Performance of Cryptographic Protocols for High-Performance, High-Bandwidth and High-Latency Grid Systems

Himanshu Khurana, Radostina Koleva and Jim Basney
National Center for Supercomputing Applications (NCSA),
University of Illinois, Urbana-Champaign, IL 61801
{hkhurana, jbasney}@ncsa.uiuc.edu, rkoleva@hcc.mass.edu

Abstract

Grid security solutions address critical requirements and are constructed from building blocks comprising cryptographic techniques and protocols. Today, the Globus toolkit's Grid Security Infrastructure (GSI) is a leading provider of such solutions and is based on the RSA cryptosystem. Recently, cryptosystems based on bilinear pairings have been proposed as an alternative with the promise of increased efficiency and scalability. In this work we evaluate this alternative and Elliptic Curve Cryptography (ECC) with RSA in terms of their computation and communication overheads for practical Grid systems. Our analysis and experiments lead us to conclude that RSA is a reasonable choice for the foreseeable future in terms of performance. Furthermore, our results are intuitively aligned with the high-performance, high-bandwidth, and high-latency nature of Grid systems.

1 Introduction

"Grid" systems and applications provide end users with a set of integrated and managed resources that are distributed across multiple organizations or "administrative domains". To do so, "virtual organizations" (VOs) are established around a common purpose that combine relevant resources located in the collaborating administrative domains. Such Grid systems are exemplified in initiatives for scientific and engineering experimentation such as TeraGrid[1] and the LHC Grid[2]. A crucial requirement for Grid systems, therefore, is the development of security policies and mechanisms that provide a balance between organizational autonomy, partially trusted organizations, common VO goals, and distributed users.

Grid security solutions address several goals with building blocks that comprise cryptographic techniques and protocols. The solutions need to (1) enable sharing and coordination of VO resources located in multiple administrative domains, (2) provide trust between users and resources and (3) support a dynamic environment where VOs add and remove resources and organizations join and leave VOs. A mature and popular security solution that addresses these goals is the Globus Toolkit Version 4 (GT4) Grid Security Infrastructure (GSI) [4, 21]. The toolkit provides several building blocks including secure communication between Grid entities, single sign-on capabilities across multiple resources, and privilege delegation from one Grid entity to another. At the core of these security building blocks are cryptographic techniques and protocols. In particular, the Globus Toolkit uses RSA public key technologies in the form of X.509 digital End-Entity Certificates (EECs) and Proxy Certificates [20],

[1] www.teragrid.org
[2] www.cern.ch/LCG

key generation and management [17, 1], public key encryption and digital signatures, TLS secure messaging [6], and privilege delegation to proxy keys [22].

Recently, alternative identity-based cryptographic (IBC) protocols based on bilinear pairings have been proposed for Grid security [5, 11, 14, 15, 16] with the aim of improving performance and scalability. This body of work argues that the overheads of key generation and exchange as well as signing and verification of existing RSA based cryptographic protocols is too high and limits scalability. To address such scalability limitations they propose alternatives using pairing-based cryptography that offer primitives with better performance. A third cryptosystem that has been gaining attention in the area of performance is Elliptic Curve Cryptography (ECC) [13]. While ECC has not been extensively studied in the context of Grid security, its improved performance owing to factors such as smaller key sizes has motivated its use in distributed computing [10] and its integration with standardized protocols such as TLS [2, 9].

In this work we argue that improved performance on individual properties of cryptosystems (e.g., faster key generation) is not enough to merit re-design of systems such as those in Grid computing. Instead, extensive experimentation must be undertaken on the specific systems and the results analyzed before any conclusions are made. We undertake such experimentation and analysis for Grid systems; in particular, Grid systems for e-science such as TeraGrid that are characterized by high-performance servers and high-bandwidth, high-latency networks. We study three cryptosystems, namely, RSA, IBC and ECC, and measure their computational and communication overheads for supporting Grid security by using the Globus Toolkit as the basis for designing protocols that provide Grid security. Our results indicate that IBC and ECC provide no significant performance improvements over RSA for Grid security. Furthermore, we study both 1024-bit and 2048-bit RSA protocols that aim to provide adequate security until the year 2030[3]. While this conclusion is contrary to the work on IBC for Grid security [14, 15, 16], it is, in fact, aligned intuitively with the high-performance, high-bandwidth and high-latency nature of Grid systems. That is, these systems constitute (1) processors that can execute cryptographic operations in a small amount of time and (2) networks where the delay of sending a packet between two Grid entities matters far more than the number of packets sent at a time.

