” in the sense that assertions applied to an individual are attached to a message and they can be transported from one point to another with the message. SAML assertions take the form of authentication, authorisation or attributes of entities. One key benefit resulting from the use of SAML is web single sign-on.

The SAML authorisation assertion request/response protocol is usually run between a policy enforcement point (PEP) and a policy decision point (PDP), typically with the support of XACML [38]. It is also often used by a PEP to request attribute assertions from a policy information point (PIP).

### 2.2.4 XKMS

The use of XML encryption and signatures is pervasive in securing web services. However, proper implementation and use of a certificate-based PKI is known to be difficult [22, 50]. XKMS [26] has been developed to shield XML clients from the complexity of the underlying PKI(s) by allowing delegation of trust processing decisions to one or more web services called XKMS services. The presence of XKMS means that XML Signature and Encryption can be used independently of their underlying PKI vendor implementations.

XKMS services can be classified into two categories:

- (i) XML Key Information Service Specification (X-KISS) describes a protocol that allows a client to delegate part or all of the tasks required to process the XML Signature or Encryption `<KeyInfo>` elements to an XKMS service. For example, a Locate service can provide name resolution, while a Validate service, as implied by its name, provides key validation to ensure that a key has not been expired or revoked.
- (ii) XML Key Registration Service Specification (X-KRSS) describes a protocol for registration and subsequent management of public key information of a user. The XKMS services are Register, Reissue, Revoke and Recover. The user must register or bind his credential to a key pair through a Register service. A Reissue service can be used to obtain a previously registered key pair. Should the user wish to revoke his previously registered key pair, the task can be performed by a Revoke service. Lastly, a Recover service permits the user to obtain a previously registered and escrowed key pair which may have been lost.

Further information about the deployment of the aforementioned standards and other XML-based security specifications and mechanisms can be found in [45, 52, 59].

### 3. IDENTITY-BASED CRYPTOGRAPHY

Shamir [58] invented identity-based cryptography (IBC) in 1984. The key idea is to generate and use public keys based on publicly available information which can be used to uniquely identify users. On the other hand, private keys are generated for users by a trusted authority (TA) in possession of a master secret. The motivation for such an approach was to obviate the use and management of public key certificates. However, Shamir was only able to develop an identity-based signature (IBS) scheme based on the RSA primitive [51]. It was not until 2001 that the first practical and provably secure identity-based encryption (IBE) scheme was proposed, by Boneh and Franklin [9]. Their seminal work makes use of pairings on elliptic curves and is related to other pairing-based cryptographic proposals from Sakai *et al.* [54] and Joux [30]. The publication of Boneh and Franklin's paper triggered a flood of research in pairing-based cryptography. A more detailed survey of pairing-based cryptography can be found in [46].

#### 3.1 Schemes

We now describe generic algorithms for an IBE scheme.

**SETUP:** Given a security parameter, the algorithm generates a set of system parameters (which will be made public) and a master secret.

**EXTRACT:** This algorithm is run to extract the private key corresponding to a given public key. It takes the system parameters, the master secret and an identifier (public key string) as input, and returns a private key.

**ENCRYPT:** This algorithm uses the system parameters and an identifier (public key) to encrypt a message. It outputs a ciphertext.

**DECRYPT:** Using the system parameters, a private key and a ciphertext as input, this algorithm returns a plaintext (or possibly an indication that the decryption process has failed).

The **SETUP** and **EXTRACT** algorithms are normally executed by the TA, while the **ENCRYPT** and **DECRYPT** algorithms are carried out by users. A TA is a trusted third party roughly equivalent to a CA in a traditional certificate-based PKI. Note that in the Boneh-Franklin IBE scheme [9], a public key can be computed by simply applying a hash function to an arbitrary identifier, and the hash function is published as part of the system parameters.

Suppose that Alice wants to send a message secretly to Bob using an IBE scheme. She does not need to first verify the authenticity of Bob's public key (which must take place in a conventional certificate-based public key cryptosystem). Instead Alice simply encrypts the message with Bob's identifier, for example 'bob@example.com'. Clearly, Alice needs to know the system parameters of Bob's TA. If Bob does not already possess his private key, he has to obtain it from his TA using a confidential and authenticated channel. If the TA is satisfied that Bob is a legitimate receiver, it takes its system parameters, master secret and Bob's identifier to extract a private key, which can be used by Bob to decrypt the ciphertext.

An IBS scheme has **SETUP** and **EXTRACT** algorithms, like those of an IBE scheme, along with **SIGN** and **VERIFY** algorithms, described below.

**SIGN:** This algorithm takes as input the system parameters, a private key and a message to be signed. It returns a signature.

**VERIFY:** This algorithm takes as input the system parameters, a signature, a message and an identifier (public key) as input. It outputs 'valid' or 'invalid'.

Shortly after Boneh and Franklin's proposal, the notion of hierarchical identity-based cryptography (HIBC) emerged, and Gentry and Silverberg developed a secure, fully scalable hierarchical identity-based encryption (HIBE) scheme and a hierarchical identity-based signature (HIBS) scheme [23]. In the hierarchical setting, it is assumed that entities can be arranged in a rooted tree and that entities at one level are trusted to issue private keys to entities immediately below them in the tree. More specifically, the root TA, located at level 0, extracts private keys for entities at level 1, who in turn act as TAs for entities in their respective domains at level 2, *etc.* Each node in the tree has an identifier. The identifier of an entity is the concatenation of the node identifiers in the path from the root to the node associated with the entity.

More recently proposed IBE and HIBE schemes can be found in [8, 10, 63].

#### 3.2 Properties

We now identify some of the properties of IBC that distinguish it from conventional certificate-based cryptography.

