An Identity-based Non-interactive Authentication Framework for Computational Grids

Wenbo Mao
Trusted Systems Laboratory
HP Laboratories Bristol
HPL-2004-96
June 3, 2004*

E-mail: wenbo.mao@hp.com

We examine the authentication framework for Globus Security Infrastructure (GSI, the current grid security standard) and identify a weakness of poor scalability due to heavy interactions between a user-side client and many resource contribution sites. We propose an alternative authentication framework for GSI using authenticated session keys which are shared between two parties without any interactions between them. Our proposal is enabled by an emerging cryptographic technique from the bilinear pairing.

An Identity-based Non-interactive Authentication Framework for Computational Grids

Wenbo Mao
Hewlett-Packard Laboratories, Bristol
United Kingdom
wenbo.mao@hp.com

May 29, 2004

Abstract

We examine the authentication framework for Globus Security Infrastructure (GSI, the current grid security standard) and identify a weakness of poor scalability due to heavy interactions between a user-side client and many resource contribution sites. We propose an alternative authentication framework for GSI using authenticated session keys which are shared between two parties without any interactions between them. Our proposal is enabled by an emerging cryptographic technique from the bilinear pairing.

1 Background

A computational grid is a distributed computing system comprising a large number of sites of computational resources from which a virtual organization (VO) of high performance services can be combined for use by demanding users. In the most general setting, these large number of resource contributing sites form different trust domains. This means that in the time of a VO setting-up, a user (more precisely, a user's proxy, UP, which is a user side client machine acting on behalf of the user) must conduct potentially a large number of instances of mutual authentication with these resource contributing sites (each site is managed by a resource proxy RP) in order to gain secure access to them.

The current grid security standard, Grid Security Infrastructure (GSI, [6], the security kernel of the Globus Toolkit), enables a secure way of VO setting-up with a set of (four) security protocols which involve mutual entity authentication between UP and RP. Two of the four GSI security protocols, called Protocol 2 and Protocol 3, are for resource allocation from a UP and that from a process, respectively. These protocols run between UP and a resource proxy RP to achieve mutual authentication between these two entities. Entity authentication in these protocols are achieved by applying the standard SSL Authentication Protocol (SAP, [10, 7], also see Chapter 12 of [13] for a comprehensive description of the SSL/TLS Protocols). Here, UP and RP have public-key cryptographic credentials called identity certificates which are under the organization of the standard certification-based public-key authentication infrastructure X.509 (PKI X.509 [11], also see Chapter 13 of [13]). Figure 1 depicts the security architecture of GSI.

In GSI, because different resource contributing sites form different trust domains, using resources in these sites by a user U requires entity authentication of U to RP in each of these sites. In fact, mutual authentication between U and these RPs are necessary since

