Alternative Security Architectures for e-Science

Jason Crampton, Hoon Wei Lim, Kenneth G. Paterson, and Geraint Price
Information Security Group
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK

Abstract

There have been recent proposals in search of alternative security architectures to PKIs for grid application and e-Science. The application of identity-based cryptography (IBC) in designing a grid security architecture seems to be interesting because of its attractive properties, such as, being certificate-free and having small key sizes. In this paper, we discuss our latest research findings of the identity-based approach in designing a grid security architecture. These include a performance analysis of the identity-based approach, and the application of identity-based secret public keys in designing a password-based version of the standard TLS protocol used in MyProxy.

1 Introduction

The majority of current grid security implementations are based on public key infrastructure (PKI) [6, 15], and the operational grid developed as part of the UK e-Science project is no exception. However, large-scale PKIs are known to have many problems which have hindered the widespread adoption of PKI technology [8, 11]. These include cost, scalability (both of registration and key management processes), revocation, management of client private keys, and support for dynamic security policies.

In addition to generic PKI problems, grid-specific security requirements bring further concerns. The Globus Toolkit (GT) [5], the de facto standard for building grids, includes the specification of a security architecture, the Grid Security Infrastructure (GSI) [6]. This in turn makes use of proxy certificates [16], in addition to the standard X.509 public key certificates, to support single sign-on and delegation services. The dependence on proxy certificates causes many performance issues. Their use leads to frequent and computationally expensive RSA key-pair generation, and the consumption of bandwidth and processing power for transmitting and checking the lengthy certificate chains that result. Moreover, the delegation protocol that is used involves a round-trip between a delegator and a delegation target, and consequent delay. The set of cryptographic algorithms that can be used is quite limited, with RSA signatures being dominant. These issues may not be serious problems for today’s grid environments, but they may limit the spread of grid technology to pervasive and mobile environments, where devices lack computational power and the communication networks have limited bandwidth.

Independent of grid computing, a variant of traditional public key technologies called identity-based cryptography (IBC) [4, 13] has recently received considerable attention. Through IBC, an identifier which represents a user can be transformed into his public key and used on-the-fly without any certificate checking. The potential of IBC to provide greater flexibility to entities within a security infrastructure and its certificate-free approach may well match the dynamic qualities of grid environments. We proposed a fully identity-based key infrastructure for grid (IKIG) [9] which meets the security requirements of the GSI. The proposal makes use of both long-term and short-term identity-based keys, by exploiting some properties from hierarchical identity-based cryptography (HIBC) [7]. More details about IKIG will be presented in Section 2.

In this paper, we intend to demonstrate our latest results of some follow-on work from [9]. These include: (i) performance analysis based on actual implementations of the cryptographic schemes adopted in [9], and (ii) the application of identity-based secret public keys in designing a password-based TLS protocol, which in turn seems to be suited to the authentication protocol used by MyProxy [3].

2 Identity-Based Key Infrastructure for Grid

2.1 Overview

One motivation for the proposal of an identity-based key infrastructure for grid (IKIG) of [9] is the attractive properties of IBC. These include:
Identity-based: The use of identity-based public keys in IBC allows any entity’s public key to be generated and used on-the-fly;

Certificate-free: IBC does not make use of certificates since public keys can be computed based on some public identifiers; and

Small key sizes: Since identity-based cryptographic schemes use pairings which are, in turn, based on elliptic curves, they can have smaller key sizes than more conventional public key cryptosystems such as RSA.

By exploiting some properties from HIBC, IKIG facilitates the creation and usage of identity-based proxy credentials in a very natural way. These identity-based proxy credentials, in turn, are needed to support features that match those provided by the GSI.

In the IKIG setting, the roles of a Certificate Authority (CA) in the current PKI-based GSI has been replaced by a Trusted Authority (TA). The TA’s roles including acting as the Private Key Generator (PKG) and supporting other user-related administration. Figure 1 shows the hierarchical setting of HIBC that matches the hierarchical relationships of various entities within a grid environment, with the TA at level 0, user at level 1 and user proxy at level 2.

Figure 1. A hierarchical structure of entities in the IKIG setting.

We note that proxy credentials must be used for secure job submissions in order to match the requirements of the GSI. These short-term credentials are used in security services such as mutual authentication and delegation. The TA distributes long-term private keys to its users (and resource providers) at level 1, who in turn generate short-term private keys for their own proxies at level 2, as illustrated in Figure 1.

IKIG has the following features:

1. Single sign-on: As with the GSI, our IKIG proposal supports single sign-on through the use of identity-based proxy credentials. Since a user’s short-term public keys are based on some predictable identifiers and the matching short-term private keys are stored locally at the user side, user authentication can be performed without any physical intervention from the users and without the need for certificates.

2. Mutual authentication and key agreement: IKIG also supports a certificate-free authenticated key agreement protocol based on the TLS handshake. Our protocol allows mutual authentication and session key establishment between two entities in a more lightweight manner than the traditional TLS as it has small key sizes and requires no certificates.

3. Delegation: We propose a non-interactive delegation protocol which works in the same way as in the GSI, in the sense that the delegator signs a new public key of the delegation target. In addition, IKIG allows partial aggregation of signatures. This can be useful when the length of the delegation chain increases. This is a new feature not available in the GSI.

Users in the IKIG setting do not need to obtain short-term private keys from their respective PKGs. This is because the users themselves act as PKGs for their local proxy clients. Thus short-term private key distribution is not an issue in IKIG. This contrasts favourably with conventional applications of IBC, where private key distribution is a complicating factor.