- *Easy-to-construct public keys:* A public key is generated based on an entity's identifier (and some public parameters) and used on-the-fly without the need for a certificate look-up or verification. In fact, a message can be encrypted before the relevant decryption key has been extracted.
- *Certificate-free:* IBC does not require certificates since public keys are computed from public identifiers. Thus the binding between an identity and a public key is

direct in IBC, rather than being enabled by a certificate as in conventional public key cryptography. Note that the size of an identifier may be negligible compared to the size of an X.509 certificate. This may present a significant advantage in terms of communication cost savings, particularly in applications where multiple certificates need to be transmitted between two nodes as part of SOAP messages.

- *Self-describing public keys*: Since public keys are computed based on identifiers, which in turn, are predictable and human readable, no tools are needed to parse and render the keys, and simple text editors are sufficient for their manipulation. Unlike identity-based public keys, certificate-based public keys, such as RSA, have no discernible structure and require Base-64 encoding to render them in a more compact, printable textual form.
- *Natural hierarchical namespace*: In HIBC, public keys can be formed from the concatenation of identifiers for nodes in a tree. This implies that the identity-based approach can be used to model the logical relationships between entities/principals at different levels of a hierarchical structure in a very natural way. This results in a simple and neat way of managing keys, particularly public key distribution and generation operations.

However, IBC has some limitations. Notice that in the identity-based approach, it is only the TA that can compute private keys. This necessarily implies that an identity-based key infrastructure has an escrow facility, which may or may not be desirable. Boneh and Franklin suggested that key escrow can be circumvented by using multiple TAs and threshold cryptography [9]. Another issue is that it may be difficult to revoke an identifier if, for example, the corresponding private key has been compromised. Boneh and Franklin proposed the use of an expiry date concatenated with the recipient's identifier to achieve automated key expiry. The idea here is that an encrypting party will first consult a policy for identifier construction associated with the relevant TA before performing an encryption, while the recipient needs to have the corresponding private key for the extended identifier in order to decrypt. Then private keys become useless after their corresponding identifiers have expired, since encrypting parties will be using an up-to-date identifier. This approach may obviate the need for a revocation mechanism. However, it has the disadvantage of increasing the TA's workload, since the TA is required to regularly generate private keys and deliver them to users. Nevertheless, we will show, in Section 5, how HIBC can be used to alleviate the TA's workload. More detailed discussions comparing identity-based and certificate-based architectures can be found in [11, 47].

## 4. ROLES OF IBC IN WEB SERVICES SECURITY

We are now ready to take a closer look at the properties of IBC and examine how the use of IBC in web services improves on conventional certificate-based approaches.

### 4.1 Simplified Public Key Distribution

In a more conventional public key cryptosystem, such as RSA, a public key is distributed in the form of a public

key certificate, with X.509 as the popular choice of certificate format. In WS-Security, a public key certificate usually forms part of the `KeyInfo` element within a SOAP header. Alternatively, a reference can be specified in the `KeyInfo` element in the form of a URI pointing to a remote location, such as a public key directory.

In IBC, however, a human-readable and self-describing identifier can be used to represent a public key. This property greatly simplifies distribution of public keys because a public key can be computed directly from an identifier, which can be specified in the `KeyInfo` element and used on-the-fly.

XML representations of data tend to be significantly larger than their equivalent binary formats. An XML message can be 4 to 10 times larger than its equivalent binary representation [14]. This can substantially increase the communication costs, the latency of sending/receiving SOAP messages, and the time needed for parsing the XML data. By using IBC in place of conventional public key techniques, we envisage that the sizes of SOAP headers may well be reduced substantially.

Let us take an XML signature as an example. If *A* wants to submit a signed request in XML format to a service provider, she would need to attach her public key and certificate in the `KeyInfo` element if the RSA signature algorithm was used. However, if we adopt identity-based techniques, she could, in principle, simply include her identifier in the `KeyValue` element of the XML signature as follows.

```
<ds:KeyInfo>
  <ds:KeyValue>
    alice@example.org
  </ds:KeyValue>
</ds:KeyInfo>
```

When the service provider receives the signed XML message, it can construct *A*'s public key merely based on *A*'s identifier (assuming the service provider has the relevant authentic cryptographic system parameters, as explained in Section 3.1). The potential saving can be even more significant if the SOAP header contains multiple signatures requiring different verification keys. Even though a reference to a remote public key directory can be used in the certificate-based approach, the service provider is still required to retrieve and verify the associated public key certificate before the public key can be used.

Our identity-based approach also has benefits for XML encryption. In the identity-based setting, a public key can be used on-the-fly to encrypt messages. Examining the message flows in Figure 1, we envisage that the service requestor can obtain the service provider's identifier (and the associated cryptographic system parameters) from the service registry through UDDI. Hence, the service requestor can compute and use the service provider's public key on-the-fly. In principle, the requestor can encrypt a SOAP message for the service provider without obtaining a public key certificate from the service provider itself or a public key directory. In fact, the service registry may act as a public key directory.

Indeed, the use of identifiers can simplify public key distribution and can potentially greatly reduce the size of a SOAP header. Note that the simplified way of exchanging public keys (or security tokens) between two parties that we have discussed so far can be extended to the SSL/TLS handshake protocol, as shown in [34], and WS-SecureConversation [41].

Moreover, an identity-based public key is predictable, self-describing and human-readable. Thus no special tools are needed to parse and render the key information. This is indeed a very desirable property and meets a fundamental requirement of XML.

## 4.2 Efficient Provision of Cryptographic Services

We discuss how properties of IBC can be exploited to minimise the overheads incurred by cryptographic operations when providing message-level security.

In XML Encryption, the common practice for encrypting XML data is to use a symmetric encryption algorithm, such as AES or Triple-DES, and the symmetric encryption key is then “wrapped” with the receiver’s public key. The encrypted data and the encrypted symmetric key are transported to the receiver as part of a SOAP message. The motive for using this hybrid approach is the far superior performance of symmetric encryption as compared to encrypting XML data using an asymmetric encryption algorithm directly. Naturally, IBE could be used in place of the usual public key encryption to wrap the symmetric key, with similar performance guarantee [17].