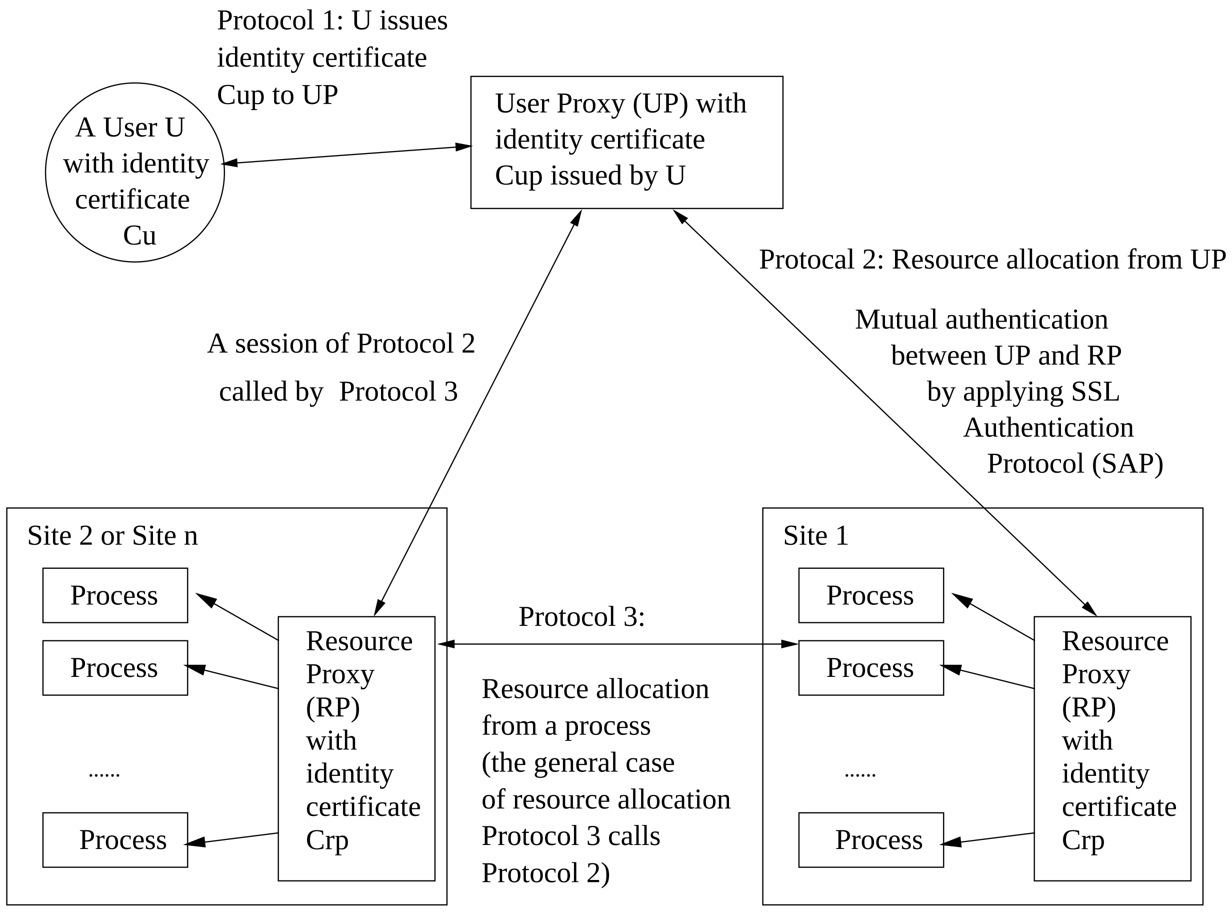
Protocol 1: U issues identity certificate Cup to UP
A User U with identity certificate Cu
User Proxy (UP) with identity certificate Cup issued by U
Protocal 2: Resource allocation from UP
Mutual authentication between UP and RP by applying SSL Authentication Protocol (SAP)
A session of Protocol 2 called by Protocol 3
Site 2 or Site n
Process
Process
......
Process
Resource Proxy (RP) with identity certificate Crp
Protocol 3:
Resource allocation from a process (the general case of resource allocation Protocol 3 calls Protocol 2)
Site 1
Process
Process
......
Process
Resource Proxy (RP) with identity certificate Crp

Figure 1: Globus Security Infrastructure

U must also guard itself against a spoofing "resource contributing site". The execution of authentication at the end of U is actually performed by the user proxy UP who is a client machine of U acting on behalf of U. Entities U and RP have long-term cryptographic credentials, which are their X.509 identity certificates, denoted by C_U and C_{RP}, respectively. These certificates are issued by some grid certification authorities (grid CAs, Figure 1 has not shown the PKI certification structure). The identity certificate of UP, denoted by C_{UP}, is issued by U using Protocol 1 (see Figure 1) and has a short life time. The private key of UP (its cryptographic credential) can be simply put in the file system of UP protected under the access control of the operating system of UP. Because C_{UP} has a short life time, this "casual" protection suffices, and it enables an important feature of *unattended user authentication* which allows UP to conduct entity authentication on behalf of U when the latter is not present.

Figure 1 conceptually illustrates two resource contributing sites only. First, Site 1 is discovered by Globus Toolkit for UP as a result of responding to UP's resource allocation request; this resource allocation from UP invokes a session of Protocol 2 running between UP and RP in Site 1 which will enable RP to allocate some number of processes for use by UP. Next, because the demand for resources from UP should in general be greater than what Site 1 can allocate, Site 2 will further be discovered by Globus Toolkit as a result of responding to a resource allocation request from a process which has been allocated to

UP. This second case of resource allocation from a process invokes a session of Protocol 3 running between a process in Site 1 and the second RP in Site 2. This session then subsequently calls a new session of Protocol 2 to run between UP and the second RP in Site 2. Resource allocation from a process is in fact the general case of resource allocation in Globus, namely, viewing Site 2 as Site n for $n \gg 1$ is the general case.