The rest of this paper is organized as followed. In Section 2 we provide background on GT4 GSI that uses RSA for Grid security as well as ECC and IBC for Grid security. In Section 3 we undertaken a comparison of computational costs. In Section 4 we undertake a comparison of communication costs and we conclude in Section 5.

2 Background

A typical Grid usage scenario involves a user (e.g., scientist) contacting a service to initiate a computational job. The service accepts the job and uses computational and storage resources to execute it. If needed, the service may contact other services to use additional resources governed by them. At the end of the job execution results are put in a storage resource for the scientist to access. In this section we review the core cryptographic protocols that support security functions for such a Grid usage scenario, the GT4 GSI toolkit that implements these protocols with RSA, and outline the design of these protocols with ECC and IBC.

Security support for Grid usage can be provided with two basic cryptographic protocols, namely, mutual authentication and delegation. Mutual authentication allows any two Grid entities such as a user and a service, or, a service and another service, to place trust in each other. GSI uses RSA based TLS for mutual authentication [6]. At the end of this trust establishment process the entities can optionally establish keys for securing (i.e., ensuring confidentiality and integrity of) bulk data exchange. Delegation allows a user to delegate privileges to a service to enable the service to access resources for executing jobs; e.g., to allow a file transfer service to access the requested dataset for a job that involves use of the dataset. Furthermore, delegation allows a service to delegate

[3] http://csrc.nist.gov/CryptoToolkit/tkkeymgmt.html

privileges to other services for access to resources as part of the user's job request; i.e., specifically for the job. In GSI delegation is achieved by the use of Proxy Certificates[4] [20] and a proxy delegation protocol [22]. This involves the delegatee generating a new key pair and signing a certificate request with the new private key, and the delegator signing the request with an existing private key after verifying the request. The same delegation protocol is used by users to obtain proxy certificates from a credential repository (such as MyProxy [1]) that protects users' long-term private keys.

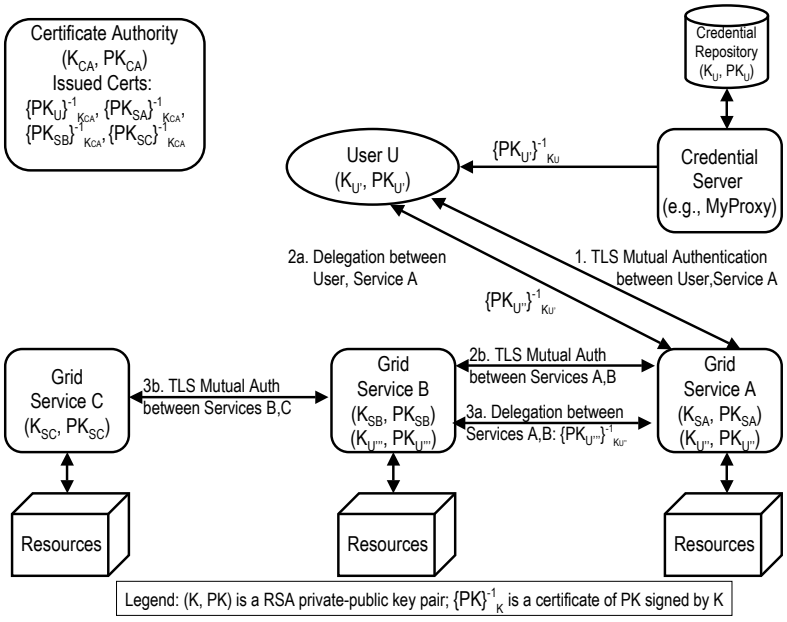

Figure 1. Grid Security with Mutual Authentication and Proxy Delegation

Figure 1 illustrates GSI in action by using the mutual authentication and delegation protocols. Here, user U stores her long-term key pair (K_U, PK_U) in a credential server for enhanced protection of the private key. When U wishes to use Grid resources she generates a proxy key pair $(K_{U'}, PK_{U'})$ and via the delegation protocol obtains a certificate for the public proxy key that is signed by her long-term private key stored at the credential repository [1]. To issue a job to Grid Service A, U executes the mutual authentication protocol with Service A whereby U uses the proxy key along with the proxy certificate as well as her EEC issued by the CA and Service A uses its EEC issued by the CA[5]. In job executions Service A may need to contact Service B for additional resources that are needed to complete the job. In this case, Service A needs to "impersonate" user U in order to obtain service. For this U and Service A execute a delegation protocol that involves Service A generating a new proxy key pair $(K_{U''}, PK_{U''})$ and obtaining U's signature on proxy public key with U's proxy key $K_{U'}$. After this delegation, Service A executes the mutual authentication protocol with Service B whereby Service A uses the new proxy key along with a chain of certificates leading to the CA and Service B uses its EEC issued by the CA. If Service B needs to further contact Service C for additional resources then Service B executes the delegation protocol with Service A using a newly generated proxy key pair $(K_{U'''}, PK_{U'''})$ followed by a mutual authentication with Service C. Similarly, this process can extend to any number of Grid services.