If integrity protection is required for data, then this is provided using methods specified in the XML Signature standard. For example, the HMAC-SHA1 algorithm can be specified in the `SignatureMethod` element [19], and a MAC key can be derived from the data encryption key. If either data origin authentication or non-repudiation services are required (for accounting purposes [52, 59], for example), then an XML signature created using a public key signature algorithm can be used in place of the MAC. Thus XML Encryption and XML Signature must be used in tandem if additional cryptographic services beyond confidentiality are required. In fact, current cryptographic theory strongly suggests that encryption must be carefully combined with integrity protection to obtain robust security in the face of active attackers. This theory is supported by a number of real-world examples, see [4, 48, 64] for example. So it would seem both necessary and prudent to always use an appropriate combination of XML Encryption and XML Signature to obtain even a basic confidentiality service for XML documents.

IBC offers an opportunity to provide certain of these additional cryptographic services in a stream-lined manner. Sakai *et al.* [54] presented a technique by which any two parties  $A$  and  $B$  who are equipped with identity-based private keys, who know one another’s identifiers, and who are registered with the same TA, can efficiently compute shared keying material  $K_{AB}$  without any direct interaction. This approach is known as *identity-based, non-interactive key distribution*. The scheme of Sakai *et al.* applies for the keying infrastructure used in the Boneh-Franklin IBE scheme. It can be extended to the setting of the Gentry-Silverberg HIBE/HIBS keying infrastructure using an adaptation of ideas presented in [23].

Identity-based, non-interactive key distribution can be exploited as follows. The shared key  $K_{AB}$  can be used as a key transport key to wrap fresh keying material (using the AES key wrap algorithm [56], for example). From the fresh keying material, a symmetric encryption key and a MAC key can be derived (by appropriate application of a hash algorithm). Note that the key transport key can be specified in

the `EncryptedKey` element of the associated SOAP message. We can then combine encryption and MAC cryptographic transforms specified in XML Encryption and XML Signature to protect XML data. This provides a confidentiality and an integrity protection service. In addition, it provides a data origin authentication service “for free”, since the keying material  $K_{AB}$  can only be computed by  $A$  and  $B$  using their respective private keys. Thus  $B$ , on receipt of  $A$ ’s message, can be assured of its source. In the usual PKI-based approach, it appears that this service can only be obtained by using a digital signature in combination with public key encryption, requiring two public key operations per party, or by using optional-to-implement Diffie-Hellman key agreement. In contrast, in our approach, each party only needs to compute  $K_{AB}$ , which, depending on the particular identity-based scheme in use, may involve less computation. We also note that since the input to the MAC is encrypted data in Base-64 encoded form, our approach largely avoids the expensive canonicalisation required in creating a normal XML signature.

The identity-based approach described so far does not provide non-repudiation. If this service is required, then XML signatures based on IBC can be used in combination with the IBE-based XML encryption sketched above.

In concluding this section, whether IBC is used or not, we reiterate our belief that message confidentiality should be provided in a cryptographically robust manner in XML Encryption. The current approach of providing encryption and integrity protection mechanisms separately using XML Encryption and XML Signature, respectively, may introduce cryptographic vulnerabilities if not carefully implemented. Thus, it seems desirable to extend the current XML Encryption standard to support authenticated encryption algorithms in addition to symmetric and asymmetric encryption primitives. For example, one might use dedicated authenticated encryption algorithms [5] or a carefully selected combination of encryption and MAC algorithms [32]. We note that this possibility was apparently considered but rejected during the development of the XML standards. We will explore the options for this approach in more detail in future work.

## 4.3 Role-Based Signatures

When two entities from different trust domains want to interact using a business-to-business (B2B) system, for example, which makes use of web services, SAML assertions can be used to establish trust. Typically, a service requestor authenticates to his local authority and obtains the necessary authentication assertion, which is then, along with his business request, transported to a service provider through a SOAP message. Clearly, the service provider has to trust the service requestor’s authentication authority. Moreover, the trust relationship between these two domains may be bound by contractual agreements.

While it seems sensible and practical to authenticate cross-organisational entities using the above framework, cross-domain principal mapping in the sense of access control appears to be challenging. One main reason for this is that entities requesting access to remote resources may be unknown to the authorisation service that controls access to the requested resources. Thus, it seems inevitable that predefined mappings of principals in one domain to those in the domain containing the resources are needed. This is a prob-

lematic issue in open distributed computing environments rather than in web services per se.

Existing authorisation frameworks that deal with the problem of principal mapping, such as KeyNote [7], SPKI/SDSI [21], RBTM [33] and Akenti [62], rely on some form of certificate-based PKI. Essentially these frameworks rely on signed statements or assertions, attesting to the requestor or the associated public key having a particular attribute. A set of such attributes is used to map the requestor to principals in the relevant authorisation policy. However, there still remains the difficult problem of interpreting the requestor's attributes in the context of a different domain's authorization policy. These aforementioned authorisation frameworks presuppose that the remote domain is aware of what the requestor's attributes mean in the requestor's own domain.

A recent proposal by Crampton and Lim attempts to alleviate the cross-domain principal mapping problem using the concept of role signatures [16], which is based on IBC. It is based on the observation that a hierarchical identity-based signature (HIBS) scheme can be used naturally for role-based access control (RBAC) [55] within a distributed computing environment, such as a grid computing system, which has a hierarchical structure. A verification key can be defined by a role identifier (using roles as identifiers) defined within a hierarchical namespace. By doing so, user authentication and access control is unified, and credential verification is trivial, unlike credential discovery and verification in PKI-based approaches. An authorisation service is only required to verify a single signature, produced from the signing key associated with the role identifier, to both confirm that the user is an authenticated member of an organisation and occupies a particular role within that organisation.