In general, a grid VO involves resource contributed from a large number of sites (GUSTO, a US grid testbed has initially 20 sites [6]). It is conceivable that the number of sites in a future grid may be in the order of 10^2. The involvement of a large number of resource contributing sites is the very reason why the grid is capable of providing very high-performance computational services. Thus, during a VO setting-up, UP in general will have to execute potentially a large number of mutual authentication sessions. In each session, UP has to conduct many expensive operations; these include:

i) Issuing a digital signature to prove to an RP that it is holding a valid cryptographic credential;

ii) Verifying a digital signature of RP to deduce that RP holds a cryptographic credential;

iii) Verifying the identity certificate C_{RP} of RP issued by a grid CA;

iv) Verifying the certificates of CAs upward until reaching that of a recognized; the completion of certificates verification in (iii) and (iv) allows UP to conclude that the digital signature of RP is valid; and

v) Agreeing with RP a shared session cryptographic key; this typically consists of random number generation, round-trip communication interactions with RP, and some key exchange operations (either a public-key encryption or a Diffie-Hellman-like key exchange).

In these operations, each certificate verification involves verifying a digital signature. Session key exchange operations in (v) also involves public-key cryptographic operations. In addition, (v) involves a couple of rounds of communication. Notice that these operations must be conducted by UP for each session of Protocol 2 run with an RP in a resource contributing site. In addition, UP must also maintain the same large number of authenticated sessions, each of which is secured by a session key agreed in (v). The maintenance of secure sessions will have an additional cost which we shall discuss later.

Thus, in GSI, UP is in a computationally heavily loaded point both in computation and in communication. Public-key operations such as signature creation and verification, and session key exchange are in general quite computationally demanding. Moreover, communication interactions can be quite latent especially when UP has to maintain a large number of communications at the same time. Despite, we should notice the fact that UP is in general a user-end average computer platform.

The authors of GSI conceded that the current GSI technique has a poor scalability which limits the number of sessions of Protocol 2 run by UP [6]. This means a limit for a user on the number of resource contributing sites it can use. We believe that this scalability problem is an inherent one due to the use of the standard X.509 certificate-based PKI authentication framework: applying this framework it doesn't seem to exist a better scalability for the user client machine.

However poor a scalability the current GSI technique is, we should notice two important features which are necessary for a grid security solution and which GSI has solved nicely from applying the standard X.509 PKI:

1. User single-sign-on: U only needs to register once with a grid certificate authority (CA) to obtain a long-term cryptographic credential which is its identity certificate C_U; using C_U and the corresponding private key, U further issues (digitally signs) a short-term cryptographic credential C_{UP}, an identity certificate for UP to use.

2. Unattended user authentication: with C_{UP} and the corresponding private key, UP can conduct authentication sessions on behalf of U even when U is not present; this is very important since the duration of a grid can be sufficiently long and new resource allocation request may be needed after U has left and so the protocols have to run by UP in an unattended manner.

With these two features being necessary for grid services, our attempt to improve the scalability of GSI must not lead to any reduction to the quality of these two features.

1.1 Our Contribution

We apply a novel cryptographic technique which enables authenticated session key sharing between two parties without them having ever interacted to one another. This technique enables (1) identity-based entity authentication which saves much computation that a user-site client has to perform in the current GSI for verifying a large number of X.509 identity certificates, and more significantly, (2) batching a large number of authentication sessions to reduce the communication complexity for the client, which in the current GSI has to maintain a large number of communications with various resource contributing sites.

1.2 Organization

In Section 2 we describe a novel cryptographic technique which is the kernel technical basis for the proposed scheme. In Section 3 we describe our scheme in detail. In Section 4 we summarize the advantages of our scheme. In Section 5 we provide some necessary discussions. Finally we conclude the work in Section 6.

2 Identity-based Non-interactively Exchanged Authenticated Session Keys

We now describe a novel cryptographic technique which enables two parties to share authenticated session keys without any interaction between them.

Using the notation of Boneh and Franklin [2], we let $\mathbb{G}_1$ be an additive group of prime order q and $\mathbb{G}_2$ be a multiplicative group of the same order q. We assume the existence of an efficiently computable, non-degenerate, bilinear map $\hat{e}$ from $\mathbb{G}_1 \times \mathbb{G}_1$ to $\mathbb{G}_2$. Typically, $\mathbb{G}_1$ will be a subgroup of the group of points on a super-singular elliptic curve over a finite field, $\mathbb{G}_2$ will be a subgroup of the multiplicative group of a related finite field and the map $\hat{e}$ will be derived from either the Weil or Tate pairing on the elliptic curve. By $\hat{e}$ being bilinear, we mean that for $P, Q, R \in \mathbb{G}_1$, both

$$\hat{e}(P+Q, R) = \hat{e}(P, R) \cdot \hat{e}(Q, R) \quad \text{and} \quad \hat{e}(P, Q+R) = \hat{e}(P, Q) \cdot \hat{e}(P, R).$$

By $\hat{e}$ being non-degenerate, we mean that for non-unity element $P \in \mathbb{G}_1$, we have $\hat{e}(P, P) \neq 1_{\mathbb{G}_2}$.