[4]The use of Proxy Certificates also provides single sign-on capabilities in GSI by allowing a user to repeatedly authenticate to multiple services with a private key but without having to re-enter a password on every authentication [22].

[5]Note that these entities are likely to obtain EECs from different CAs but for simplicity we using a single trusted CA in the illustration.

IBC for Grid Security. IBC using bilinear pairings has received a lot of attention after the seminal work by Boneh and Franklin [3] in designing a practical identity-based encryption scheme. Properties of IBC such as the ability to generate public keys from identifiers and short key-sizes have motivated their use in Grid systems [5, 11, 14, 15, 16]. Of these, Lim and Paterson's Identity-based Infrastructure for Grid (IKIG) [14] represents a culmination of research ideas from previous efforts and proposes IBC protocols for Grid security that aim to achieve the same goals as GSI in GT4 with similar assumptions. Furthermore, they also discuss what it might take to implement and integrate their techniques with tools such as TLS. We use this work as the basis of comparison with GSI in GT4. IKIG proposes alternatives to Proxy Certificates, proxy delegation, and mutual authentication using bilinear pairings. We provide an overview of IKIG here highlighting differences with Figure 1 and refer the reader to [14] for details. We note that unlike RSA and ECC the use of IBC for Grid security has not yet been standardized.

In IKIG a Trusted Authority (TA), which is akin to a CA, holds a master secret key that is used to generate and distribute long-term public-private key pairs to Grid entities (users and services). The public key of each entity is derived from its identity; e.g., a X.500 Distinguished Name. A user can further create a "proxy" key from his long-term private key establishing a hierarchy of keys as in Hierarchical Identity Based Encryption (HIBE) and Hierarchical Identity Based Signature (HIBS) schemes [8]. Like GT4 the proxy keys are created when a job needs to be submitted. There are several key differences in IKIG with GT4. For mutual authentication, entities verify a single HIBS signature on the proxy key without needing to verify additional signatures on a certificate chain as required by RSA. Entities need not exchange public key, only their identifiers, as public keys for these identifies can be computed by the entities. For delegation, the delegator can compute the proxy key using the delegatee's identifier and sign a delegation to that key in a single step. On the negative side, the delegator must send the private proxy key over the network while in GT4 the private key is never sent over the network.

ECC for Grid Security. ECC is gaining momentum primarily because of smaller key sizes [9, 10]. We feel that this cryptosystem deserves its place in this comparison for Grid security because it shares the property of smaller key sizes with IBC and has already been integrated into TLS [2]. The mutual authentication and proxy delegation protocols using ECC would be very similar to the ones in GT4 except that they would use ECC keys and certificates[6] and the ECC version of TLS [2].

3 Computational Costs

Our first study involves undertaking a comparison of computational costs of RSA, ECC, and IKIG cryptosystems. The goals of the study are to determine the time it would take two Grid entities to execute mutual authentication and delegation protocols. To do so, we first determine the operations (e.g., generation, encryption, signing) and primitives (e.g., exponentiation, multiplication, pairing) that would be required for these protocols and then measure the time taken to execute these operations on a common processor. Keeping the relationship between security and key sizes in mind, we execute the operations for equivalent key sizes in the three cryptosystems. To minimize the impact of variations in implementation and optimization we use the MIRACL[7] library that provides efficient implementations of all three cryptosystems. We execute these operations on an AMD Opteron 2.2Ghz Processor 248 running Debian Linux, which is representative of the kind of processor found in mid-range Grid systems and user desktops. All operations are averaged over 1,000 iterations.

Equivalent Key Sizes. Keeping the foreseeable future in mind, we undertake a comparison with both 1024 and 2048 bit RSA systems with 2048 bit security being considered secure until the year 2030. Table 1 lists the

[6] X.509 Proxy Certificates, like X.509 End Entity Certificates, can use the ECC algorithms being standardized by the IETF Public Key Infrastructure (X.509) working group.