Crampton and Lim's proposal can be adapted for enterprise or B2B environments. We envisage that a federated domain will have a hierarchical structure, enabling member organisations and principals within those organisations to be identified uniquely within a hierarchical namespace. We assume that the federated domain specifies a small number of generic roles that can be used as principals in the authorisation policy of each member organisation. We also assume that the federated domain is a principal who is trusted to enrol new member organisations into the federation, and that member organisations are trusted to assign their own users to generic roles. A HIBS scheme is then used to generate signing/verifying key pairs based on role identifiers defined in the hierarchical namespace.

Access requests are signed using a role's signing key. If the associated role identifier is correctly formed and the signature on the request can be verified, then the requestor is known to be authorised for that role in his home organisation. The verifier then uses its local policy to map the generic role to local roles and thus evaluate the request. Note that apart from the definition and publication of a comparatively small number of generic roles by the federated domain, there does not need to be agreement between individual member organisations about how to map principal identifiers. This seems to greatly alleviate the principal mapping problem.

Fitting this approach into web services, the notion of portable trust, which SAML offers, may well be provided in a much simplified manner using role signatures. This is because authentication and authorisation information of a

requestor is unified and can be derived from a single role signature. This potentially simplifies the SAML architecture, which currently involves multiple authorities for separate issuance of authentication, attribute and authorisation assertions. A PEP that receives a role signature verifies it using the associated role identifier. The PEP can then directly map the requestor's role information to the local policy information.

So far we have only discussed the case of multiple trust domains (physical organisations) within a single federated domain. This framework for supporting role signatures can be extended to multiple TAs/hierarchies in the context of HIBC, assuming each federated domain may share the same TA with or have a different TA from another federated domain. Within such a setting, we assume that these TAs would agree on a set of generic roles through out-of-band mechanisms. In the grid community, for example, this can be achieved through an independent body such as the EU Grid Policy Management Authority (PMA) or the Americas Grid PMA. More details about establishing multiple hierarchies will be provided in Section 5.

## 4.4 Role-Based Encryption

Our description of an architecture supporting role-based signatures can also be used to enable role-based encryption.

We envisage that role-based encryption can be useful in designing a "loosely-coupled" authorisation framework, in line with the spirit of web services. In such a framework, service providers encrypt their resources, in the form of data sets or electronic documents, using randomly generated symmetric keys. These symmetric keys are, in turn, encrypted using a hierarchical identity-based encryption (HIBE) scheme whereby the key encryption keys are derived from generic role identifiers. By doing so, our approach offers a means of decoupling service requestors and service providers. This is because the requestors (potentially clients/agents acting on behalf of users in the form of web services) can be configured to retrieve the encrypted resources automatically based on role information. Only requestors who possess the correct asymmetric decryption keys (which correspond to role identifiers assigned by their home organisations) would be able to recover the symmetric keys and access the resources. In other words, our access control approach can be deployed without binding to any particular PEP and PDP, and it does not rely on SAML-style access control techniques.

However, we remark that our approach is only practical when generic role identifiers are used. The more fine-grained the identifiers, for example using specific policies or attributes, the greater the number of encrypted resources that the service providers have to produce. Hence, this would result in increased storage requirements and a less scalable authorisation framework.

It is worth noting that encryption-based access control techniques are not new. Our approach is closely related to policy-based encryption [3, 60] and attribute-based encryption [24, 49]. The central idea of these proposals is the use of a thresholding primitive to control access to some data (through encryption), whereby only users who fulfill  $k$ -of- $n$  policies or who have  $k$ -of- $n$  attributes can access the data (through decryption). These proposals present constructions of more expressive cryptographic schemes in terms of policy or attribute specification and enforcement, without

dealing with the underlying principal mapping issue. Moreover, as pointed out earlier, encryption based on policy or attribute information seems to be a less practical solution in the context of web services.

## 4.5 Semantic Public Keys

In recent years, semantic web services [6, 36] seem to have received almost as much, if not more, attention as web services do from the research community. While web services are concerned with well-defined, reusable software components that perform specific tasks across organisational boundaries via standardised web-based and message-oriented mechanisms, semantic web services are related to formal descriptions of properties, capabilities, interfaces and effects of web services in an unambiguous, machine-understandable form. In other words, semantic web services are web services augmented with automated service discovery, composition and invocation, and dynamic binding.

Since identity-based public keys are self-describing, identifiers used to construct public keys can be generalised to provide machine-readable semantics so that public keys can be interpreted by software agents or web services automatically. This fits nicely and naturally with the notion of semantic web services. By including meaningful information, such as a validity period, a task description of a workflow process, or data type used in a process, in identifiers, public keys can be related to non-cryptographic processes or services in a natural way.

The use of “semantic public keys” potentially allow automation of public key generation and dynamic binding of keys with processes or services. For example, IBC provides a natural way of encrypting messages which can only be decrypted after a time determined by the sender [12, 37]. This can be achieved by adding a release time to an identifier when its associated public key is constructed. The corresponding private key can only be obtained from the associated TA after the specified release time. An example application is scheduled online payments, in which case a user’s credit card or bank account details can be encrypted with identifiers (public keys) including release dates. We envisage that web services can be configured to extract the release dates, compare them with a current date, and release the credit card or bank account details when there is a match, for payment purposes.

## 5. XKMS FOR IBC

Currently, XKMS assumes that security mechanisms for web services are supported by a conventional certificate-based PKI. In this section, we consider what services, which we collectively call ID-XKMS, would need to be provided if identity-based cryptography were to be used instead. We identify five ID-XKMS services: Locate, Validate, Obtain, Revoke and Recover. Of these, Locate, Validate and Revoke can re-use the corresponding XKMS services. We briefly consider these services, explain how hierarchical IBC concepts can be used to provide ID-XKMS, and then describe the Obtain and Recover services in more detail.

