

User-friendly and certificate-free grid security infrastructure

Jason Crampton · Hoon Wei Lim ·
Kenneth G. Paterson · Geraint Price

Published online: 10 February 2011
© Springer-Verlag 2011

Abstract Certificate-based public key infrastructures are currently widely used in computational grids to support security services. From a user's perspective, however, certificate acquisition is time-consuming and public/private key management is non-trivial. In this paper, we propose a security infrastructure for grid applications, in which users are authenticated using passwords. Our infrastructure allows a user to perform single sign-on based only on a password, without requiring a public key infrastructure. Moreover, hosting servers in our infrastructure are not required to have public key certificates. Nevertheless, our infrastructure supports essential grid security services, such as mutual authentication and delegation, using public key cryptographic techniques without incurring significant additional overheads in comparison with existing approaches.

A preliminary version of this work appeared in the Proceedings of the 6th Annual PKI R&D Workshop 2007 [18]. The research in this paper was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) through Grant EP/D051878/1. The second author was supported in part by the Singapore National Research Foundation under Research Grant NRF-CRP2-2007-03.

J. Crampton · K. G. Paterson · G. Price
Information Security Group, University of London,
Royal Holloway, London, UK
e-mail: jason.crampton@rhul.ac.uk

K. G. Paterson
e-mail: kenny.paterson@rhul.ac.uk

G. Price
e-mail: geraint.price@rhul.ac.uk

H. W. Lim (✉)
Coding and Cryptography Research Group,
Nanyang Technological University, Singapore, Singapore
e-mail: hoonwei@ntu.edu.sg

1 Introduction

The vision of grid computing [22,24] is to provide easy access to “unlimited” resources, thereby enabling computationally complex tasks to be performed and huge amounts of data to be stored and shared. There has been some suspicion, since the term *grid* was first used more than a decade ago, that grid computing might be another technological vision that turns out to be more hype than substance. Despite that, the vision prevails and the gap between vision and reality is narrowing quickly. This is evident from the large and growing number of grid projects and testbeds worldwide [29]. TeraGrid [55] is one of the pioneering grid projects and as of late 2009 is capable of providing 2 petaflops (2×10^{15} floating-point operations per second) of computing power and more than 60 petabytes (6×10^{16} bytes) of online data storage. Moreover, grid computing has very recently become the underlying core technology of *cloud* computing, which allows users, particularly of business enterprises, to access applications on demand from anywhere in the world [15]. As commercial interest grows in grid computing, security is an issue that will become increasingly important [34].

1.1 Motivations

Currently, the grid security infrastructure (GSI) [23] of the Globus Toolkit (GT) [21] plays an essential role in supporting various grid security services, such as single sign-on, mutual authentication, and delegation. The GSI assumes the use of a certificate-based PKI in which digitally signed certificates are used to provide authenticity of public keys. GSI users are required to possess and manage long-term credentials, typically RSA public/private key pairs. Users' public keys must be certified by a Grid Certificate Authority (CA) and usually presented in the form of X.509 certificates [33].

However, *credential acquisition* is a fairly complicated and time-consuming process. In the UK National Grid Service,¹ for example, obtaining a public key certificate requires the generation of a key pair using a web browser, transmission of the public component to a CA for certification, and a face-to-face meeting with a Registration Authority (RA) to prove user identity and membership in a grid project and to provide justification for grid usage. Studies have shown that users tend to share credentials within their peer groups because of the cumbersome and time-consuming process of credential acquisition (which may take up to few weeks) [6,7]. Clearly, this increases the risk of private key exposure. Therefore, it is highly desirable to develop grid systems in which user credential acquisition and management are simpler.

We note that the GSI is a rather heavyweight apparatus, mainly because of the extensive use of public key certificates and proxy certificates [56]. Various aspects of traditional public key management in PKIs, for example generation, certification and verification of public keys, and distribution of certificates, incur non-trivial overheads. Clearly, a more lightweight security architecture is desirable. This need is amplified by the fact that the availability of mobile/wireless devices, such as smart phones and personal digital assistants (PDAs), has been increased greatly in recent years. Hence, it is very desirable to minimise the communication overheads of any grid security infrastructure if we are to allow mobile devices to efficiently interact with computational grids [16,49].

Orthogonal to the development of grid computing, identity-based public key cryptography (ID-PKC) [13,52] is emerging as an alternative to more traditional certificate-based public key cryptography. This growing maturity of ID-PKC presents an opportunity to reassess the appropriateness of certificate-based PKI as the backbone of the GSI.

Previous research [39,40,42] shows that ID-PKC has a number of properties that make it suitable for use in grid environments. Lim and Paterson proposed a fully identity-based security infrastructure for grid applications [39]. Key management in this approach is simpler than in the PKI-based GSI because it does not use certificates and key sizes are relatively small. This yields substantial efficiency gains: communication overhead for mutual authentication and delegation between two entities can be reduced by up to 90%, when appropriately chosen elliptic curves and system parameters are used [38], for example.

Nevertheless, *key revocation* in the identity-based setting can be complicated. Boneh and Franklin [13] proposed the use of a date concatenated with a user's identifier to achieve automated key expiry. However, this approach has the disadvantage of increasing the workload of the Private

Key Generator (PKG), since the PKG is required to regularly issue private keys to its users.

In addition to the key revocation issue, the security infrastructure of [39] does not appear to be more user-friendly than the PKI-based GSI in terms of credential acquisition and management. This is because users still need to contact the relevant trusted third parties before obtaining their credentials.

The above issues and observations have led us to our investigation into a grid security infrastructure which is not only certificate-free, but also user-friendly in the sense that the infrastructure is “PKI-free” from the user perspective.

1.2 Contributions

In this paper, we propose a password-enabled and certificate-free grid security infrastructure (PECF-GSI). Briefly, our proposal enhances the earlier work of Lim and Paterson [39] so that users are authenticated using *only* passwords. We make use of users' local authentication servers to authenticate users based on their already established usernames/passwords. These servers play a similar, but not identical, role to the MyProxy server in the PKI-based GSI [5,45].² Our approach has the benefit that neither client nor server certificates are required during user authentication. Our proposal also completely removes the need for long-term public keys for end users and hence the need for a revocation mechanism for these public keys too. We still require mechanisms for handling revocation of server public keys, however. Instead, users are given short-lived, identity-based credentials by the authentication server upon successful authentication. All subsequent security services make use of these credentials on behalf of users, without requiring direct user intervention.

Our contributions can be summarised as follows:

- We design a lightweight and user-friendly grid security infrastructure. Our proposal makes use of attractive properties of the identity-based approach, in particular being certificate-free and using small key sizes. Mutual authentication between a user and a server is based on a provably secure password-based authentication protocol that does not rely on certificate verification by end users. Nevertheless, our architecture still provides full support for grid security services, such as single sign-on, mutual authentication, and delegation.
- We employ “just-in-time” issuance of short-lived keys to avoid any complications related to revoking users' long-term public keys in the identity-based setting [13,47]. Our approach is similar to the use of short-lived symmetric keys in Kerberos [44]. This, and the fact that

¹ <http://www.grid-support.ac.uk/>.

² MyProxy is typically used as an online credential repository to provide protection for long-term user private keys.

system parameters of the identity-based primitives do not necessarily need to be pre-distributed or bootstrapped, gives rise to easy, flexible, and user-friendly deployment of ID-PKC. We also show how timely revocation of hosting servers' long-term public keys can be carried out very simply in our architecture.

- We devise an efficient and natural delegation technique that exploits the properties of hierarchical ID-PKC [28]. The mathematical properties of hierarchical ID-PKC allow very efficient credential verification of a delegatee for a particular delegation. A verifier needs *only* to check the credential of the delegatee, instead of having to verify the credentials of the delegatee *and* all of his ancestors along the delegation chain, as in existing proposals [39,58].
- We show how our security infrastructure is easily extended to support grid access control in a natural way using the concept of *role signatures* [17]. Using role signatures, inter-domain principal mappings become simple and user authentication and access control is unified.

1.3 Organisation

The remainder of this paper is organised as follows. In the next section, we present the design of a password-enabled and certificate-free grid security infrastructure. In Sect. 3, we describe specific hierarchical identity-based encryption and signature schemes that can be used to realise our proposal.

Section 4 provides a high-level security analysis of our approach, while Sect. 5 compares our proposal with existing approaches. In Sect. 6, we discuss related work, and finally, we conclude in Sect. 7.

2 Designing PECF-GSI

We begin by describing a threat scenario for grid applications. We then give a conceptual view of PECF-GSI that addresses major security threats in grid applications. We explain how the concept of hierarchical identity-based cryptography is used by PECF-GSI. Moreover, we provide details of the protocols that underpin PECF-GSI and how they are used to support grid security services.

2.1 Threat scenario

Let us assume that a user *A* is a geologist and she wishes to perform a series of analyses of geological survey data relating to oil deposits. Since this is a rather resource-intensive job, she submits it along with some relevant job instructions to a computational grid within a *virtual organisation* (VO) formed by her employer and its business partners in order to

share resources. Once the job has been submitted, it is fed to a job scheduler and the required resources, such as computation power, data storage, and memory space, are located. The job is then executed automatically without any intervention from *A* until its completion. We note that the hosting server which runs the job may delegate parts of the job to other resource providers within the VO for additional resources not available locally. When the geological analysis results are ready, the system informs *A* by email, for example.

In this simple scenario, there is a clear incentive to protect information related to the analyses of oil deposits, because of its commercial sensitivity. Moreover, the computational grid is in fact composed of shared resources contributed by different organisations. Hence, it is essential that only authenticated and authorised users are allowed access to the resources. We summarise some relevant threats as follows:

- Exposure of confidential information related to the data used during job execution or the resulting data from the execution;
- Unauthenticated or unauthorised access to resources, potentially leading to malicious activities or attacks which make use of the resources;
- Insider attacks, such as stealing of business secrets, by employees who have legitimate user accounts in the VO.