[7] `http://www.shamus.ie`

equivalent key sizes for ECC [2] and IKIG using supersingular curves in characteristic 2 (F_{2^m}) [18]. For IKIG, 457 bits provides security equivalent to 1828-bit RSA and we use that for comparison.

RSA	**ECC**	**IKIG (F_{2^m})**
1024	163	271
2048	233	457

Table 1. Equivalent Key Sizes in bits

Operations and primitives. In Table 2 we identity the primary operations for mutual authentication and delegation, the primitives that must be executed to complete these operations, and the time taken to complete each operation with the three cryptosystems and varying key sizes. For IKIG protocols we use Tate pairings.

RSA		**512 bits**	**1024 bits**	**2048 bits**	**Operation Description**
GEN	GEN	12.52	85.6	784.5	GEN = RSA parameter generation
ENC, VER	EXP_m	0.07	0.2	0.65	EXP_m = RSA modular exponentiation
DEC, SIG	EXP_m	0.5	2.4	13.1	
ECC		**163 bits**	**233 bits**		
GEN, DH, SIG	1 MUL	1.45	2.90		MUL = Elliptic curve point multiplication
VER	2 MUL	1.91	4.15		
IBC		**271 bits**	**457 bits**		
EXT, SIG	1MUL	1.68	3.5		MUL = Elliptic curve point multiplication
ENC	2 MUL, 1 EXP_f	4.73	9.5		PAI = Pairing computation
DEC	2 PAI	6.85	20.6		EXP_f = Exponentiation in a finite-field
VER	2 PAI, 2 MUL	7.03	20.5		mutiplicative group
ENC, DEC, SIG, VER, GEN denote encryption, decryption, signature, verification and key generation in the corresponding cryptographic setup. DH denotes a Diffie-Helman computation and EXT denotes HIBE/HIBS private key extraction.					

Table 2. Operations, Primitives, and Operation Timings (in ms).

Overall computational costs. In Table 3 we list the operations involved in executing mutual authentication and delegation protocols, as well the time taken to execute the protocols at the client, server, delegator, and delegatee. From Figure 1 we see that as the requirement for additional resources increases, the proxy delegation depth increases. This affects the signature verification on the server side for mutual authentication. This is indicated as depth d in Table 3. In IKIG the use of HIBS provides an advantage in that only part of a signature verification (labeled VER') computation is added for every additional delegation; specifically, 1 pairing and 1 multiplication [8]. In mutual authentication, while signature verification begins with the proxy key, it ends with verification of the CA's self-signed certificate[8] so we include the cost of verifying self-signed CA and TA certificates. Proxy keys are typically short-lived and are therefore usually shorter in length than long-term keys; e.g., 512 bits for RSA in GT4. Keeping this in mind we provide measurements for mixed key lengths except for IBC where the use of HIBE/HIBS requires all keys to be part of a single set of parameters. All operations in the delegation protocols use the shorter key length while operations in TLS mutual authentication protocol were counted separately with each key length after analyzing the TLS protocol and its use in GSI (though for simplicity in the Table we do not show the individual counts). All of these computational measurements assume no pre-computation for RSA key generation, ECC point multiplications, or IBC pairings.

Analysis. A major motivation for the work on IBC for Grid security [5, 11, 16] was to reduce the computational burden on the user, which may be accessing Grid resource from her desktop machine. This motivation came from

[8]Openssl implementation of TLS: `www.openssl.org`

TLS Auth	RSA Operations	RSA 1024/512 Time	RSA 2048/1024 Time	ECC Operations	ECC 163/163 Time	ECC 233/163 Time	IBC Operations	IBE 271/271 Time	IBC 457/457 Time (ms)
TLS Client	2 VER, 1 SIG, 1 ENC	0.97	3.9	2 VER, 1 SIG, 1 DH	6.72	11.2	1 VER, 1 ENC, 1 SIG	13.44	33.5
TLS Server	(2 + d) VER, 1 DEC	0.9 + (d*0.07)	3.7 + (d*0.2)	(2 + d) VER, 1 DH	5.27 + (d*1.91)	11.2 + (d*1.91)	1 DEC, 1 VER, d VER'	13.88 + (d*5.1)	40.9 + (d*13.6)
Proxy Delegation	**RSA Operations**	**RSA 1024/512 Time**	**RSA 2048/1024 Time**	**ECC Operations**	**ECC 163/163 Time**	**ECC 233/163 Time**	**IBC Operations**	**IBE 271/271 Time**	**IBC 457/457 Time**
Delegator	1 VER, 1 SIG	0.57	2.6	1 VER, 1 SIG	3.36	3.36	1 EXT, 1 SIG	3.36	6.9
Delegatee	1 GEN, 1 SIG	13.02	88	1 GEN, 1 SIG	2.9	2.9	N/A	N/A	N/A