When $a \in \mathbb{Z}_q$ and $P \in \mathbb{G}_1$, we write aP for P added to itself a times, also called scalar multiplication of P by an integer a. As a consequence of bilinearity, we have that, for any $P, Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q$:

$$\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab} = \hat{e}(abP, Q) = \hat{e}(P, abQ).$$

The finite field containing G_2 as a subgroup typically uses a security parameter k which is the same as that for most popular public-key cryptographic systems, such as RSA or discrete-logarithm based systems, in order to obtain a degree of security protection similar to that in those popular public-key cryptographic systems. Consequently, the cost of computing a bilinear pairing is similar to that of computing a public-key cryptographic operation in those popular cryptographic systems. We refer to [14, 1, 2, 8] for a more comprehensive description of how these groups, pairings and other parameters should be selected in practice for efficiency and security.

In the above setting, an initialization function I on input the security parameter k selects suitable groups $\mathbb{G}_1$, $\mathbb{G}_2$ and map $\hat{e}$. Then I generates a random key $s \in \mathbb{Z}_q$. This key will play the role of the master secret of a Trusted Authority (TA) in the identity-based key sharing system. Upon a key registration request from a party P whose identity we also denote by P, TA issues P a key pair consisting of public key $Q = H(P)$ and private key $S = sQ$. Here H is a cryptographic hash function deterministically mapping strings in $\{0,1\}^*$ onto $\mathbb{G}_1$. Under the discrete logarithm assumption (i.e., it is a hard problem to solve the discrete logarithm problem), P cannot find the master secret s of TA from its key pair (Q, sQ).

Now any two principals with identities P_i, P_j, after having register with TA, can efficiently calculate the shared key between them by computing

$$K_{ij} = \hat{e}(S_i, Q_j) = \hat{e}(Q_i, Q_j)^s = \hat{e}(S_j, Q_i). \tag{1}$$

Here, party P_i (respectively, P_j) derives the shared key using the left-hand side (respectively, right-hand side) pairing computation: i.e., it pairs its private key with the other part's public key P_j (respectively, P_i) which is simply the other party's identity. This method of *identity-based, non-interactive key distribution* is due to Sakai et al [15]. Without counting TA, the shared key K_{ij} computed in (1) is exclusively available to the two parties P_i and P_j. This is due to the difficulty of solving the bilinear Diffie-Hellman Problem which is defined below.

Bilinear Diffie-Hellman Problem (BDHP): Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\hat{e}$ be as above. The BDHP in $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$ is as follows: Given $\langle P, xP, yP, zP \rangle$ with $P \in \mathbb{G}_1$ and $x, y, z \in \mathbb{Z}_q$, compute $\hat{e}(P, P)^{xyz} \in \mathbb{G}_2$. An algorithm A has advantage ϵ in solving the BDHP in $\langle \mathbb{G}_1, \mathbb{G}_2, e \rangle$ if

$$\Pr\left[\mathrm{A}(\langle P, xP, yP, zP \rangle) = \hat{e}(P, P)^{xyz}\right] = \epsilon.$$

Here the probability is measured over random choices of $P \in \mathbb{G}_1$, $x, y, z \in \mathbb{Z}_q$, the random operations of the algorithm A and the all possible algorithms. We will use the Bilinear Diffie-Hellman Assumption, which states that, for all efficient algorithms A, the advantage ϵ is negligible as a function of the security parameter k used in generating the system parameters $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$. Because $\mathbb{G}_1$ is of a prime order and hence is cyclic, we can consider that Q_i, Q_j are generated by a generator point P, i.e., $Q_i = aP$ and $Q_j = bP$ for some integers a and b; then knowing P, aP, bP, xP to find $K_{ij} = \hat{e}(P, P)^{abx}$ is clearly an instance of the Bilinear Diffie-Hellman Problem described above. This is why we have claimed that the shared key computed in (1) by parties P_i and P_j is an exclusive secret to these two parties (without counting TA).