Note that there are other threats relevant to the scenario, for example denial-of-service attacks, malware infection, and system intrusion. However, these are beyond the scope of this paper.

In order to address the aforementioned threats, as with the PKI-based GSI [23], PECF-GSI is designed to provide essential grid security services including single sign-on, access control, mutual authentication (typically between a user and a resource provider), and credential delegation.

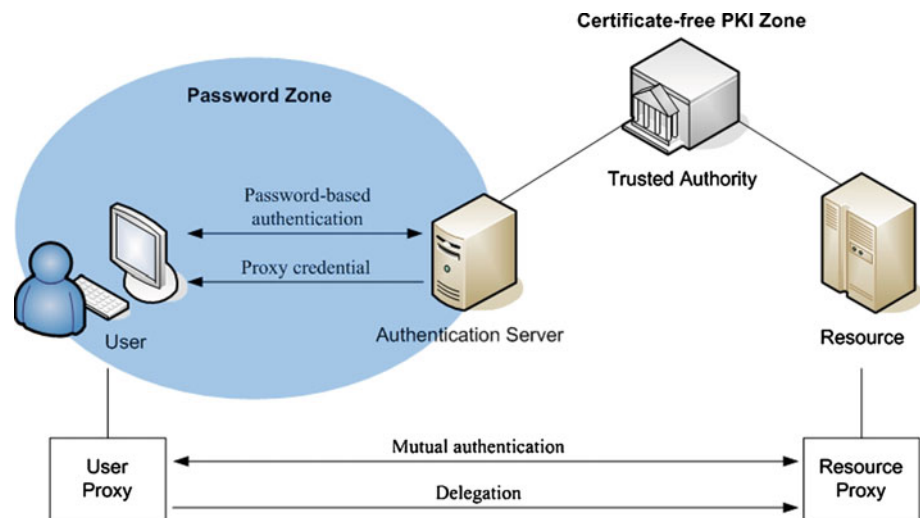
2.2 PECF-GSI overview

PECF-GSI employs a Trusted Authority (TA), instead of a Certificate Authority (CA), as the root of trust within a grid environment. The TA's roles include acting as the PKG in the identity-based setting [13,52] and providing a key management service.

In PECF-GSI, a user's long-term credential is simply a password, which he shares with his institutional authentication server. We assume that the user re-uses the username/password which he has already established with the authentication server beforehand, for example, for accessing an email account or a web portal.

The authentication server is assumed to be accredited by the TA and hosting servers (or resource providers) within the grid environment. Unlike the user, who only has to remember a password, the authentication and hosting servers must

Fig. 1 A conceptual view of PEFC-GSI



obtain the TA's authenticated system parameter set through out-of-band mechanisms. This hybrid approach divides our architecture into two zones: (i) a password (or user-centric) zone, where only passwords are involved, and (ii) a certificate-free PKI (or server-centric) zone, which is hidden from the users' view and makes use of full-strength identity-based public key techniques. This is illustrated in Fig. 1.

As with the current GSI, we make use of proxy credentials when providing grid security services: before a user A submits a job to a resource X , for example, through the Grid Resource Allocation and Management (GRAM) module of the GT [59], she must be in possession of a proxy credential, *i.e.* a short-lived public/private key pair.³ The proxy credential is used by a user proxy \bar{A} , a short-lived agent created by A to perform security services on the user's behalf during a job submission.

\bar{A} signs the job request (with the proxy private key), which is then submitted to X . X verifies \bar{A} 's signed request and checks if A is an authorised user. If the checks are successful, X creates a resource proxy \bar{X} and the associated managed job service. This is followed by mutual authentication and a secure session establishment between \bar{A} and \bar{X} . A may then, at her discretion, delegate her credential through \bar{A} to \bar{X} for later use, using the established secure channel.

2.3 Cryptographic primitives and protocols

We now give an overview of the generic cryptographic schemes and protocols that we employ in PEFC-GSI (descriptions of specific cryptographic schemes are provided in Sect. 3). These include hierarchical identity-based encryption and signature schemes, a password-based TLS protocol

³ Proxy credentials are currently widely used in grid applications to minimise exposure of long-term credentials and to support single sign-on and credential delegation [56].

used to support single sign-on, and a certificate-free TLS protocol to provide mutual authentication between a user and a hosting server.

2.3.1 Hierarchical identity-based cryptography

Identity-based cryptography (ID-PKC) was proposed to simplify public key management and eliminate the use of certificates for attesting public keys [52]. This is achieved by enabling a public key to be derived from an identifier, some public information that can be used to uniquely identify a user, such as user name or email address. The corresponding private key is produced and distributed by a Private Key Generator (PKG). It is assumed that a set of system parameters are used to derive these keys and to perform identity-based encryption and signing operations.

Hierarchical identity-based cryptography, a variant of ID-PKC, was then designed to ease private key distribution in the identity-based setting by having multiple levels of PKG [32]. It is assumed that these PKGs can be arranged in a rooted tree with users located at the bottom of the tree. More specifically, the root PKG, located at level 0, is trusted to produce private keys for entities at level 1, who in turn act as PKGs for entities in their respective domains at level 2, and so on, with users at level t , say, obtaining their private keys from their respective PKG at level $t - 1$. Note that all entities in the hierarchy share the same set of system parameters.

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. Hence, the identifier $\langle ID_1, \dots, ID_t \rangle$ represents an entity at level t whose ancestor at level 1 has identifier ID_1 and whose ancestor at level j has identifier $\langle ID_1, \dots, ID_j \rangle$.

To illustrate the use of hierarchical identity-based cryptography in PEFC-GSI, we now map the entities in Fig. 1, each of which will require some form of credential to interact with

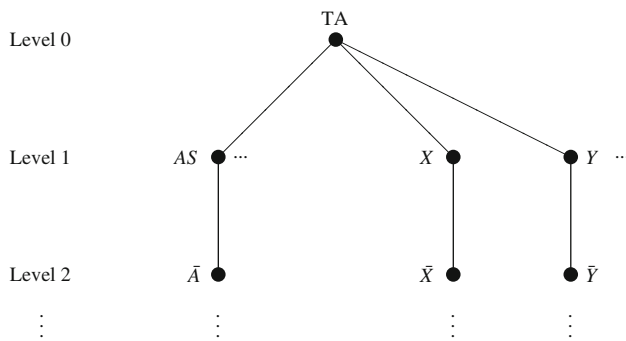


Fig. 2 The hierarchical relationships between entities in PEFCF-GSI

another entity, into the hierarchical identity-based setting. Let AS be an authentication server, A be a user, and X and Y be hosting servers. Then the TA , acting as the root PKG, is a level 0 entity in the hierarchy and issues private keys to AS , X and Y at level 1 based on their identifiers ID_{AS} , ID_X , and ID_Y , respectively. These entities, in turn, issue private keys to entities at level 2, as shown in Fig. 2. Note that A does not possess any long-term credential issued by the TA ; instead, she obtains proxy credentials from AS . Hence, \bar{A} , a user proxy for A , becomes a child of AS , under an identifier of the form $\langle ID_{AS}, ID_A || LT_A \rangle$ where LT_A denotes a lifetime associated with the proxy \bar{A} . On the other hand, \bar{X} , a proxy for X , is a child of X , under an identifier $\langle ID_X, ID_X || LT_X \rangle$ where LT_X denotes a lifetime associated with the proxy \bar{X} . Entities at level 3 and below in our architecture are invoked when handling delegation (see Sect. 2.4.5).

Subsequently, entities in PEFCF-GSI make use of a hierarchical identity-based encryption (HIBE) scheme and a hierarchical identity-based signature (HIBS) scheme when performing mutual authentication and delegation. Details about how HIBE and HIBS schemes are used are provided in Sect. 2.4. Detailed descriptions of concrete HIBE and HIBS schemes can be found in Sect. 3.1.

In this paper, we use a hierarchy with a single TA for ease of exposition. In actual implementations, we can expand the hierarchy of Fig. 2 to support multiple TAs . This can be achieved by adding a root TA at the top of the hierarchy with the TAs becoming level 1 entities. Similarly, lower-level entities are moved down to the next level in the hierarchy. If a single root TA model is not desirable, multiple TAs can also be employed at level 0 of the hierarchy. However, the authentic system parameter set of each TA must be made available to other TAs , which can then distribute the parameter sets to their respective users.

2.3.2 Password-based authentication

In the existing PKI-based GSI setting, MyProxy typically serves as an online key issuing centre for users to obtain

their proxy credentials before submitting job requests. This is analogous to the role played by a domain authentication server in our approach. The MyProxy system makes use of the existing standard TLS protocol [19] to provide mutual authentication between a user and a MyProxy server. This is typically based on the MyProxy server's public key certificate and a password shared by the server and the user. The user authenticates the server by verifying its certificate (and other associated certificates in the certificate chain up to the root CA), whereas the server authenticates the user based on a password that he provides. However, in such a set-up, where the user enters his password only after the secure channel is established, the authentication of the user is not directly tied to the secure channel [54]. This may give a false sense of security if management of certificates of the relevant parties is not handled properly.⁴ Furthermore, the user's password would be compromised if he accepts a bogus server certificate.

These weaknesses in traditional password-based user authentication over TLS has prompted the study of password-based TLS protocols [1, 54]. These eliminate the reliance on certificates and provide stronger security, in the sense that authentication of both client and server is based on proving knowledge of the password and takes place during a single authentication protocol run (rather than in two distinct phases). Our proposal of PEFCF-GSI makes use of such a protocol to provide mutual authentication between a user and her domain authentication server. Further details of the protocol are provided in Sect. 3.2.