2.2 Performance Analysis

We examined the efficiency of IKIG in terms of its communication and computational overheads.

Since identity-based cryptographic schemes have small key sizes and does not rely on the use of certificates, it is obvious that IKIG consumes much less bandwidth than the RSA-based GSI does. More details of the performance trade-offs in communication costs between the GSI and IKIG can be found in [9].

To obtain more accurate real performance figures in terms of computation times of the underlying cryptographic schemes of IKIG, we have recently implemented the Gentry-Silverberg hierarchical encryption and signature schemes (HIBE/HIBS). Our implementation was based on the MIRACL library [14], written in C/C++ and compiled with Microsoft Visual C++ 6.0. By using known optimisation techniques, we observed that the computational cost of IKIG is comparable to the GSI, even though pairing computations are generally perceived to be computationally intensive. The details of the computation timings of the cryptographic schemes used in the GSI and IKIG are shown in Table 1.

It is worth noting that recent results (see for example [2, 12]) have shown improvements in computing pairings with the use of various optimisation techniques and
Table 1. Performance trade-offs in computation times (in milliseconds) between the GSI and the IKIG settings on a Pentium IV 2.4 GHz machine.

<table>
<thead>
<tr>
<th>Operation</th>
<th>GSI</th>
<th>Time</th>
<th>IKIG</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RSA</td>
<td></td>
<td>HIBE/HIBS</td>
<td></td>
</tr>
<tr>
<td>Key generation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a.) Long-term</td>
<td>1 GEN</td>
<td>149.90</td>
<td>1 EXT</td>
<td>1.69</td>
</tr>
<tr>
<td>(b.) Short-term</td>
<td>1 GEN</td>
<td>34.85</td>
<td>1 EXT</td>
<td>1.74</td>
</tr>
<tr>
<td>Authenticated key agreement</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a.) Requestor</td>
<td>1 1024-bit VER</td>
<td>2.67</td>
<td>1 ENC, 1 SIG</td>
<td>8.79</td>
</tr>
<tr>
<td></td>
<td>1 512-bit ENC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 512-bit SIG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 512-bit VER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(b.) Resource</td>
<td>1 1024-bit VER</td>
<td>2.67</td>
<td>1 DEC, 1 VER</td>
<td>20.16</td>
</tr>
<tr>
<td></td>
<td>1 512-bit DEC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 512-bit VERs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delegation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a.) Delegator</td>
<td>1 512-bit SIG</td>
<td>1.86</td>
<td>1 SIG</td>
<td>3.35</td>
</tr>
<tr>
<td></td>
<td>1 512-bit VER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(b.) Delegation target</td>
<td>1 GEN</td>
<td>35.63</td>
<td>1 EXT</td>
<td>1.74</td>
</tr>
<tr>
<td></td>
<td>1 512-bit SIG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c.) Verifier</td>
<td>3 512-bit VERs</td>
<td>0.84</td>
<td>1 VER</td>
<td>8.42</td>
</tr>
</tbody>
</table>

GEN = RSA parameter generation  EXT = HIBE/HIBS private key extraction
ENC = Encryption              DEC = Decryption
SIG = Signing                  VER = Verification

this should give hope to faster HIBE and HIBS schemes in the near future. On the other hand, we remark that it is unlikely that significant algorithmic improvements for RSA computations will be forthcoming, since this field has been intensively researched for many years.

2.3 Identity-Based Secret Public Keys

In a password-based authentication protocol, a secret public key is a standard public key which can be generated by a user or an authentication server, and is known only to them but is kept secret from third parties. A secret public key, when encrypted with a user’s password, should serve as an unverifiable text. This may significantly increase the difficulty of password guessing even if it is a poorly chosen password as an attacker has no way to verify if he has made the correct guess. However, it may not be easy to achieve unverifiability of text by simply performing naive symmetric encryption on public keys of standard types, such as RSA or ElGamal, which contain certain number theoretic structure.

We have recently investigated the use of identity-based secret public keys [10] and designed a password-based TLS protocol which, in turn, seems to fit nicely into MyProxy. The identity-based approach of [10] shows that secret public keys can be constructed in a very natural way using arbitrary random strings, eliminating the structure found in, for example, RSA or ElGamal keys.

An identity-based secret public key protocol can be naturally converted into a password-based version of the standard TLS protocol. The resulting protocol allows passwords to be tied directly to the establishment of secure TLS channels (rather than transmitting plain passwords through secure TLS channels). Such a protocol can be constructed without radical modification to the existing TLS handshake messages. The identity/password-based TLS protocol is directly applicable to securing interactions between grid users and MyProxy in IKIG.

3 Future Work

Certificateless public key cryptography (CL-PKC) [1] offers an interesting combination of features from IBC and standard public key cryptography. In particular, each user selects a public/private key-pair (in addition to a TA-generated private key component that matches an identifier), thereby eliminating the problem of key escrow found
in IBC. Moreover, the public keys do not need to be supported by certificates, so as with IBC, many of the problems associated with certificate management in PKI are eliminated in CL-PKC.

As part of our future work, we plan to study the application of CL-PKC in grid environments. We will develop a security architecture for grid systems based on CL-PKC that parallels our existing work on IKIG. We will examine how CL-PKC can be used to support authentication and establishment of secure communications via a TLS-like protocol. We will also study the key management aspects of our architecture, such as, key updating, key revocation and the use of credential storage systems, such as MyProxy, in our architecture.

Subsequently, we plan to conduct a performance analysis to compare our CL-PKC approach with the existing GSI and IKIG approaches.

References