In IBC/HIBC, a public key can be computed and used on-the-fly using the associated cryptographic system parameters, without locating and checking the validity of any certificates. Although the identity-based approach is certificate-free, authentic parameter sets must be made available to the relevant parties. This can be achieved by storing these

parameter sets in a registry service or bootstrapping them into the system.<sup>1</sup> Alternatively, these parameter sets can be stored in standard X.509 certificates, as suggested in [61]. In this case, we envisage that the Locate and Validate services, as defined in X-KISS [26], can be used to obtain and check the authenticity of these parameter sets.

In the identity-based setting, it is common to adopt an automated key expiry approach; thus, the identity-based approach does not usually rely on an explicit key revocation mechanism (as discussed in Section 3). The lifetime of a key pair can be set to some acceptable window of exposure, typically a day or a week, which will be determined by the specific application for which the key pair will be used.

Nevertheless, in order to provide for more fine-grained revocation, we envisage that an identifier revocation list (IRL) [17], analogous to a certificate revocation list (CRL) [28], or other existing key revocation mechanisms, such as Online Certificate Status Protocol (OCSP) [40], can be employed or adapted for use in ID-XKMS. In other words, the Revoke service would implement such functionality in a similar way to the corresponding service in XKMS, if required.

## 5.1 HIBC and ID-XKMS

We assume that web services are hosted by an organisation, and that service requesters are authenticated (using a shared secret such as a password) by some authentication service hosted by an organisation. The authentication service will also act as a server for ID-XKMS. We assume that an organisation obtains certification of a set of system parameters from a third party provider of IBC services. Within this framework, organisations are level 0 TAs, web services and ID-XKMS servers are level 1 entities, and clients (service requesters) are level 2 entities.<sup>2</sup>

We assume the use of an automated key expiry approach in which clients are issued short-lived identity-based public/private key pairs on a daily basis. This is comparable to a certificate-based revocation mechanism which makes use of a CRL that is updated daily. We remark that although XML clients use short-term keys, we envisage that ID-XKMS servers would obtain their private keys from the TAs less frequently. This implies the need for a more fine-grained key revocation mechanism, such as IRLs, for entities above the clients in the hierarchy.

## 5.2 The Obtain Service

The Obtain service allows a client to obtain a private key associated with a public identifier. The client requests the key from the ID-XKMS server, which runs the EXTRACT algorithm and returns the key to the client. Since the outcome of an Obtain service request is the provision of a private key, our Obtain service is analogous to the Register and Reissue services specified in X-KRSS [26].

When a client wishes to obtain the private key corresponding to an identifier, she submits an Obtain request message, an example of which is shown in Figure 2. In the Obtain

<sup>1</sup>The latter approach is analogous to incorporating root CA certificates within a web browser, for example, in the existing certificate-based approach.

<sup>2</sup>We can also envisage a hierarchy with an extra level, in which an umbrella organisation for a group of related organisations (a federation), acts as a level 0 entity, individual organisations acts as level 1 entities, etc.

---

```

01 <?xml version="1.0" encoding="utf-8"?>
02 <ObtainRequest Id="I1494ac4351b7de5c174d455b7000e18f"
03   Service="http://www.example.org/XKMS"
04   xmlns="http://www.w3.org/2005/07/xkms#">
05   <RespondWith>http://www.w3.org/2005/07/xkms#PrivateKey</RespondWith>
06   <PrototypeKeyBinding Id="I269e655567dbae568591c0a06957529e">
07     <ds:KeyInfo>
08       <ds:KeyValue>alice@example.org/20070801</KeyValue>
09     </ds:KeyInfo>
10     <KeyUsage>http://www.w3.org/2002/03/xkms#Signature</KeyUsage>
11     <KeyUsage>http://www.w3.org/2002/03/xkms#Encryption</KeyUsage>
12     <KeyUsage>http://www.w3.org/2002/03/xkms#Exchange</KeyUsage>
13   </PrototypeKeyBinding>
14   <Authentication>
15     <KeyBindingAuthentication>
16       <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
17         <SignedInfo>
18           <CanonicalizationMethod
19             Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
20           <SignatureMethod
21             Algorithm="http://www.w3.org/2000/09/xmldsig#hmac-sha1"/>
22           <Reference URI="#I269e655567dbae568591c0a06957529e">
23             <Transforms>
24               <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
25             </Transforms>
26             <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
27             <DigestValue>WcbpkifxJ1zIJ+V6/knZgxRhr34=</DigestValue>
28           </Reference>
29         </SignedInfo>
30         <SignatureValue>iJSKM+98hj5ae+btC2WjwBYP+/k=</SignatureValue>
31       </Signature>
32     </KeyBindingAuthentication>
33   </Authentication>
34 </ObtainRequest>

```

---

Figure 2: An ID-XKMS Obtain request.

request message, the public key identifier (`alice@example.com/20070801`) is specified in the `KeyValue` element (line 08). Note that Alice’s identifier is the concatenation of her email address and a date, indicating the validity of the matching private key.

As in the Register service in XKMS, Alice must authenticate herself to the ID-XKMS server as part of the Obtain request. In our example, she computes a MAC (line 27, Figure 2) of the request message based on the secret she shares with the ID-XKMS server.

Once Alice is authenticated, the ID-XKMS server extracts the requested private key, encrypts it and sends it to Alice in an Obtain response message. Figure 3 illustrates a typical response to the request shown in Figure 2. The symmetric encryption algorithm used to encrypt the private key in this example is AES-128 in CBC mode (line 23, Figure 3). As in standard XKMS, the AES encryption key could be derived from the shared secret [26].

### 5.3 The Recover Service

The ID-XKMS Recover service, like the Recover service defined in X-KRSS, is used to obtain past private keys that may have been lost. Our approach, being based on IBC, has built-in key recovery capability since a private key can always be extracted by a TA based on a given identifier. This may be a useful in situations where a client has lost or unintentionally deleted his private key, which is needed to recover some data in encrypted form. Moreover, since our identity-based approach makes use of short-lived private

keys, the Recover service provides a convenient means of accessing old encrypted data.