2.3.3 Certificate-free key agreement

The current standard (certificate-based) and widely deployed TLS protocol [19] is a key component of the GSI. It is typically based on OpenSSL [46] and used to perform mutual authentication and establish a session key between a user (user proxy) and a resource provider (resource proxy). Thus, it makes sense that our PEFCF-GSI approach adopts TLS-like protocols to minimise the amount of changes if PEFCF-GSI is to be implemented using existing tools.

We propose the use of a TLS protocol in the identity-based setting. In Sect. 3.3, we will show that public key certificates required for the standard TLS protocol can be replaced by identifiers, a key feature of ID-PKC.

2.4 Security services

In this section, we explain how security services mentioned in Sect. 2.1 can be provided through PEFCF-GSI using cryp-

⁴ We have to rely on the user to respond correctly to messages from his web browser, for example, when certificates fail to verify successfully.

tographic schemes and protocols described in the previous subsection.

2.4.1 User authentication

In PECF-GSI, a user A is authenticated by her local authentication server AS using a password-based TLS protocol. Note that a successful run of the protocol also authenticates AS to A based solely on a shared password, without any certificate verification being required.

2.4.2 Single sign-on

Once A and AS have been mutually authenticated and established a secure channel, AS extracts a short-lived public and private key pair $(P_{\bar{A}}, S_{\bar{A}})$ corresponding to an identifier of the form $(ID_{AS}, ID_A || LT_A)$ for A 's proxy \bar{A} . Subsequently, AS sends the following information to A through the secure channel:

- (1) the newly created proxy credential $(P_{\bar{A}}, S_{\bar{A}})$;
- (2) a copy of the TA system parameters;⁵
- (3) an up-to-date Identity Revocation List (IRL).

Upon receiving the proxy credential, A stores the private key $S_{\bar{A}}$ in a local file system accessible by her proxy \bar{A} when necessary. This completes the process of single sign-on by A .

Note that in order to make use of the role signature concept introduced in [17] in our architecture, we envisage that the identifier on which $P_{\bar{A}}$ is based contains A 's username and role information, which is in turn determined by A 's organisation and/or virtual organisation (VO), as well as a lifetime for the identifier. The system parameters that A receives from AS are needed to run HIBE and HIBS schemes when performing mutual authentication and delegation (see Sects. 2.4.4 and 2.4.5).

The IRL in step 3 above is analogous to a CRL in a certificate-based environment. It is used by \bar{A} to check the continuing validity of the identifier of the hosting server to which \bar{A} is going to submit her job. It is worth noting that this approach "forces" the user's proxy into receiving an up-to-date IRL. Additionally, the user's proxy does not have to check the authenticity of the IRL, assuming AS behaves in an honest manner. Moreover, the user herself never directly checks the status of identifiers; rather this task is performed on behalf of the user by her proxy. Upon expiry of the proxy credential, all the information that \bar{A} obtained from AS can be destroyed.

We remark that the whole process of single sign-on does not involve any kind of certificate or parameter verification

from the user's perspective (recall that users are in a password-based zone in our PECF-GSI setting).

2.4.3 Access control

We now describe how the concept of role signatures introduced in [17] can fit nicely into our PECF-GSI approach.

Briefly, role signatures are intended to provide both user authentication and authorisation in a unified way, as well as to simplify user credential verification. The main motivation for using role signatures is to address the well-known inter-domain principal mapping problem [20], which usually arises in open-distributed environments. This is achieved by exploiting a hierarchical namespace within an organisational structure to define role information. By using role information as the input to constructing a public key in the identity-based setting, the matching private key can be used to produce a role signature. A successful verification of a role signature using the associated role information then authenticates the user who generated the signature [17].

We observe that the concept of role signatures can be directly applicable to a grid environment by exploiting the hierarchical structure of a VO. To integrate the concept with our PECF-GSI setting, we require that the TA is trusted to make available a set of generic roles and to issue credentials to all legitimate member organisations of a VO. These member organisations, in turn, are trusted to assign their respective users to generic roles defined by the TA and also trusted to issue the relevant credentials to the users. Subsequently, when a user signs a job request (thus creating a role signature) using a key associated with some generic roles assigned to the user, these generic roles can then be mapped to local roles using the access control policies defined by resource owners.

More specifically, once user A has performed single sign-on (as previously described in Sect. 2.4.2), A can invoke a job using the GT's GRAM component, for example, and sign a job request (describing the job to be run) using a HIBS scheme. Suppose the job request is targeted at resource X . Upon receiving the signed job request and the relevant information from A , the policy decision point (PDP) for resource X performs the following steps:

1. the PDP verifies the signed request;
2. if the signature can be successfully verified, the PDP extracts the appropriate role information from the identifier;
3. the PDP grants the request if the role is authorised for the request.

Since X trusts A 's organisation to issue the correct credential to A , X can be convinced of A being an authorised user with the appropriate role, if the signed request is

⁵ We assume that AS has obtained an authentic parameter set from the TA.

valid. By adopting role signatures here, inter-domain principal mapping is made trivial and avoids the need for pre-establishment of user accounts at remote resources. Moreover, user authentication and authorisation are unified and become simpler, in the sense that the PDP needs to verify only a single role signature when making an access control decision.

2.4.4 User-resource mutual authentication and key agreement

While a password-based TLS protocol is a convenient mechanism for an authentication server and its (known) users, the standard PKI-based TLS protocol is clearly more suitable for mutual authentication and key agreement between two entities who have not previously communicated.

As in many GSI-based grid systems, once user A has been granted the rights to access resource X in PECF-GSI, these two entities (through their proxies) authenticate each other and establish a fresh session key for subsequent secure data transmission. In PECF-GSI, this is achieved using a certificate-free (TLS-like) key agreement protocol, as described in Sect. 3.3.

At this stage, X is ready to execute A 's tasks based on the description specified in A 's job request. A may also delegate her credential to X , so that X can act on her behalf in accessing other additional resources when necessary.

2.4.5 Delegation

The current delegation technique used in the GSI requires a round-trip interaction between a delegator (typically a grid user) and a delegatee (typically a hosting server or resource provider). Also, verification of a delegatee's credential requires validating both long-term and proxy certificates of all the parties involved along the delegation chain. Further details on the GSI delegation technique can be found in Sect. 6.1.

Lim and Paterson proposed an identity-based one-pass delegation technique [39], in which the delegator signs a delegation token and forwards it to the delegatee. One advantage of this approach is that the delegator can bind the delegatee's public key information to the delegation token without acquiring the delegatee's proxy public key, thus requiring only one message in the delegation technique. However, verification of a delegatee's status as the delegation target requires validation of all the signed delegation tokens (analogous to validation of certificates in the GSI) issued by all the delegators along the delegation chain.

We now introduce a new delegation technique that is not only certificate-free and one-pass, but also has a very efficient verification mechanism, in the sense that a delegatee's delegated credential can be validated by performing only one

signature verification, regardless of the length of the delegation chain. This is a significant improvement on the two aforementioned delegation methods.

We now explain the details of such a delegation technique that we employ in PECF-GSI using an example between the delegator A (through her proxy \bar{A}) and the delegatee \bar{X} . In the delegation process, \bar{A} performs the following steps:

1. compute a proxy public key $P_{\bar{A}/\bar{X}}$ based on an identifier of the form

$$\langle \text{ID}_{AS}, \text{ID}_A \parallel \text{LT}_A, \text{ID}_X \parallel \text{LT}_{\bar{X}} \parallel \text{Job}_{\bar{X}} \parallel \text{Policy}_{\bar{X}} \rangle,$$

where $\text{LT}_{\bar{X}}$ is the lifetime that \bar{A} decides for \bar{X} , $\text{Job}_{\bar{X}}$ describes A 's job request, and $\text{Policy}_{\bar{X}}$ indicates the policy that A wishes to enforce on \bar{X} ;⁶

2. extract a proxy private key $S_{\bar{A}/\bar{X}}$ using her secret value $s_{\bar{A}}$;
3. transmit

$$\langle \text{ID}_X \parallel \text{LT}_{\bar{X}} \parallel \text{Job}_{\bar{X}} \parallel \text{Policy}_{\bar{X}}, S_{\bar{A}/\bar{X}} \rangle$$

to \bar{X} through the secure channel⁷ that was previously established between \bar{A} and \bar{X} as described in Sect. 2.4.4.

In this case, \bar{A} actually acts as a PKG and issues a private key to \bar{X} , which becomes a child of \bar{A} at level 3 in Fig. 2. This can be seen in the identifier used to construct $P_{\bar{A}/\bar{X}}$, in which the first two parts are identifiers of \bar{X} 's ancestors, i.e. ID_{AS} and $\text{ID}_A \parallel \text{LT}_A$ (these identifiers are distributed to \bar{X} when it performs the TLS handshake with \bar{A}).

If a third party, for example \bar{Y} , wants to verify that \bar{X} is indeed acting on \bar{A} 's behalf, then \bar{Y} must (i) authenticate \bar{X} and (ii) check that \bar{X} is in possession of $S_{\bar{A}/\bar{X}}$. These two checks will be carried out as part of the TLS handshake that takes place between \bar{X} and \bar{Y} .

When \bar{X} further delegates \bar{A} 's credential to another hosting server Y , \bar{X} can construct a new proxy public key $P_{\bar{A}/\bar{Y}}$ based on identifier

$$\langle \text{ID}_{AS}, \text{ID}_A \parallel \text{LT}_A, \text{ID}_X \parallel \text{LT}_{\bar{X}} \parallel \text{Job}_{\bar{X}} \parallel \text{Policy}_{\bar{X}}, \\ \text{ID}_Y \parallel \text{LT}_{\bar{Y}} \parallel \text{Job}_{\bar{Y}} \parallel \text{Policy}_{\bar{Y}} \rangle,$$

⁶ Here we assume that a verifier would check the lifetime and enforce the policy specified by \bar{A} . These are roughly equivalent to verifying the validity period and enforcing policy stated in a proxy certificate in the GSI.