It is important to notice that the computation of the shared secret key in (1) does not involve any data exchange between the two parties. Any two parties, after both having registered with TA, already share such a non-interactively exchanged key between them even they have not ever communicated one another!

An exclusively shared secret key can of course be used to enable mutual authentication between the two key sharing parties. For example, when party P_i receives

$$P_j, M, f(K_{ij}, P_j, M) \tag{2}$$

where $f()$ is a one-way hash function, and P_i itself has not computed $f(K_{ij}, P_j, M)$, it knows that it is P_j who has computed the value since besides itself, only P_j is the party who has also in possession of the correct cryptographic credential (its private key issued by TA). So the authorship or the origin of the message M is authenticated. If P_i could further deduce the freshness of $f(K_{ij}, P_j, M)$ (we shall see a method in the next section), then it can further conclude that P_j has corresponded recently.

The message tuple in (2) forms a statically keyed authenticator for secure entity authentication under a well-known formal security model for provably secure authentication protocols [4]. Boyd, Mao and Paterson conducted a rigorous formal proof on the security for this authenticator under that formal security model [3].

We should further notice that, if P_i, P_j are public keys and at the same time are uniquely identifiable distinguished names (DNs), then they are non-random. Consequently, this authentication technique needn't involve public key certification, which is usually needed in the case of applications of conventional public-key techniques because in that case a public key looks random and has to be certified by a well-known party.

The elimination of public-key certification is a non-trivial achievement from the identity-based cryptography. The system complexity of X.509 certification based PKI has been a big obstacle for a wide deployment of public-key cryptography in applications. However, we shall further see in moment that the surprising feature of two parties being able to share a cryptographic session key without interaction will have a bigger impact on the performance improvement for GSI.

3 An Identity-based Non-interactive Authentication Framework for the Grid

The identity-based authentication technique described in the preceding section implicitly assumes that the public key of a party is simply its identity string. This simple understanding has the following problem: if a user's private key is compromised, then how can its public key credential, i.e., its identity, be revoked? An obvious solution to this problem is to change the party's identity.

Indeed, in our proposed application of the identity-based authentication for the grid security, we stipulate that a user's identity has two parts, one part is a string of a distinguished name in the usual sense, and the other part is a string which specifies a validity period. The identity-based public key of a user is the concatenation of the two strings. We now describe a identity-based key management scheme which suits the grid security application.

3.1 An Identity-based Key Management Scheme for GSI

Before becoming a grid user member, a user U must register its DN identity (which we still denote by U) with a grid trusted authority GTA. This initial registration is what can be called a user single-sign-on (SSO) session. In an SSO session, GTA conducts a thorough identity validation on U. A successful SSO session will result in entering of U into the database of GTA as a unique DN.

Suppose that now U has successfully registered from an SSO session. For a period p, GTA will issue U a *period private key* which corresponds to the period public key $U||p$. Here p is a string specifying the current period and "$||$" denotes string concatenation. For example, p can be a date value specified in a universal time such as Greenwich Mean Time. Knowing the DN U and the current period p, any party can compute the period public key $U||p$. From the formulation of the identity-based public/private key pair described in Section 2 we know that the period private key corresponding to the period public key $U||p$ is $sH(U||p)$. The period private key can be sent to U (more precisely, to UP) via a secure channel, for example, an SSL session between UP (as a client) and GTA (as a server). To reduce the work load of GTA, U on UP can initiate an SSL session to pull its period private key from GTA. This SSL pull session only needs to be invoked when U wishes to use a grid service. The cryptographic basis for mutual authentication between U and GTA in the SSL pull session can be any one of several standard means, e.g., smartcard based (a smartcard holds U's private key which has been setup in the SSO session), user password based, or one-time password authentication token based (e.g., from a SecureID taken). This secure pull session is of course possible after the initial SSO session between U and GTA, which has setup the necessary cryptographic credential for U.

The length of the validity period p can be the same as what GSI stipulates on the lifetime for the conventional public-key certificate of UP. Thus, the period private key $sH(U||p)$ can also be stored in UP under the protection of UP's operating system's access control mechanism, i.e., the period private key $sH(U||p)$ is treated with the same level of protection as that GSI provides on the conventional private key of UP. This method enables unattended user authentication: UP can use the period private key even when U is not present.