Our Recover service essentially uses the same schema as the Obtain service, except the `ObtainRequest` element in the Obtain service is replaced by the `RecoverRequest` element. Executing the ID-XKMS Recover service involves a stringent security check, as in the XKMS Recover service [26]. If Alice, for some reason, wishes to recover a private key, she then must contact her ID-XKMS server (her authentication server in this case). The ID-XKMS server, after authenticating Alice, will issue a new one-time secret which is only used to encrypt the recovered key. Once she has recovered the desired private key, the shared secret used to provide authentication in the Obtain service should be changed to prevent malicious use of a compromised shared secret.

## 6. CONCLUSIONS

Identity-based cryptography is emerging as a serious contender to conventional, certificate-based public key cryptography, and has some attractive properties that seem to be well suited to message-level security, a crucial aspect of web services.

We have proposed and discussed how various identity-based cryptographic techniques can be used to simplify public key distribution and access control, and to secure XML messages in a more lightweight and clean way compared to certificate-based approaches. We also proposed ID-XKMS, a suite of key management services suited to the deployment of identity-based cryptography in web services.

---

```

01 <?xml version="1.0" encoding="utf-8"?>
02 <ObtainResult xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
03   Id="I2eb0b29bf38eeecfc5f099c8ca149f98"
04   Service="http://www.example.org/XKMS"
05   ResultMajor="http://www.w3.org/2005/07/xkms#Success"
06   RequestId="I1494ac4351b7de5c174d455b7000e18f"
07   xmlns="http://www.w3.org/2005/07/xkms#">
08   <KeyBinding Id="Ia500663f4e4e578447407a38b9049c8b">
09     <ds:KeyInfo>
10       <ds:KeyValue>alice@example.org/20070801</KeyValue>
11     </ds:KeyInfo>
12     <KeyUsage>http://www.w3.org/2002/03/xkms#Signature</KeyUsage>
13     <KeyUsage>http://www.w3.org/2002/03/xkms#Encryption</KeyUsage>
14     <KeyUsage>http://www.w3.org/2002/03/xkms#Exchange</KeyUsage>
15     <Status StatusValue="http://www.w3.org/2002/03/xkms#Valid">
16       <ValidReason>http://www.w3.org/2002/03/xkms#IssuerTrust</ValidReason>
17       <ValidReason>http://www.w3.org/2002/03/xkms#ValidityInterval</ValidReason>
18     </Status>
19   </KeyBinding>
20   <PrivateKey>
21     <xenc:EncryptedData>
22       <xenc:EncryptionMethod
23         Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
24       <xenc:CipherData>
25         <xenc:CipherValue>
26           kEODQrnoIvGZLa0Z8UQ7jnI92W0TySdxylFL2ZvEHad1UKKN2KtW4zs7nhcQ6Tf
27           6gjYLNXX9ztkNgCKCZRzI4TXOM2khtPhxTv83rfV0hlx1mtRjliFdDbiWInrCW7
28           7IPgMEIkEa1oXoKcYb1pUw+W9xzeTp4hTx16izBqC9aWNSYJrT1AvX/Xa+oY8F4
29           p+YGGg0Svn9Cb2h1Va8Ytb3ntqWafSE+0/Yy0HYGCIIsaeYXV9YFN8A+fw
30         </xenc:CipherValue>
31       </xenc:CipherData>
32     </xenc:EncryptedData>
33   </PrivateKey>
34 </ObtainResult>

```

---

Figure 3: An ID-XKMS Obtain response.

It does seem that identity-based cryptography offers a number of potential advantages for web services security. Nevertheless, further research is needed to unearth other potential advantages of identity-based techniques and to identify the associated practical and implementation issues. Note that identity-based cryptographic techniques using pairings are currently undergoing standardisation through the IEEE P1363 Working Group. That said, a wider set of standardisation efforts will be required in order to ensure the rapid adoption of identity-based cryptography.

The key escrow that is inherent in identity-based cryptography may not be desirable in some applications. In future work, we will explore the use of certificateless public key cryptography (CL-PKC) [1] and threshold cryptographic techniques [57] to eliminate key escrow in web services which make use of identity-based cryptography.

## 7. ACKNOWLEDGEMENT

The research in this paper was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) through Grant EP/D051878/1.

## 8. REFERENCES

- [1] S.S. Al-Riyami and K.G. Paterson. Certificateless public key cryptography. In C.S. Lai, editor, *Advances in Cryptology - Proceedings of ASIACRYPT 2003*, pages 452–473. Springer-Verlag LNCS 2894, November 2003.
- [2] G. Alonso, F. Casati, H. Kuno, and V. Machiraju. *Web Services: Concepts, Architectures and Applications*. Springer-Verlag, Berlin, 2004.
- [3] W. Bagga and R. Molva. Policy-based cryptography and applications. In A.S. Patrick and M. Yung, editors, *Proceedings of the 9th International Conference on Financial Cryptography and Data Security (FC 2005)*, pages 72–87. Springer-Verlag LNCS 3570, February 2005.
- [4] M. Bellare, T. Kohno, and C. Namprempre. Breaking and provably repairing the SSH authenticated encryption scheme: A case study of the Encode-then-Encrypt-and-MAC paradigm. *ACM Transactions on Information and System Security*, 7(2):206–241, May 2004.
- [5] M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In T. Okamoto, editor, *Advances in Cryptology - Proceedings of ASIACRYPT 2000*, pages 531–545. Springer-Verlag LNCS 1976, December 2000.
- [6] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.
- [7] M. Blaze, J. Feigenbaum, J. Ioannidis, and A.D. Keromytis. The KeyNote trust-management system version 2. *The Internet Engineering Task Force (IETF)*, RFC 2704, September 1999.
- [8] D. Boneh, X. Boyen, and E. Goh. Hierarchical identity based encryption with constant size ciphertext. In