⁷ It might be thought that the need for the secure channel to transport the proxy private key from \bar{A} to \bar{X} is a limitation of this approach. In fact, the secure channel between these two parties will exist anyway; the parties have to authenticate each other using an authenticated key agreement protocol, before the delegation can take place. This is to ensure that the delegation is targeted at the right entity and that the delegation target is convinced of the identity of the delegator.

where $\text{Job}_{\bar{Y}}$ refers to the job (potentially subtasks of $\text{Job}_{\bar{X}}$) that \bar{X} wants \bar{Y} to execute and $\text{Policy}_{\bar{Y}}$ refers to the policy that \bar{X} imposes on \bar{Y} , respectively. The matching private key is $S_{\bar{A}/\bar{X}/\bar{Y}}$. This private key and the relevant information can then be forwarded to \bar{Y} , which subsequently becomes subordinate to \bar{X} at level 4 of the hierarchy.

To verify \bar{Y} 's delegated proxy credential, the verifier only needs to authenticate \bar{Y} and check whether \bar{Y} knows the private key corresponding to $P_{\bar{A}/\bar{X}/\bar{Y}}$, even though the delegation chain now has two delegates (\bar{X} and \bar{Y}). This can be done, in principle, by verifying a signature produced by \bar{Y} using $S_{\bar{A}/\bar{X}/\bar{Y}}$.

3 Realising PECF-GSI

In this section, we discuss how to instantiate PECF-GSI using existing cryptographic schemes and protocols. We also discuss the key management aspects of PECF-GSI.

3.1 The Gentry–Silverberg HIBE and HIBS schemes

Soon after Boneh and Franklin proposed the first practical and secure identity-based encryption [13], Gentry and Silverberg proposed hierarchical identity-based encryption (HIBE) and hierarchical identity-based signature (HIBS) schemes with total collusion resistance, regardless of the number of levels in the hierarchy [28]. It is worth noting that other HIBE and HIBS schemes are available, for example [12]. We chose the Gentry–Silverberg schemes because they are efficient, their security is based on reasonable computational assumptions, and they support the dynamic addition of new levels in the hierarchy of identifiers.

Before we look at the Gentry–Silverberg schemes, we briefly explain the concept of *pairings*.

Let \mathbb{G}_1 and \mathbb{G}_2 be two groups of order q for some large prime q , where \mathbb{G}_1 is an additive group and \mathbb{G}_2 denotes a related multiplicative group. A pairing in the context of identity-based cryptography is a function $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Typically, \mathbb{G}_1 is a subgroup of the group of points on a suitable elliptic curve over a finite field, \mathbb{G}_2 is obtained from a related finite field, and \hat{e} is obtained from the Weil or Tate pairing on the curve. Further details on pairings and their implementations using elliptic curves can be found in [27].

We now sketch Gentry and Silverberg's HIBE and HIBS schemes (see [28] for full details):

- **ROOT SETUP:** The root TA chooses a generator $P_0 \in \mathbb{G}_1$, picks a random $s_0 \in \mathbb{Z}_q^*$, and sets $Q_0 = s_0 P_0$. It also selects five cryptographic hash functions H_1, H_2, H_3, H_4, H_5 . The root TA's master secret is s_0 and the system parameters are

$$\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P_0, Q_0, H_1, H_2, H_3, H_4, H_5 \rangle.$$

- **LOWER-LEVEL SETUP:** A lower-level entity (lower-level PKG or user) at level t picks a random $s_t \in \mathbb{Z}_q^*$ which will be kept secret.
- **EXTRACT:** For an entity at level t with identifier $\langle \text{ID}_1, \dots, \text{ID}_t \rangle$, where $\langle \text{ID}_1, \dots, \text{ID}_i \rangle$ is the identifier of the entity's ancestor at level i ($1 \leq i \leq t - 1$), the entity's parent computes $P_i = H_1(\text{ID}_1, \dots, \text{ID}_i) \in \mathbb{G}_1$, sets the secret point S_i to be

$$\sum_{i=1}^t s_{i-1} P_i = S_{t-1} + s_{t-1} P_t$$

(note that S_{t-1} is the parent's secret point given by the parent's ancestor and s_{t-1} is a secret value only known to the parent), and defines Q-values by setting $Q_i = s_i P_0$ for $1 \leq i \leq t - 1$. The entity at level t is given both S_t , as his private key, and the Q-values by its parent.

- **ENCRYPT:** Given system parameters, a message m , and identifier $\langle \text{ID}_1, \dots, \text{ID}_t \rangle$, this algorithm computes the ciphertext $\langle U_0, U_2, \dots, U_t, V, W \rangle$. Note that U_0, U_2, \dots, U_t are all elements of \mathbb{G}_1 and the sizes of the last two components of the ciphertext are dependent on n , the bit length of plaintexts.
- **DECRYPT:** Given a ciphertext $\langle U_0, U_2, \dots, U_t, V, W \rangle$, this algorithm takes as input the associated private key S_t and recovers m . It also checks if U_0, U_2, \dots, U_t have the correct structure. Otherwise, the recovered m is rejected.
- **SIGN:** Given the private key S_t for a signer with identifier $\langle \text{ID}_1, \dots, \text{ID}_t \rangle$, the algorithm outputs a signature of the form $\langle \sigma, Q_1, \dots, Q_t \rangle$, where each component is in \mathbb{G}_1 .
- **VERIFY:** Given a signature $\langle \sigma, Q_1, \dots, Q_t \rangle$ of a message m , this algorithm takes as input the associated identifier $\langle \text{ID}_1, \dots, \text{ID}_t \rangle$, and returns a value indicating the success or failure of the verification.

As discussed before, the Gentry–Silverberg HIBE and HIBS schemes can be used for performing mutual authentication, key agreement, access control, and delegation.

3.2 The Abdalla et al. password-based TLS protocol

Abdalla et al. [1] proposed a provably secure password-based TLS protocol, based on earlier work of Steiner et al. [54]. The protocol makes use of a discrete logarithm-based mask generation function to instantiate a symmetric encryption primitive, as suggested by Bellare et al. [8, 10]. The mask generation function is based on a group \mathbb{G} , with generators g_1 and g_2 , and a hash function H whose outputs are integers modulo the size of group \mathbb{G} . Then, A with password PW_A encrypts a Diffie–Hellman component g_2^a by

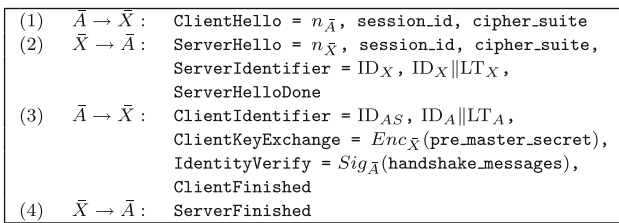


Fig. 3 A certificate-free authenticated key agreement protocol

calculating $g_2^a \cdot \pi_A$, where $\pi_A = g_1^{H(PW_A)}$. Thus, the result of the encryption is a group element. To decrypt and recover g_2^a , one can simply divide the ciphertext by π_A .

The important point about this protocol is that dictionary attacks are of little value to an adversary. If the adversary guesses a password and uses it to decrypt $g_2^a \cdot \pi_A$, he simply obtains a random group element. See [1] for full details of the protocol.

We use the password-based TLS protocol of Abdalla et al. to support single sign-on in PECF-GSI. We make one minor modification to the protocol as presented in [1] in order to ease its deployment in our architecture: we simply translate the scheme of [1] to the elliptic curve setting, re-using the group G_1 in place of G . We stress that this change does not undermine any of the strong security guarantees that were proven for this protocol in [1], provided that G_1 is chosen so that the Computational Diffie–Hellman problem in G_1 is sufficiently hard (as it must be in order to guarantee the security of the Gentry–Silverberg HIBE and HIBS schemes in any case).

It is worth noting that Steiner et al. and Abdalla et al. suggest that parameters such as G, g_1, g_2 , and H should be fixed or form part of a standardised ciphersuite. This obviates the need for the user A to verify the number-theoretic appropriateness of these parameters.

3.3 Certificate-free TLS protocol

We explained before that in our proposal of PECF-GSI, a TLS-like protocol is required for mutual authentication and key agreement between a user proxy and a resource.

Figure 3 shows a certificate-free authenticated key agreement protocol, which is obtained from the standard TLS handshake protocol (based on RSA encryption for secure transport of the pre-master secret) by replacing the conventional encryption and signature primitives by HIBE and HIBS, respectively, and by replacing public key certificate chains by the appropriate identifiers.

In the first message of the protocol, $n_{\bar{A}}$ denotes a nonce chosen by \bar{A} , `session_id` is a string used to distinguish concurrent sessions, and `cipher_suite` contains a cipher specification that handles HIBE and HIBS schemes, for example

TLS_HIBE_HIBS_WITH_DES_CBC_SHA.

Here, $Enc_{\bar{X}}(\cdot)$ denotes an encryption using a HIBE scheme with X 's proxy public key $P_{\bar{X}}$ (derived from identifier $\langle ID_X, ID_X, ||LT_X \rangle$), while $Sig_{\bar{A}}(\cdot)$ represents a signing operation in a HIBS scheme using A 's proxy private key $S_{\bar{A}}$.