Table 3. Measurements (in ms) for Mutual Authentication and Delegation protocol Computations with Delegation Depth d

the assumption that the user's machine is involved in *every* delegation that is a part of the sequence of service calls for a job. Even before analyzing our computational results we note that motivation is not quite correct as the authors' analysis was based on an earlier GSI design [7], which is not used in the current GT4 GSI. As illustrated in Figure 1, in GT4 the user's machine is involved only in two delegations: establishing a proxy at the desktop and delegating to the first service for enabling mutual authentication with the second service. After that, the interaction is limited to services and does not require user intervention. To that end, we argue that the computational overhead at the user's side is reasonable and the overhead on services is a lesser concern as these are typically very high-performance systems.

Moving from overall GSI design to the specific cryptographic protocols in question, we see from Table 3 that both RSA 1024-bit and RSA 2048-bit cryptosystems significantly outperform their ECC and IBC equivalents, except in the case of the delegatee. This is true for any depth d. In case of the delegatee the primary costs involve key generation. For this we argue that pre-generation of RSA keys at services that regularly engage in delegations for supporting Grid computing is a reasonable means of eliminating this cost. In general, pre-computation such as RSA key generation, ECC multiplications, and IBC pairings can reduce computation costs.

4 Communication Costs

Our second study involves undertaking a comparison of networking costs of RSA, ECC and IKIG cryptosystems. The goals of the study are to determine the bandwidth costs, the number of roundtrips, and the time taken for each round trip between two Grid entities to execute mutual authentication and delegation protocols. To estimate bandwidth costs and determine the number of roundtrips we analyze the TLS mutual authentication and proxy delegation protocols. We then measure the actual time taken for each roundtrip on the Teragrid network between servers at NCSA (Urbana, Illinois) and (1) SDSC (San Diego, California), (2) TACC (Austin, Texas), and (3) PSC (Pittsburgh, Pennsylvania). We vary the size of the message on each roundtrip and observe the difference in roundtrip times. These measurements are taken using the *ping* utility and we take the minimum time across 50 runs as a basis of comparison. This allows us to observe the basic networking costs ignoring issues such as jitter.

In Table 4 we provide estimates of bandwidth needs of the three cryptosystems and varying key sizes. Certificate

sizes for RSA were measured using OpenSSL and for ECC were obtained from the ECC/TLS Interoperability Forum[9]. For IKIG estimates were based on Lim and Paterson's IKIG work [14]. The estimates also provide a means to gauge the bandwidth impacts of increasing delegations (Q-values are IKIG parameters). These estimates take two optimizations into account. First, we assume that CA (or TA) certificates are not distributed in TLS as that is optional [6]. Second, the chain of proxy certificates leading to the EEC is assumed to be excluded from the delegation protocol while in practice (i.e., in GT4 GSI) this is typically included. We make this assumption because a TLS mutual authentication typically preceded a delegation and, therefore, the delegatee already has this chain of proxy certificates.

Keys	**Certificate**	**Encryption**	**Signature**	**TLS Components**	**TLS Bandwidth**	**Delegation Components**	**Delegation Bandwidth**
RSA 1024/512	6384/5832	512	512	(2+*d*)Certs, 1Enc, 1Sig	13792 + *d**5900	1Key, 1Sig, 1Cert	7424
RSA 2048/1024	9792/6384	1024	1024	(2+*d*)Certs, 1Enc, 1Sig	21632 + *d**6384	1Key, 1Sig, 1Cert	8432
ECC 163	6400	N/A	326	(2+*d*)Certs, 1Key, 1Sig	13126 + *d**6400	1Key, 1Sig, 1Cert	7424
ECC 233/163	6400/6400	N/A	326	(2+*d*)Certs, 1Key, 1Sig	13126 + *d**6400	1Key, 1Sig, 1Cert	6889
IKIG 271	N/A	1056	816	1Enc, 1Sig, *d*Q-values	1872 + *d**272	1Sig	816
IKIG 457	N/A	1428	1374	1Enc, 1Sig, *d*Q-values	2802 + *d**458	1Sig	1374