- R. Cramer, editor, *Advances in Cryptology - Proceedings of EUROCRYPT 2005*, pages 440–456. Springer-Verlag LNCS 3494, May 2005.
- [9] D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In J. Kilian, editor, *Advances in Cryptology - Proceedings of CRYPTO 2001*, pages 213–229. Springer-Verlag LNCS 2139, August 2001.
- [10] X. Boyen and B. Waters. Anonymous hierarchical identity-based encryption (without random oracles). In C. Dwork, editor, *Advances in Cryptology - Proceedings of CRYPTO 2006*, pages 290–307. Springer-Verlag LNCS 4117, August 2006.
- [11] M. Burmester and Y. Desmedt. Identity-based key infrastructures. In *Proceedings of the IFIP TC11 19th International Information Security Conference (SEC 2004)*, pages 167–176. Kluwer, August 2004.
- [12] L. Chen, K. Harrison, D. Soldera, and N.P. Smart. Applications of multiple trust authorities in pairing based cryptosystems. In G.I. Davida, Y. Frankel, and O. Rees, editors, *Proceedings of Infrastructure Security Conference (InfraSec 2002)*, pages 260–275. Springer-Verlag LNCS 2437, October 2002.
- [13] R. Chinnici, J. Moreau, A. Ryman, and S. Weerawarana. *Web Services Description Language (WSDL) Version 2.0*, June 2007. Available at <http://www.w3.org/TR/wsdl20/>.
- [14] K. Chiu, M. Govindaraju, and R. Bramley. Investigating the limits of SOAP performance for scientific computing. In *Proceedings of 11th IEEE Symposium on High Performance Distributed Computing*, pages 246–254. IEEE Computer Society Press, July 2002.
- [15] L. Clement, A. Hatley, C.v. Riegen, and T. Rogers, editors. *Universal Description Discovery and Integration (UDDI) Version 3.0*. OASIS Standard 200502, February 2005.
- [16] J. Crampton and H.W. Lim. *Role Signatures for Access Control in Grid Computing*. Royal Holloway, University of London, Technical Report RHUL-MA-2007-2, May 2007.
- [17] J. Crampton, H.W. Lim, K.G. Paterson, and G. Price. A certificate-free grid security infrastructure supporting password-based user authentication. In *Proceedings of the 6th Annual PKI R&D Workshop 2007*. NIST, 2007.
- [18] T. Dierks and C. Allen. The TLS protocol version 1.0. *The Internet Engineering Task Force (IETF)*, RFC 2246, January 1999.
- [19] D. Eastlake, J.M. Reagle, and D. Solo. (Extensible Markup Language) XML-Signature syntax and processing. *The Internet Engineering Task Force (IETF)*, RFC 3275, March 2002.
- [20] D. Eastlake and J.M. Reagle, editors. *XML Encryption Syntax and Processing*, December 2002. Available at <http://www.w3.org/TR/xmlenc-core/>.
- [21] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. SPKI certificate theory. *The Internet Engineering Task Force (IETF)*, RFC 2693, September 1999.
- [22] C. Ellison and B. Schneier. Ten risks of PKI: What you’re not being told about public key infrastructure. *Computer Security Journal*, 16(1):1–7, 2000.
- [23] C. Gentry and A. Silverberg. Hierarchical ID-based cryptography. In Y. Zheng, editor, *Advances in Cryptology - Proceedings of ASIACRYPT 2002*, pages 548–566. Springer-Verlag LNCS 2501, December 2002.
- [24] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In R.N. Wright, S.D.C. di Vimercati, and V. Shmatikov, editors, *Proceedings of the 13th ACM Computer and Communications Security Conference (CCS 2006)*, pages 89–98. ACM Press, October 2006.
- [25] M. Gudgin, M. Hadley, N. Mendelsohn, J. Moreau, H.F. Nielsen, A. Karmarkar, and Y. Lafon. *Simple Object Access Protocol (SOAP) Version 1.2*, April 2007. Available at <http://www.w3.org/TR/soap/>.
- [26] P. Hallam-Baker and S.H. Mysore, editors. *XML Key Management Specification (XKMS 2.0)*, June 2005. Available at <http://www.w3.org/TR/xkms2/>.
- [27] J. Horwitz and B. Lynn. Towards hierarchical identity-based encryption. In L.R. Knudsen, editor, *Advances in Cryptology - Proceedings of EUROCRYPT 2002*, pages 466–481. Springer-Verlag LNCS 2332, May 2002.
- [28] R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile. *The Internet Engineering Task Force (IETF)*, RFC 3280, April 2002.
- [29] M.N. Huhns and M.P. Singh. Service-oriented computing: Key concepts and principles. *IEEE Internet Computing*, 9(1):75–81, January/February 2005.
- [30] A. Joux. A one round protocol for tripartite Diffie-Hellman. In W. Bosma, editor, *Proceedings of 4th Algorithmic Number Theory Symposium (ANTS-IV)*, pages 385–394. Springer-Verlag LNCS 1838, July 2000.
- [31] J. Kemp, S. Cantor, P. Mishra, R. Philpott, and E. Maler, editors. *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) Version 2.0*. OASIS Standard 200503, March 2005.
- [32] H. Krawczyk. The order of encryption and authentication for protecting communications (or: How secure is SSL?). In J. Kilian, editor, *Advances in Cryptology - Proceedings of CRYPTO 2001*, pages 310–331. Springer-Verlag LNCS 2139, August 2001.
- [33] N. Li, J.C. Mitchell, and W.H. Winsborough. Design of a role-based trust management framework. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pages 114–130. IEEE Computer Society Press, May 2002.
- [34] H.W. Lim and K.G. Paterson. Identity-based cryptography for grid security. In H. Stockinger, R. Buyya, and R. Perrott, editors, *Proceedings of the 1st IEEE International Conference on e-Science and Grid Computing (e-Science 2005)*, pages 395–404. IEEE Computer Society Press, 2005.
- [35] D.S. Linthicum. *Enterprise Application Integration*. Addison-Wesley Professional, Boston, 1999.
- [36] S.A. McIlraith, T.C. Son, and H. Zeng. Semantic web services. *IEEE Intelligent Systems*, 16(2):46–53, March/April 2001.