When \bar{A} (playing the role of client) performs mutual authentication with \bar{X} (playing the role of server), she needs to forward her public key information, i.e. $\langle ID_{AS}, ID_A || LT_A \rangle$ to \bar{X} as part of the protocol handshake, and vice versa. Note that since \bar{X} includes its long-term identifier in the `ServerIdentifier` message, \bar{A} must check whether the identifier is still valid using the IRL that she received from AS . Similarly, since \bar{A} has a key that sits under that of AS in the hierarchy, \bar{X} must check that AS 's identifier has not been revoked before interacting with \bar{A} .

In step (3), `pre_master_secret` denotes a secret value chosen by \bar{A} that will be used to derive session keys for the subsequent TLS record layer protocol. On the other hand, `handshake_messages` refers to all handshake messages sent or received starting at `ClientHello` up to but not including this message. The `ClientFinished` and `ServerFinished` messages are constructed just as they are in standard TLS, using the key `master_secret` derived from `pre_master_secret` to compute a MAC on all the messages exchanged in the protocol. This provides explicit authentication of \bar{X} to \bar{A} and protection against version roll-back and ciphersuite downgrade attacks [19], as in standard TLS.

We note that our certificate-free authenticated key agreement protocol can be adapted straightforwardly to support user-to-user authentication (performed at the level of user proxies using their short-term keys) within a grid environment.

3.4 System parameters and keys

We now describe the system parameters and keys that will be used in PECF-GSI, assuming the underlying cryptographic schemes are the Gentry–Silverberg HIBE and HIBS schemes.

3.4.1 Parameter generation and distribution

During the system set-up phase, the TA runs a Bilinear Diffie–Hellman (BDH) parameter generator to obtain groups G_1, G_2 of large prime order q and an admissible pairing $\hat{e} : G_1 \times G_1 \rightarrow G_2$. It then performs the ROOT SETUP of the Gentry–Silverberg HIBE and HIBS schemes to produce a master secret s_0 . The system parameters are $\langle G_1, G_2, \hat{e}, P_0, Q_0, H_1, H_2, H_3, H_4, H_5 \rangle$. We will discuss concrete parameter choices in Sect. 5.2.

Table 1 Credentials and keys in PECF-GSI

Scheme	Entity	Long-term credential		Proxy credential	
		Public key	Private key	Public key	Private key
Gentry–Silverberg (HIBE/HIBS)	AS	$P_{AS} = H_1(\text{ID}_{AS})$	$S_{AS} = s_0 P_{AS}$	–	–
	A	–	–	$P_{\bar{A}} = H_1(\text{ID}_{AS}, \text{ID}_A \parallel \text{LT}_A)$	$S_{\bar{A}} = S_{AS} + s_{AS} P_{\bar{A}}$
	X	$P_X = H_1(\text{ID}_X)$	$S_X = s_0 P_X$	$P_{\bar{X}} = H_1(\text{ID}_X, \text{ID}_X \parallel \text{LT}_X)$	$S_{\bar{X}} = S_X + s_X P_{\bar{X}}$

An authentic set of the TA parameters must be made available to each authentication and hosting server. One way to achieve this is by bootstrapping these parameters into the grid middleware that needs to be installed on the servers (Note that this is roughly comparable to bootstrapping conventional CA certificates into web browsers.)

3.4.2 Key generation

Once the system parameters have been set up, the TA can issue private keys to its subordinates at level 1 (see Fig. 2) using its master secret s_0 . For example, authentication server AS 's long-term private key is $S_{AS} = s_0 P_{AS}$, where $P_{AS} = H_1(\text{ID}_{AS})$ is the matching public key. Hosting servers' long-term public/private keys are generated in a similar way. A proxy's public key at level 2 can be computed based on its ancestor's identifier and its own identifier concatenated with a lifetime LT in some fixed format. For example, user A 's proxy public key would be $P_{\bar{A}} = H_1(\text{ID}_{AS}, \text{ID}_A \parallel \text{LT}_A)$, and the corresponding private key can be obtained from AS , who will run the EXTRACT algorithm to generate $S_{\bar{A}} = S_{AS} + s_{AS} P_{\bar{A}}$. Here, s_{AS} is a secret value chosen by AS . The upper part of Table 1 summarises the credentials possessed by the authentication server AS , user A , and hosting server X .

We recall that A does not possess any long-term credential, except a password that she shares with AS . In fact, in our architecture, the user proxies have identifiers placing them below AS in the hierarchy, meaning that these proxies are effectively subordinates of AS (as shown in Fig. 2).

3.5 Key revocation

We remark that the commonly used key revocation approach through Certificate Revocation Lists (CRLs) in the existing PKI-based GSI also has its limitations. Retrieving an up-to-date CRL is an ongoing maintenance task that a user must carry out and it is not uncommon to find systems with out-of-date CRLs [7]. Furthermore, at least in the case of the UK e-Science community, it is not clear how a user's public key certificate can be revoked after the user has left his organisation. Unless the user informs the CA of his departure, he can continue using the certificate until its expiry.

Our proposed design deals with revocation of user keys and of hosting server keys in different ways. The users are never given long-term keys; instead, they are only ever provided with proxy keys. As with proxy certificates [56] and Kerberos tickets [44], proxy keys have a short lifetime, typically less than 12 h. As the window of exposure to compromise is minimised, there is no need for an explicit revocation mechanism for user keys. Instead, the hosting servers trust AS to generate a short lifetime proxy public/private key pair for a user only if she is authorised.

Conversely, hosting servers are issued with long-term keys, for which we need an explicit revocation mechanism. To allow for the revocation of servers' public keys, we make use of an Identity Revocation List (IRL) issued by the TA. The IRL includes the identifier of any server whose key has been revoked.⁸ This allows users' proxies to verify the validity of a particular hosting server's public key prior to submitting a job to that server. IRLs are distributed to users by AS via the secure channel established at the time of authentication. From the user's perspective, this "push" method of distribution simplifies the process of verifying whether a hosting server has had its public key revoked.

4 Security analysis

In this section, we provide an informal and high-level security analysis of PECF-GSI against potential threats described in Sect. 2.1.

4.1 Authentication

Abdalla et al. have proved in [1] that their password-based TLS protocol is secure in the security model introduced in [2]. When translating their protocol to the elliptic curve setting, we need only ensure that the parameters chosen provide an appropriate level of security. However, as noted previously, this is achieved through an appropriate choice of the group \mathbb{G}_1 .

⁸ We envisage that an identifier of a hosting server would contain its permanent subject name, to which an issue number is appended. Analogous to the use of issue numbers in credit cards, a server will be assigned an incremental issue number every time its private key is compromised.

In our single sign-on approach, we assume that AS is a party trusted to issue the correct system parameters, most importantly Q_0 , and up-to-date IRLs to its users through secure channels. Therefore, no additional infrastructure is required to verify the authenticity of the parameters and IRLs. Note that most of the components of the system parameters discussed in Sect. 3.4.1 can be fixed and made public, except $Q_0 = s_0 P_0$, where s_0 is the TA's master secret. Failure to obtain Q_0 from a trusted source would allow a trivial man-in-the-middle attack. Our single sign-on technique is secure against such an attack, assuming AS behaves honestly. Also, we assume that hosting servers always trust AS in issuing proxy credentials to the correct users. These assumptions are essential for the protocol in Fig. 3 and our delegation technique to work as intended.

4.2 Secure communication

The security of the certificate-based TLS (or SSL) protocol has been well studied, see for example [37,48,57]. In our certificate-free key agreement protocol (shown in Fig. 3), we replace public key certificates required in the TLS protocol with identifiers and the standard public key algorithms with hierarchical identity-based equivalents. Here, we discuss to what extent the changes impact on the security of the TLS protocol.

In step (2) of our protocol, we have removed the `Certificate` message that the server must send to the client in the original TLS specification. It has been replaced by `ServerIdentifier` that contains $(ID_X, ID_X || LT_X)$. Moreover, the `Certificate` message in step (3) which is supposed to contain the client's certificates has become `ClientIdentifier` = $(ID_{AS}, ID_A || LT_A)$. Neither \bar{A} nor \bar{X} have to verify the validity of the respective identifier that they each received. Since we assume that the TA has carried out its responsibility appropriately, \bar{A} and \bar{X} can each trust that the exchanged identifier came from the genuine party if this party can prove its possession of the associated private component. Thus, if \bar{X} can successfully verify the signed handshake messages from \bar{A} using a public key constructed from \bar{A} 's identifier, then \bar{A} is authenticated to \bar{X} . On the other hand, for \bar{A} to authenticate \bar{X} , she must receive the correct verification value in `ServerFinished` from \bar{X} , based on the pre-master secret that she has chosen. This confirms that \bar{X} has recovered the correct pre-master secret that \bar{A} sent encrypted under \bar{X} 's identifier. Should the adversary attempt to impersonate either the user or the server, he must break the Gentry–Silverberg HIBE/HIBS schemes (or recover a private key belonging to \bar{A} or \bar{X}). These schemes were proved secure in the Random Oracle model [9] in an appropriate model in [28].

We can safely conclude that the replacement of certificates with identifiers and standard public key algorithms with

hierarchical identity-based equivalents does not weaken the security protection that the original TLS protocol offers.

4.3 Delegation

One important security requirement of our delegation technique is the need to establish an authenticated, confidential, and integrity-protected channel between a delegator and a delegatee. The failure to do so could easily allow a simple man-in-the-middle-attack or leakage of information about a delegated proxy private key. Our certificate-free authenticated key agreement protocol enables a secure channel with the desired security properties to be established, assuming the underlying Gentry–Silverberg HIBE/HIBS schemes are secure.