For each RP to become a grid resource server, we also suppose that RP registers its DN (which we still denote by RP) with GTA. Hence, for a period q, RP will receive its period private key $sH(RP||q))$. It is reasonably conceivable that the validity period q for RP can be much longer than the validity period p for U since RP, as a server in a secure domain, can easily implement a strong mechanism (e.g., hardware based) to protect a private key. For this reason and for exposition simplicity, in the following description we shall omit the presentation of RP's validity period, i.e., we will simply use RP in place of $RP||q$.

We shall name the protocol for initial user registration "Single-Sign-On" Protocol, and the protocol for U to pull an identity-based period private key "Pull-Period-Identity-Key" Protocol.

3.2 Non-interactive Entity and Message Authentication

In the current version of GSI, the main security protocol for resource allocation, Protocol 2, begins with a session of mutual entity authentication between UP and RP. This authentication session applies the SSL Authentication Protocol (SAP). A successful SAP session outputs the following two things:

i) A belief by each of the two parties that the other end of the communication is the party who is claiming to be, i.e., UP (RP) believes that the other end is indeed RP (UP);

ii) A shared secret session key which can be used to encrypt data transmissions in a follow-up session of secure communication to take place in the transport layer. Remaining tasks in Protocol 2 (and Protocol 3 after it calls Protocol 2) do require transmissions of encrypted data between the two parties.

It is now evident that the technique of identity-based non-interactively exchanged session key can also output these two things to UP and RP. These two parties, knowing the unique identity strings U and RP and the current period p, can compute the current period session key

$$K_p = \hat{e}(sH(U||p), H(RP)) = \hat{e}(H(U||p), sH(RP)).$$

With this period session key, entity authentication, e.g., from U to UP, can be achieved by the following single message transmission:

$$U \rightarrow RP : U, M, f(K_p, U, RP, M). \tag{3}$$

Here f is a one-way hash function which can be evaluated extremely fast. Notice that this entity authentication does not involve validating any certificate by any party. RP is convinced that U is indeed the one who claims to be since the computation of the correct hashed value needs using the period session key whose computation in turn needs using the correct private key matching the period identity $U||p$; thus, being able to compute the hashed value implies that this private key must have been issued by GTA and hence U is a grid member granted by GTA. This forms the belief (of RP with respect to U) which we have described in (i) above. Moreover, the period session key K_p is exactly the second output described in (ii) above.

The other direction of entity authentication, i.e., that from UR to U, is analogous.

We point out that because the period private key is fresh, these two parties can conclude that the communication partner is corresponding lively. Boyd, Mao and Paterson provided a rigorous formal proof on the security of this style of entity authentication from using static (i.e., non-interactively shared) keys [3].

3.3 Saving in Computation

Comparing UP's computational load in our proposal (one pairing evaluation and one hash function evaluation) with the five public key operations that a UP in the current GSI has to perform (review UP's actions listed in Section 1), UP in our proposal performs about 1/5 fraction of it has to perform in the current GSI. This measure is based on that the cost of computing the period session key K_p by evaluating one bilinear pairing is similar to that for one instance of verifying a digital signature (see quantitative analysis in [14]).

We would like to express that being able to save UP from validating a large number of identity certificates of RPs and those of CAs is only a small part of the performance improvement that our proposal offers. We now explain a more significant performance improvement which is obtained from non-interactive authentication.

3.4 Significance of Non-interactive Authentication

An important element in the grid which has not been illustrated in Figure 1 is an entity called Resource Broker (RB) who discovers resources for UP. Typically, an RB arbitrates between different requests for resources. Therefore, Figure 1 should actually have shown an RB in between UP and various RPs who are discovered by RB for UP.

With RB working in between UP and various RPs, and now with session keys non-interactively shared between UP and various RPs, entity authentication sessions between UP and various RPs can be done *in batch* which are tunneled through RB. Such batched authentication sessions tunneled via RB can be as follows.

Batched Authentication and Authenticated Session Key Agreement

1. RB discovers a sufficient number of RPs, and sends their DNs to UP;
2. For each RP introduced from RB, UP computes a tuple $(RP, M, f(K_p, U, RP, M))$; but instead of sending the tuple directly to RP as in (3), UP sends these tuples in a batch to RB;
3. RB sends to RP the tuple $(U, M, f(K_p, U, RP, M))$; it does so for each RP it has discovered for UP; then RB waits for response $(RP, M', f(K_p, RP, U, M'))$ from RP;
4. Upon receipt of all responses from all targeted RPs, or upon a pre-set timeout, RB forwards the responses in batch to UP; these will enable these RPs to be authenticated to UP.