- [37] M.C. Mont, K. Harrison, and M. Sadler. The HP time vault service: Exploiting IBE for timed release of confidential information. In *Proceedings of the 12th ACM International Conference on World Wide Web (WWW 2003)*, pages 160–169. ACM Press, May 2003.
- [38] T. Moses, editor. *eXtensible Access Control Markup Language (XACML) Version 2.0*. OASIS Standard 200502, February 2005.
- [39] T.J. Mowbray and R. Zahavi. *The essential CORBA: systems integration using distributed objects*. John Wiley & Sons, New York, 1995.
- [40] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. Internet X.509 public key infrastructure online certificate status protocol (OCSP). *The Internet Engineering Task Force (IETF)*, RFC 2560, June 1999.
- [41] A. Nadalin, M. Goodner, M. Gudgin, A. Barbir, and H. Granqvist, editors. *WS-SecureConversation 1.3*. OASIS Standard 200703, March 2007.
- [42] A. Nadalin, C. Kaler, R. Monzillo, and P. Hallam-Baker, editors. *Web Services Security: SOAP Message Security 1.1 (WS-Security 2004)*. OASIS Standard 200602, February 2006.
- [43] B.C. Neuman and T. Ts'o. Kerberos: An authentication service for computer networks. *IEEE Communications*, 32(9):33–38, September 1994.
- [44] E. Newcomer. *Understanding Web Services: XML, WSDL, SOAP, and UDDI*. Addison-Wesley Professional, Boston, 2002.
- [45] Mark O'Neill. *Web Services Security*. McGraw-Hill, New York, 2003.
- [46] K.G. Paterson. Cryptography from pairings. In I.F. Blake, G. Seroussi, and N.P. Smart, editors, *Chapter 10 of Advances in Elliptic Curve Cryptography*, pages 215–251, Cambridge, 2005. Cambridge University Press, LMS 317.
- [47] K.G. Paterson and G. Price. A comparison between traditional public key infrastructures and identity-based cryptography. *Information Security Technical Report*, 8(3):57–72, 2003.
- [48] K.G. Paterson and A.K.L. Yau. Cryptography in theory and practice: The case of encryption in IPsec. In S. Vaudenay, editor, *Advances in Cryptology - Proceedings of EUROCRYPT 2006*, pages 12–29. Springer-Verlag LNCS 4004, May 2006.
- [49] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters. Secure attribute-based systems. In R.N. Wright, S.D.C. di Vimercati, and V. Shmatikov, editors, *Proceedings of the 13th ACM Computer and Communications Security Conference (CCS 2006)*, pages 99–112. ACM Press, October 2006.
- [50] G. Price. PKI challenges: An industry analysis. In J. Zhou, M-C. Kang, F. Bao, and H-H. Pang, editors, *Proceedings of the 4th International Workshop for Applied PKI (IWAP 2005)*, pages 3–16. Volume 128 of FAIA, IOS Press, 2005.
- [51] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.
- [52] J. Rosenberg and D. Remy. *Securing Web Services with WS-Security: Demystifying WS-Security, WS-Policy, SAML, XML Signature, and XML Encryption*. Sams, Indiana, 2004.
- [53] W. Rosenberry, D. Kenney, and G. Fisher. *Understanding DCE*. O'Reilly and Associates, Sebastopol, 1992.
- [54] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing. In *Proceedings of the 2000 Symposium on Cryptography and Information Security (SCIS 2000)*, January 2000.
- [55] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, February 1996.
- [56] J. Schaad and R. Housley. Advanced Encryption Standard (AES) key wrap algorithm. *The Internet Engineering Task Force (IETF)*, RFC 3394, September 2002.
- [57] A. Shamir. How to share a secret. *Communications of the ACM*, 22(1):612–613, January 1979.
- [58] A. Shamir. Identity-based cryptosystems and signature schemes. In G.R. Blakley and D. Chaum, editors, *Advances in Cryptology - Proceedings of CRYPTO '84*, pages 47–53. Springer-Verlag LNCS 196, August 1985.
- [59] A. Singhal and T. Winograd. *Guide to Secure Web Services (Draft)*. NIST, Special Publication 800-95, September 2006. Available at <http://csrc.nist.gov/publications/drafts/Draft-SP800-95.pdf>.
- [60] N.P. Smart. Access control using pairing based cryptography. In M. Joye, editor, *Proceedings of the RSA Conference: Topics in Cryptology - the Cryptographers' Track (CT-RSA 2003)*, pages 111–121. Springer-Verlag LNCS 2612, April 2003.
- [61] D.K. Smetters and G. Durfee. Domain-based administration of identity-based cryptosystems for secure email and IPSEC. In *Proceedings of 12th USENIX Security Symposium*, pages 215–229, August 2003.
- [62] M.R. Thompson, A. Essiari, and S. Mudumbai. Certificate-based authorization policy in a PKI environment. *ACM Transactions on Information and System Security*, 6(4):566–588, November 2003.
- [63] B. Waters. Efficient identity-based encryption without random oracles. In R. Cramer, editor, *Advances in Cryptology - Proceedings of EUROCRYPT 2005*, pages 114–127. Springer-Verlag LNCS 3494, May 2005.
- [64] T. Yu, S. Hartman, and K. Raeburn. The perils of unauthenticated encryption: Kerberos version 4. In *Proceedings of Symposium on Network and Distributed System Security (NDSS 2004)*. The Internet Society, 2004.