We note that repeated exposure of proxy private keys and delegated keys does not by itself lead to the exposure of any higher-level keys (for example, the AS private key or server private keys). This is a property guaranteed by the collusion resistance of the Gentry–Silverberg HIBE/HIBS schemes. Moreover, the exposure of any number of (delegated) proxy private keys does not help an attacker to learn any further private keys, except those for identities below the affected proxies in the hierarchy. Again, this follows from the collusion resistance of the selected HIBE/HIBS schemes. Thus, an inside attacker E , in possession of a proxy private key for identifier $(ID_{AS}, ID_E || LT_E)$, cannot use this key to help it construct private keys for other identifiers of the form $(ID_{AS}, ID_A || LT_A)$ or any of their children.

It is worth pointing out that during delegation, a hosting server may use its private key to both decrypt (HIBE) and sign (HIBS) while establishing secure channels with the relevant parties. While there are no known security issues with such dual use of private keys in the Gentry–Silverberg schemes, it would be advisable to employ separate private keys for decryption and signing functions. This can be done by simply encoding identifiers with the pre-fixes 'Dec' and 'Sig' to produce encryption and signature identifiers and their respective private keys.

5 Discussion

In this section, we compare PECF-GSI with existing architectures in terms of features and performance. We also discuss interoperability issues between our proposal and the existing GSI approach.

5.1 Feature comparison

To provide a more clear view of the differences between our proposal of PECF-GSI and existing approaches in the literature, we summarise their architectural features in Table 2.

Table 2 A comparison of features

Feature	GSI _{PKI}	GSI _{MP}	GSI _{Ker}	IKIG	PECF-GSI
PKI-free (from user perspective)			✓		✓
Certificate-free				✓	✓
User key revocation not required			✓	✓	✓
Password-enabled		✓	✓		✓
Protection of long-term keys not required			✓		✓

We use GSI_{PKI} to denote the standard PKI-based GSI, whereas GSI_{MP} represents the PKI-based GSI with MyProxy plug-in. The Kerberised-GSI, denoted by GSI_{Ker}, as described in Sect. 6.1, allows existing Kerberos users to access grid resources managed by the PKI-based GSI. Lim and Paterson's earlier proposal of an identity-based key infrastructure for grid applications [39] is denoted by IKIG.

Since GSI_{Ker} and PECF-GSI do not make use of long-term user private keys, protection and revocation mechanisms for these keys are not required. Thus, these architectures are generally simpler in terms of credential management and thus more user-friendly compared with the others. While the degrees of user-friendliness for GSI_{Ker} and PECF-GSI are roughly similar, our certificate-free approaches may well have lower communication costs.

To summarise, our PECF-GSI approach is easy to use and especially suitable for grid systems with bandwidth constraints.

5.2 Performance

We now compare the communication costs of the protocols used in GSI and PECF-GSI for key agreement and delegation. Subsequently, we compare the computational costs of long-term and proxy key generation, key agreement, and delegation. Note that here the performance characteristics that we use in our comparison should yield similar results for GSI_{PKI}, GSI_{MP}, and GSI_{Ker}. Hence, we use GSI to represent GSI_{PKI}, GSI_{MP}, and GSI_{Ker}. On the other hand, PECF-GSI represents an architectural improvement on IKIG, while the underlying protocols in PECF-GSI are similar to those in IKIG except for a more efficient delegation protocol.

In the GSI setting, each user has a long-term RSA public/private key pair with a 1,024-bit modulus. The short-term keys for the user's proxy credential have only 512-bit moduli. This substantial reduction in the size of short-term keys is driven by the fact that generating an RSA key pair is a computationally expensive operation. It has been shown that generating a key pair with 512-bit moduli can reduce the processing time by approximately 77% of the time required for a 1,024-bit key pair [58]. Since the proxy credential has a relatively short lifetime, it is currently believed that the reduction in security implied by using only 512-bit moduli

poses an acceptably low risk in grid systems. However, it is arguable that this position is not sustainable in the long term, given recent progress in factoring [35]. We assume the size of a 1,024-bit RSA public key certificate is 1.5 kilobytes (ignoring small fields, such as subject and validity period). Similarly, a 512-bit RSA proxy certificate is 0.8 kilobytes.⁹ Ciphertexts and signatures generated using a short-term RSA key are 512 bits.

For PECF-GSI, we work with a supersingular elliptic curve of embedding degree 4 over \mathbb{F}_{271} [25, 26] to obtain the system parameters described in Sect. 3.4.1.¹⁰ This choice results in a corresponding group of prime order q approximately equal to 2^{252} and gives roughly the same security level as 1024-bit RSA. Using point compression, elements of this group can be represented using 272 bits. Since all arithmetic is carried out in fields of characteristic 2, group operations and pairing computations can be implemented efficiently [4, 27].

In addition to the curve and group selections, we require hash functions for the Gentry–Silverberg HIBE and HIBS schemes. The outputs of H_1 and H_3 are elements of \mathbb{G}_1 , while H_4 gives an output with approximately 252 bits. Note that the size of outputs of H_2 and H_5 are dependent on n , the bit length of plaintexts. We assume that $n = 256$, since this is sufficient for our protocol messages (see Sect. 3.3). Hence, the size of ciphertexts and signatures produced by the Gentry–Silverberg HIBE and HIBS schemes can be computed and are 1056 bits and 816 bits, respectively.

Table 3 summarises the estimated communication costs for the protocols that underpin the GSI and our proposal of PECF-GSI. Actual computational costs in milliseconds are also summarised in this table. These timings were obtained by implementing the key generation algorithms in RSA and the Gentry–Silverberg HIBE/HIBS schemes using C++ and

⁹ It is worth mentioning that RSA keys can be replaced by much shorter keys, which are based on elliptic curve cryptography (ECC) [11]. However, this does not eliminate the fact that certificates will still be used and the associated limitations of certificated-based architectures will be incurred.

¹⁰ We note that this curve is only chosen so that concrete timings and bit counts can be given. A wide variety of other curves and associated parameters are available. Some of these will require working in the more general setting of asymmetric pairings; see [27] for more details.

Table 3 A comparison of performance characteristics

Type of cost (Units)	Operation	GSI	PECF-GSI
Communication (KB)	Key agreement	37.8	1.9
	Delegation	7.8	0.3
Computation time (ms)	Long-term key generation	143.19	2.04
	Proxy key generation	32.36	2.17
	Key agreement	5.18	20.87
	Delegation	34.95	7.25

the MIRACL library [53]. The experiments were performed on a Pentium IV 3.2-GHz processor with 1-GB memory. For simplicity, we limit the length of the delegation chain to one.

5.2.1 Communication costs

In the GSI, the communication cost of the key agreement protocol through the standard TLS handshake is approximately 37.8 kilobits; the corresponding cost in PECF-GSI is approximately 1.9 kilobits. For simplicity, we ignore small components in both the TLS protocols, such as the `ClientHello` and `ClientFinished` messages.

The communication costs for delegation in the GSI can be estimated straightforwardly from the protocol described in Sect. 6.1. Delegation in PECF-GSI is very lightweight because it only involves issuance of a private key. It is evident that our certificate-free approach suits wireless environments well, in which transmission of data using battery-powered mobile devices is a relatively expensive operation.

5.2.2 Computational costs

We can see from Table 3 that key generation in the GSI is significantly more costly than in PECF-GSI.¹¹ This suggests that the single sign-on protocol in our proposal is computationally cheaper than in the GSI incorporating MyProxy.

The figures for key agreement (including mutual authentication) are obtained by summing the times taken for the user and authentication server to perform their respective parts of the protocol. A similar method is used to obtain a single figure for the computational costs of delegation. Key agreement in the GSI (using the standard TLS protocol) is computationally less expensive than the corresponding operations in PECF-GSI (using the modified identity-based TLS protocol). In contrast, delegation in PECF-GSI is faster than in the GSI.

¹¹ Note that we only compare private key extraction in PECF-GSI to RSA public/private key pair generation in the GSI, because the time taken to compute a public key using the hash function H_1 in PECF-GSI is negligible given the parameters we have chosen. Furthermore, construction of a public key by hashing an identifier occurs as part of the associated encryption/decryption scheme.

It is unfortunate that key agreement is slower in PECF-GSI, but we note that the cumulative time for key agreement and delegation is still lower in PECF-GSI. Overall, it can be seen that the computational costs of PECF-GSI are comparable with those of the GSI. Our approach allows a different trade-off to be struck between the computational costs at the user side and at the server side. Particularly, operations involved in both the key agreement and delegation protocols at the user side are more lightweight than the server, when compared with those of the GSI. This is because in the PECF-GSI the more computationally expensive pairing evaluations are performed at the high-performance server. In the GSI, however, the dominant computational cost from RSA key generation is incurred at the user side.

5.3 Interoperability

Considering that certificate-based PKI has already been widely used in the grid community, it is essential that our architecture interoperates with existing solutions. We now show that this can be achieved without making substantial changes to PECF-GSI.

Let us consider PECF-GSI running in VO_1 and PKI-based GSI running in VO_2 as an example. When a user from VO_2 who possesses a valid X.509 public key certificate wishes to access resources not available locally but located at a member organisation within VO_1 , the user can present his certificate together with a signed request (using the associated private key) to the authentication server AS_1 for VO_1 . AS_1 , in principle, can authenticate the user and issue him a set of identity-based credentials, system parameters and other necessary information as it would to its own users, under the following assumptions:

- The user has the necessary identity-based cryptographic schemes installed and running on his machine;
- AS_1 trusts the CA who issued the user's certificate and is in possession of the CA's certificate;
- AS_1 can extract the username and the user's role information from the certificate itself, or from the certificate and a separate attribute certificate;

- The role information is based on a set of generic roles shared by AS_1 's TA and the user's CA.¹²

Note that the above scenario considers a “certificatised” variant of PECF-GSI, analogous to a Kerberised-GSI, which enables interoperability between a Kerberos client and a PKI-based resource provider.