We believe that the batched entity authentication sessions described here will form a significant performance improvement for UP, and hence for the grid. Our belief is based on the fact that communication interactions over the internet can be quite latent, and the latency can become critical at the point of UP in the current GSI since UP has to interact with a large number of RPs at the same time.

We also believe that it is the novel cryptographic technique described in Section 2 for authenticated session key sharing between two parties without interaction that is the kernel element to have enabled the simple way of batching a large number of entity authentication sessions with agreement of shared authenticated session keys.

3.5 Security of Grid Trusted Authority

Since a GTA distributes private keys for system wide users, it can masquerade as any user and any resource proxy. This problem can be resolved by using a plural number of GTAs who collectively share the system master key s. Let GTA_i denote an individual GTA whose master secret key is s_i. Then U can obtain its period private key from a list of GTAs. By adding a list of period private keys, the combined period private key is

$$\sum_i s_i H(U||p) = (\sum_i s_i) H(U||p).$$

Suppose that these GTAs do not collude, then no one will know the combined system master key $\sum_i s_i$ and so this private key is exclusively known to U.

Using multiple GTAs also allows a flexibility for U to choose a subset of favorable GTAs among a list. However, a choice of U must be made available to an RP so that it will also use the same choice to enable the correct session key sharing.

More sophisticated secret sharing method, such as threshold secret sharing, e.g., [16] with publicly verifiable correctness, e.g., [17] can also be applied. However, the non-trivial costs of these techniques need to be carefully analyzed against the frequency GTAs are used.

4 Summary

We now summarize the properties of the proposed non-interactive authentication scheme for the grid.

- The proposed identity-based entity authentication scheme involves no X.509 style of PKI certificates validation. Each party computes one pairing which has a computational cost comparable to one usual public-key cryptographic operation. Mutual entity authentication based on the agreed period session key can use symmetric cryptographic technique, such as keyed one-way hash function, which is extremely fast.

- The non-interactively shared authenticated session keys between UP and various RPs enable entity authentication sessions between these two parties to be conducted in batch to be tunneled through a resource broker. This will greatly reduces the communication complexity for UP.

- This is a public-key authentication framework. Notice that unlike a Kerberos based authentication framework [5], the two parties does not use on-line trusted third party. Therefore the technique suits to serve an open system such as the grid.

- The period session key can be updated "on-the-fly": if a grid service lasts longer than a period p specifies, i.e., if a grid service started in the period p remains alive and enters the next period p', secure communications between two parties can continue smoothly in that the two parties can compute the new period session key $K_{p'}$ to replace K_p. In contrast, a similar session updating in the current GSI (when a grid service lasts longer than the lifetime of UP's identity certificate) must involve U to rerun Protocol 1 to create a new identity certificate for UP, followed by UP to rerun Protocol 2, etc.

- The proposed scheme, which includes the "Pull-Period-Identity-Key" Protocol described in §3.1 and the "Batched Authentication" Protocol described in §3.4, can be implemented by purely using commercial-off-the-shelf software, such as SSL. Therefore the proposed scheme can be easily adapted by the web-based Open Grid Services Architecture (OGSA).

5 Discussion

One may observe that, our scheme unloads the computational burden of UP, which is an ordinary user client platform, and uploads it on to GTA for the latter to compute the period private key for U.

While this is true in terms of computation, we believe that this shift of the computational load is reasonable. A GTA as a dedicated server should be sufficiently capable of maintaining the task of period private key computations for a system wide users. Note that a GTA only needs to perform a period private key computation for a user upon a key pull session invoked by the user.

In terms of communication, it is evident that the batched authentication sessions in our scheme achieves a great deal of UP's communication complexity by eliminating the need of maintaining many rounds of communications with many RPs, some of which can be very latent.

One may also argue that our scheme requires GTAs to stay on-line, while CAs in the current GSI needn't do so. Strictly speaking, the system of X.509 certificate infrastructure can have never achieved off-line authentication servers: in a real use of such a system, one must always check a certificate revocation list maintained by a certification revocation authority which therefore must always be on-line. We consider that the working load of our on-line GTA is comparable to that of an on-line certificate revocation authority.

6 Conclusion

We have proposed an identity-based, public-key, non-interactive authentication framework for the grid security. Our proposal should improve the user side performance for the current GSI authentication scheme in a considerable degree. The performance improvement is in both computation and communication. The improvement in communication due to being able to batch authentication sessions via a resource broker is significant.