On the other hand, it seems more difficult for an entity from VO_1 to securely communicate with an entity from VO_2 using certificate-based protocols. For example, if hosting server W within VO_1 wants to access additional resources residing in hosting server Z within VO_2 to complete a job, W must be able to perform mutual authentication with Z using the standard certificate-based TLS protocol. That would require W to be in possession of two sets of credentials: identity-based and certificate-based key pairs. Obviously, it is easier for AS_1 , as a domain authentication server, to obtain and store trustworthy CAs' certificates, than requiring each resource provider to register with a CA and obtain a certificate, even though users of VO_1 and VO_2 are not affected in either case.

A longer and more general discussion of interoperability issues for identity-based and certificate-based architectures can be found in [50].

6 Related work

6.1 The grid security infrastructure

The PKI-based GSI focuses on authentication, message protection, and the use of proxy credentials to support single sign-on and credential delegation [23,58,59]. In grid applications that employ the GSI, each entity, who can be a member of one or more virtual organisations, is assigned a unique identity or distinguished name and given a public key certificate signed by a Grid CA. Public key certificates are used to support authentication and key agreement protocols, such as TLS. Proxy certificates are used for single sign-on and credential delegation.

Before a user submits a job request, he must create a proxy certificate that includes generating a new public/private key pair and signing the proxy certificate with his long-term private key. This newly created proxy certificate can then be used for repeated authentication with other grid entities. The user's long-term private key does not need to be accessed again until the expiry of the proxy certificate. For rights delegation from a user A to a target service provider X , three steps are required [58]:

1. X generates a new public/private key pair and sends a request (that is signed with the new private key) to A ;
2. A verifies the request using the new public key, creates a new proxy certificate, and signs it with her current proxy credential (short-lived private key);
3. A forwards the new proxy certificate to X .

There are a small number of grid projects that use Kerberos [44] as the backbone of their security infrastructures. It is generally believed that Kerberos, being based on symmetric key cryptography, is more efficient than PKI-based approaches. However, Kerberos is unlikely to be a suitable long-term solution because many computational grids have a dynamic user population, and the establishment and management of shared symmetric keys will be impractical. Furthermore, it is not clear how the dynamic delegation mechanism of [58] can be supported using Kerberos. Therefore, PKI is preferred for grid applications, while Kerberos seems to be best suited for intra-domain security. In order to achieve inter-operability with PKI-based systems, some Kerberos-based grid projects make use of a Kerberised client-side programme, called KX.509, to acquire X.509 certificates using a client's existing Kerberos ticket [36,43].

6.2 MyProxy

Inevitably, some machines within the scope of a grid community may lack up-to-date protection in the form of the latest vulnerability patches and virus definitions. This may lead to such machines falling under the partial or complete control of attackers who are able to remotely exploit vulnerabilities and hence obtain long-term user credentials. To minimise the risk of compromise, many recent grid implementations make use of the MyProxy system [5,45] to securely store and protect long-term user credentials. MyProxy, an implementation of the virtual smart card concept [51], also offers the benefit of “credential mobility”, enabling users to access their credentials from any machine through, for example, a web browser.

Our proposal is architecturally similar to MyProxy. However, MyProxy relies on a certificate-based PKI. In the MyProxy protocol [5], although users are authenticated to their respective MyProxy servers using conventional username/password techniques, server authentication is achieved using the server-authenticated version of the TLS handshake protocol [19]. This implies the need to protect the root CA's public key certificate on the users' machines and for users to know how to interpret advice concerning certificate checks. There are ways for an attacker to install a bogus root key in the user's browser [3,31]. Hence, if a desktop is vulnerable to private key exposure, then the desktop may also be at risk from replacement of the associated CA's certificate by the attacker. In short, the user must ensure that the associated certificates

¹² The definition and sharing of generic roles are possible through an independent body called a Policy Management Authority (PMA), such as the European Grid PMA.

bootstrapped in his machine are trustworthy and have not been replaced. On the other hand, our proposal of PECF-GSI requires a user to remember only a password which she shares with her local authentication server.

6.3 Improving usability

Beckles et al. [7] considered issues related to the usability of the PKI-based GSI, noting that managing certificates can be onerous for general grid users. In an effort to improve the usability of the PKI-based GSI, they adopted Gutmann's plug-and-play PKI (PnP PKI) concept [30], which emphasises automated and transparent set-up of PKI for the end user. In so doing, Beckles et al. make use of the PKIBoot service of PnP PKI to allow a user to authenticate himself to a PKIBoot server with the standard username/password method. Subsequently, the user can securely retrieve his public key certificate (and optionally his private key) and/or CAs' certificates. This approach can eliminate the difficult tasks involved in correctly establishing trust roots of CAs from the user side. It can also minimise the user's direct involvement in certificate management. Our proposal for a user-friendly and certificate-free security architecture is influenced by Beckles et al.'s work, in the sense that user key management is minimal.

Although the plug-and-play PKI concept seems to make PKI more usable for users, there are still many aspects of PKI that need to be addressed. For example, how can we improve the effectiveness of current key revocation mechanisms, such as Certificate Revocation Lists (CRLs), by exploiting the advantages that the plug-and-play PKI concept could bring? Furthermore, the application of the plug-and-play PKI to the GSI does not reduce the extensive use of certificates, and certificate chain verification is still required for all grid security services that require certificates. These issues are addressed in our proposal.

7 Conclusions and future work

We have proposed a grid security infrastructure which is password-enabled and certificate-free. Our infrastructure offers the following distinct advantages.

- The only long-term secret required by users is a password. This is likely to improve usability and accessibility of grid applications considerably. Moreover, revocation of users' public keys, a considerable problem in certificate-based architectures, is not a major concern here.
- Key agreement and delegation in our certificate-free approach require much less bandwidth than the GSI. In addition, our delegation technique requires only a single verification.

- Our one-pass delegation technique exploits the properties of hierarchical identity-based cryptography to produce an efficient credential verification mechanism for a delegatee of a particular delegation, regardless of the length of the delegation chain.
- Our access control mechanism only makes use of the assumption that all members of a virtual organisation can recognise a comparatively small number of shared generic roles, which in turn fits nicely with the hierarchical namespace of our architecture.

Moreover, we showed that our proposal can interoperate with the already widely deployed PKI-based GSI without major changes to our architectural setting.

Identity-based public key cryptography is relatively new. Nevertheless, standardisation of the use of identity-based cryptographic techniques using pairings is well underway, for example, through the IEEE P1363 Working Group and the IETF [14].

For future work, we are interested in building a prototype of our proposal by modifying the existing GSI, which is built on the Generic Security Service Application Program Interface (GSS-API) [41] and the GSI-enabled OpenSSL [46], to incorporate our identity-based techniques.

References

1. Abdalla, M., Bresson, E., Chevassut, O., Möller, B., Pointcheval, D.: Provably secure password-based authentication in TLS. In: Proceedings of the 1st ACM Symposium on Information, Computer and Communications Security (ASIACCS 2006), pp. 35–45. ACM Press (2006)
2. Abdalla, M., Fouque, P., Pointcheval, D.: Password-based authenticated key exchange in the three-party setting. In: Vaudenay, S. (ed.) Proceedings of the 8th International Workshop on Theory and Practice in Public Key Cryptography—PKC 2005, LNCS, vol. 3386, pp. 65–84. Springer (2005)
3. Alsaïd, A., Mitchell, C.J.: Installing fake root keys in a PC. In: Chadwick, D., Zhao, G. (eds.) Proceedings of the 2nd European Public Key Infrastructure Workshop (EuroPKI 2005), LNCS, vol. 3545, pp. 227–239. Springer (2005)
4. Barreto, P.S.L.M., Kim, H.Y., Lynn, B., Scott, M.: Efficient algorithms for pairing-based cryptosystems. In: Yung, M. (ed.) Advances in Cryptology—Proceedings of CRYPTO 2002, LNCS, vol. 2442, pp. 354–368. Springer (2002)
5. Basney, J., Humphrey, M., Welch, V.: The MyProxy online credential repository. *J. Softw. Pract. Experience* **35**(9), 817–826 (2005)
6. Beckles, B.: Removing digital certificates from the end-user's experience of grid environments. In: Proceedings of the UK e-Science All Hands Meeting 2004, pp. 756–762 (2004)
7. Beckles, B., Welch, V., Basney, J.: Mechanisms for increasing the usability of grid security. *Int. J. Human Comput. Stud.* **63**(1–2), 74–101 (2005)
8. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated key exchange secure against dictionary attacks. In: Preneel, B. (ed.) Advances in Cryptology—Proceedings of EUROCRYPT 2000, LNCS, vol. 1807, pp. 139–155. Springer (2000)

9. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: Proceedings of the 1st ACM Computer and Communications Security Conference (CCS '93), pp. 62–73. ACM Press (1993)
10. Bellare, M., Rogaway, P.: The AuthA Protocol for Password-Based Authenticated Key Exchange. Contribution to IEEE P1363 (2000)
11. Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., Möller, B.: Elliptic curve cryptography (ECC) cipher suites for transport layer security (TLS). The Internet Engineering Task Force (IETF), RFC 4492 (2006)
12. Boneh, D., Boyen, X., Goh, E.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) Advances in Cryptology—Proceedings of EUROCRYPT 2005, LNCS, vol. 3494, pp. 440–456. Springer (2005)
13. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) Advances in Cryptology—Proceedings of CRYPTO 2001, LNCS, vol. 2139, pp. 213–229. Springer (2001)
14. Boyen, X., Martin, L.: Identity-based cryptography standard (IBCS) #1: supersingular curve implementations of the BF and BB1 cryptosystems. The Internet Engineering Task Force (IETF), RFC 5091 (2007)
15. Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. *Future Gener. Comput. Syst.* **25**(6), 599–616 (2009)
16. Chu, D.C., Humphrey, M.: Mobile OGSINET: grid computing on mobile devices. In: Proceedings of 5th IEEE/ACM International Workshop on Grid Computing (GRID 2004), pp. 182–191. IEEE Computer Society Press (2004)
17. Crampton, J., Lim, H.W.: Role signatures for access control in open distributed systems. In: Jajodia, S., Samarati, P., Cimato, S. (eds.) Proceedings of the IFIP TC-11 23rd International Information Security Conference (SEC 2008), pp. 205–219. Springer (2008)
18. Crampton, J., Lim, H.W., Paterson, K.G., Price, G.: A certificate-free grid security infrastructure supporting password-based user authentication. In: Proceedings of the 6th Annual PKI R&D Workshop 2007, pp. 103–118. NIST Interagency Report 7427 (2007)
19. Dierks, T., Allen, C.: The TLS protocol version 1.0. The Internet Engineering Task Force (IETF), RFC 2246 (1999)
20. Du, S., Joshi, J.B.D.: Supporting authorization query and inter-domain role mapping in presence of hybrid role hierarchy. In: Proceedings of the 11th ACM Symposium on Access Control Models and Technologies (SACMAT 2006), pp. 228–236. ACM Press (2006)
21. Foster, I., Kesselman, C.: Globus: a metacomputing infrastructure toolkit. *Int. J. Supercomput. Appl.* **11**(2), 115–128 (1997)
22. Foster, I., Kesselman, C. (eds.): *The Grid 2: Blueprint for a New Computing Infrastructure*. Elsevier, San Francisco (2004)
23. Foster, I., Kesselman, C., Tsudik, G., Tuecke, S.: A security architecture for computational Grids. In: Proceedings of the 5th ACM Computer and Communications Security Conference (CCS '98), pp. 83–92. ACM Press (1998)
24. Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the Grid: enabling scalable virtual organizations. *Int. J. High Perform. Comput. Appl.* **15**(3), 200–222 (2001)
25. Galbraith, S.D.: Supersingular curves in cryptography. In: Boyd, C. (ed.) Advances in Cryptology—Proceedings of ASIACRYPT 2001, LNCS, vol. 2248, pp. 495–513. Springer (2001)
26. Galbraith, S.D., Harrison, K., Soldera, D.: Implementing the Tate pairing. In: Fieker, C., Kohel, D.R. (eds.) Proceedings of the 5th International Symposium on Algorithmic Number Theory (ANTS-V), LNCS, vol. 2369, pp. 324–337. Springer (2002)
27. Galbraith, S.D., Paterson, K.G., Smart, N.P.: Pairings for cryptographers. *Discrete Appl. Math.* **156**(16), 3113–3121 (2008)
28. Gentry, C., Silverberg, A.: Hierarchical ID-based cryptography. In: Zheng, Y. (ed.) Advances in Cryptology—Proceedings of ASIACRYPT 2002, LNCS vol. 2501, pp. 548–566. Springer (2002)
29. GridCafé.: Grid Projects in the World. Available at <http://gridcafe.web.cern.ch/>
30. Gutmann, P.: Plug-and-play PKI: a PKI your mother can use. In: Proceedings of the 12th USENIX Security Symposium, pp. 45–58 (2003)
31. Hayes, J.M.: The problem with multiple roots in web browsers—certificate masquerading. In: Proceedings of the IEEE 7th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprise, pp. 306–311. IEEE Computer Society Press (1998)
32. Horwitz, J., Lynn, B.: Towards hierarchical identity-based encryption. In: Knudsen, L.R. (ed.) Advances in Cryptology—Proceedings of EUROCRYPT 2002, LNCS, vol. 2332, pp. 466–481. Springer (2002)
33. Housley, R., Polk, W., Ford, W., Solo, D.: Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile. The Internet Engineering Task Force (IETF), RFC 3280 (2002)
34. Humphrey, M., Thompson, M.R., Jackson, K.R.: Security for grids. *Proc. IEEE* **93**(3), 644–652 (2005)
35. Kleinjung, T., Aoki, K., Franke, J., Lenstra, A., Thomé, E., Bos, J., Gaudry, P., Kruppa, A., Montgomery, P., Osvik, D.A., te Riele, H., Timofeev, A., Zimmermann, P.: Factorization of a 768-bit RSA modulus. Cryptology ePrint Archive, Report 2010/006 (2010). Available at <http://eprint.iacr.org/2010/006>
36. Kornievskaja, O., Honeyman, P., Doster, B., Coffman, K.: Kerberized credential translation: a solution to web access control. In: Proceedings of the 10th USENIX Security Symposium, pp. 235–250 (2001)
37. Krawczyk, H.: The order of encryption and authentication for protecting communications (or: how secure is SSL?) In: Kilian, J. (ed.) Advances in Cryptology—Proceedings of CRYPTO 2001, LNCS, vol. 2139, pp. 310–331. Springer (2001)
38. Lim, H.W.: On the Application of Identity-Based Cryptography in Grid Security. Ph.D thesis, University of London (2006)
39. Lim, H.W., Paterson, K.G.: Identity-based cryptography for grid security. In: Stockinger, H., Buyya, R., Perrott, R. (eds.) Proceedings of the 1st IEEE International Conference on e-Science and Grid Computing (e-Science 2005), pp. 395–404. IEEE Computer Society Press (2005)
40. Lim, H.W., Robshaw, M.J.B.: On identity-based cryptography and GRID computing. In: Bubak, M., Albada, G.D.v., Sloot, P.M.A., Dongarra, J.J. (eds.) Proceedings of the 4th International Conference on Computational Science (ICCS 2004), LNCS, vol. 3036, pp. 474–477. Springer (2004)
41. Linn, J.: Generic security service application program interface version 2, update1. The Internet Engineering Task Force (IETF), RFC 2743 (2000)
42. Mao, W.: An Identity-based Non-interactive Authentication Framework for Computational Grids. HP Lab, Technical Report HPL-2004-96 (2004)
43. Moore, P.C., Johnson, W.R., Detry, R.J.: Adapting Globus and Kerberos for a secure ASCII Grid. In: Proceedings of the 2001 ACM/IEEE Conference on Supercomputing (SC2001), CD-ROM, p. 21. ACM Press (2001)
44. Neuman, B.C., Ts'o, T.: Kerberos: an authentication service for computer networks. *IEEE Commun.* **32**(9), 33–38 (1994)
45. Novotny, J., Tuecke, S., Welch, V.: An online credential repository for the Grid: MyProxy. In: Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10 2001), pp. 104–111. IEEE Computer Society Press (2001)

46. The OpenSSL Project: OpenSSL: The Open Source Toolkit for SSL/TLS (2010). Available at <http://www.openssl.org/>
47. Paterson, K.G., Price, G.: A comparison between traditional public key infrastructures and identity-based cryptography. *Inf. Secur. Tech. Report* **8**(3), 57–72 (2003)
48. Paulson, L.C.: Inductive analysis of the Internet protocol TLS. *ACM Trans. Inf. Syst. Secur.* **2**(3), 332–351 (1999)
49. Phan, T., Huang, L., Dulun, C.: Challenge: integrating mobile wireless devices into the computational grid. In: *Proceedings of the 8th ACM International Conference on Mobile Computing and Networking (MOBICOM 2002)*, pp. 271–278. ACM Press (2002)
50. Price, G., Mitchell, C.J.: Interoperation between a conventional PKI and an ID-based infrastructure. In: Chadwick, D., Zhao, G. (eds.) *Proceedings of the 2nd European Public Key Infrastructure Workshop (EuroPKI 2005)*, LNCS, vol. 3545, pp. 73–85. Springer (2005)
51. Sandhu, R.S., Bellare, M., Ganesan, R.: Password-enabled PKI: virtual smartcards versus virtual soft tokens. In: *Proceedings of the 1st Annual PKI R&D Workshop*, pp. 89–96 (2002)
52. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) *Advances in Cryptology—Proceedings of CRYPTO '84*, LNCS, vol. 196, pp. 47–53. Springer (1985)
53. Shamus Software Ltd.: MIRACL. Available at <http://www.shamus.ie/>
54. Steiner, M., Buhler, P., Eirich, T., Waidner, M.: Secure password-based cipher suite for TLS. *ACM Trans. Inf. Syst. Secur.* **4**(2), 134–157 (2001)
55. The TeraGrid Project.: TeraGrid. Available at <http://www.teragrid.org/>. Last accessed Jan 2009
56. Tuecke, S., Welch, V., Engert, D., Pearlman L., Thompson M.R.: Internet X.509 public key infrastructure proxy certificate profile. The Internet Engineering Task Force (IETF), RFC 3820 (2004)
57. Wagner, D., Schneier, B.: Analysis of the SSL 3.0 protocol. In: *Proceedings of the 2nd USENIX Workshop on Electronic Commerce*, pp. 29–40 (1996)
58. Welch, V., Foster, I., Kesselman, C., Mulmo, O., Pearlman, L., Tuecke, S., Gawor, J., Meder, S., Siebenlist, F.: X.509 proxy certificates for dynamic delegation. In: *Proceedings of the 3rd Annual PKI R&D Workshop*, pp. 42–58. NIST Interagency Report (2004)
59. Welch, V., Siebenlist, F., Foster, I., Bresnahan, J., Czajkowski, K., Gawor, J., Kesselman, C., Meder, S., Pearlman, L., Tuecke, S.: Security for Grid services. In: *Proceedings of the 12th IEEE International Symposium on High Performance Distributed Computing (HPDC-12 2003)*, pp. 48–61. IEEE Computer Society Press (2003)