Our focus has been on improving the performance (hence scalability) of the mutual entity authentication problem in GSI. Mutual entity authentication is the main task that GSI faces (GSI uses "identity certificates" which means that its attention is on the user's identity rather than what a user is authorized to do). For this reason, we have not discussed any authorization issue in our proposal (authorization is another major issue in the grid). In this regard, we follow the GSI approach which considers that authorization and some other security services could be based on entity authentication.

Finally, we envision that entity authentication and session key exchange using the non-interactively shared authenticated keys based on the identity-based public-key framework, which we have proposed for GSI in this paper, should have wider applications in Internet Security [9, 12] (also see Chapter 12 of [13]) in which one may expect that IP addresses concatenated with a validity period to be the basis for identity-based public keys.

Acknowledgments

The work is improved by comments from Miranda Mowbray, Nigel Edwards and John Erickson on a earlier draft.

References

[1] P. S. L. M. Barreto, H. Y. Kim, B. Lynn, and M. Scott. Efficient algorithms for pairing-based cryptosystems. In *Advances in Cryptology - CRYPTO 2002*, Lecture Notes in Computer Science. Springer-Verlag, 2002.

[2] D. Boneh and M. Franklin. Identity based encryption from the Weil pairing. In J. Killian, editor, *Advances in Cryptology — Proceedings of CRYPTO'01, Lecture Notes in Computer Science 2139*, pages 213–229. Springer-Verlag, 2001.

[3] C. Boyd, W. Mao and K. Paterson. Key agreement using statically keyed authenticators Accepted by 2004 Applied Cryptography and Network Security (ACNS'04),

Yellow Mountain, China, June 2004, to appear in Lecture Notes in Computer Science, Springer, 2004. `www.hpl.hp.com/personal/wm/papers/bmp.pdf`.

[4] Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In *Advances in Cryptology – Eurocrypt 2001*, volume 2045 of *LNCS*, pages 453–474. Springer-Verlag, 2001. `eprint.iacr.org/2001/040.ps.gz`.

[5] D. Davis and R. Swick. Workstation services and Kerberos authentication at Project Athena. Technical Memorandum TM-424, MIT Laboratory for Computer Science, February 1990.

[6] I. Foster, C. Kesselman, G. Tsudik and S. Tuecke. A security architecture for Computational Grids, 5th ACM Conference on Computer and Communications Security, pages 83–92, 1998.

[7] A.O. Freier, P. Karlton, and P.C. Kocher. The SSL Protocol, Version 3.0. INTERNET-DRAFT, draft-freier-ssl-version3-02.txt, November 1996.

[8] S.D. Galbraith, K. Harrison, and D. Soldera. Implementing the Tate pairing. In *Algorithmic Number Theory 5th International Symposium, ANTS-V*, volume 2369 of *Lecture Notes in Computer Science*, pages 324–337. Springer-Verlag, 2002.

[9] D. Harkins and D. Carrel. The Internet key exchange protocol (IKE). The Internet Engineering Task Force Request For Comments (IETF RFC) 2409, November 1998. Available at `www.ietf.org/rfc/rfc2409.txt`.

[10] K.E.B. Hickman. The SSL Protocol. Online document, February 1995. Available at `www.netscape.com/eng/security/SSL_2.html`.

[11] ITU-T. Rec. X.509 (revised) the Directory — Authentication Framework, 1993. International Telecommunication Union, Geneva, Switzerland (equivalent to ISO/IEC 9594-8:1995.).

[12] C. Kaufman. Internet Key Exchange (IKEv2) Protocol. The Internet Engineering Task Force: INTERNET-DRAFT, draft-ietf-ipsec-ikev2-03.txt, October 2002. Available at `www.ietf.org/internet-drafts/draft-ietf-ipsec-ikev2-03.txt`.

[13] W. Mao. *Modern Cryptography: Theory and Practice*, Prentice-Hall PTR, 2003.

[14] W. Mao and K. Harrison. Divisors, bilinear pairings and pairing enabled cryptographic applications. Online presentation, Available at `www.hpl.hp.com/personal/wm/research/pairing.pdf`.

[15] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing. In *Proceedings of the 2000 Symposium on Cryptography and Information Security, Okinawa, Japan*, January 2000.

[16] A. Shamir. How to share a secret. Communications of the ACM, Vol 22 (1979) pages 612–613.

[17] M. Stadler. Publicly verifiable secret sharing. Advances in Cryptology: Proceedings of EUROCRYPT 96 (U. Maurer, ed.), Lecture Notes in Computer Science 1070, Springer-Verlag (1996), pages 190–199.