Table 4. Bandwidth Estimates (in bits) with Delegation Depth d

Review of the protocols indicates that TLS mutual authentication involves two network roundtrips between the client and the server and that and proxy delegation protocol involves one roundtrip between the delegator and delegatee. To understand the impact of bandwidth requirements on actual networking costs we measured roundtrip times with varying message sizes using the *ping* utility. We take these measurements on the Teragrid network, which connects its sites via a high-speed optical network (10 - 30 gigabits per second) using resources such as the Internet2 Abilene[10] network and the National Lamdba Rail[11]. The minimum time taken for each message size across 50 runs is reported in Table 5 for measurements taken between Urbana, IL and San Diego, CA using the Teragrid network. Measurements for the same message sizes on the Teragrid network between Urbana, Illinois and Pittsburgh, Pennsylvania provide similar results with values ranging between 37.987 ms and 38.876 ms. Measurements taken between Urbana, Illinois and Austix, Texas provide values ranging between 39.184 ms and 39.923 ms.

Analysis. While the estimates in Table 4 indicate that IKIG has signiqficantly lower bandwidth requirements as compared to RSA or ECC, the network measurements in Table 5 indicate that the impact of this reduced requirement on actual network costs is minimal. (A message size of 96,000 bits allows for a delegation depth greater than 10). This is because Grid networks are high-bandwidth, high-latency networks designed for transfer of large amounts of scientific data; therefore, message size differences on the order to 10s of kilobits do not significantly impact performance.

[9] `http://dev.experimentalstuff.com:8082/`

[10] `http://abilene.internet2.edu/`

[11] `http://www.nlr.net/`

Message Size (bits)	Roundtrip Time (ms)	Message Size (bits)	Roundtrip Time (ms)	Message Size (bits)	Roundtrip Time (ms)
1024	67.824	12,000	68.04	60,000	68.575
2048	67.91	24,000	68.119	72,000	68.746
4096	67.91	36,000	68.332	84,000	68.775
8192	68.04	48,000	68.485	96,000	68.839

Table 5. Roundtrip Times on the Teragrid Network between Urbana, IL and San Diego, CA

5 Conclusion

Recently, Identity-Based Cryptography (IBC) has been proposed as an alternative to RSA for supporting Grid security functions. In this work we argue that for Grid systems improved performance on individual operations of cryptosystems are not sufficient for replacement of existing software systems. Instead, extensive experimentation on the actual systems needs to be undertaken. To that end, we have experimentally measured and compared the individual computation and communication costs of mutual authentication and delegation protocols that provide the cornerstone of Grid security with three cryptosystems, namely, RSA, ECC and IBC/IKIG. Our experiments included the 2048 bit RSA system, which is considered secure until the year 2030. Our results indicate that (1) RSA outperforms ECC and IKIG in most operations for computational costs except for key generation and (2) the bandwidth savings of IKIG over ECC or RSA have a minimal impact on actual network roundtrip times. We believe that key generation costs that are part of the delegation protocol can be minimized or even eliminated with key pre-generation. Overall, this leads us to conclude that at least for the time being the choice of RSA for GT4 GSI is reasonable and that the merits of ECC or IKIG are insufficient to motivate a re-design of GSI. Replacement with IKIG would also be compounded by the lack of standards and mature software implementations, which take a lot of time, energy and money.

In general, however, we do not promote RSA over ECC or IBC/IKIG. We believe that all cryptosystems have their strengths and weaknesses and, consequently, system designers must carefully evaluate their needs and decide on a choice prior to development. Replacing existing software systems, however, is quite costly and should be undertaken only in exceptional cases such as discovery of vulnerabilities[12] or significant practical benefits. It is possible that development and use of Grid technologies may create scenarios where alternative cryptosystems can provide significant practical benefits. An example could be the use of Grid technologies via low-powered systems such as handhelds, which are strong drivers behind growth and development of ECC. However, even in that setting one would have to compare the advantages of alternative cryptosystems with emerging Grid portal technologies that offload security functions to the portals as opposed to executing them on the low-powered devices.

References

[1] J. Basney, M. Humphrey, and V. Welch. The MyProxy Online Credential Repository. In Software: Practice and Experience, volume 35(8), 2005.

[2] S. Blake-Wilson, N. Bolyard, V. Gupta, C. Hawk, and B. Moeller. RFC4492: Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS). In RFC4492, Internet Engineering Task Force, May 2006.

[3] D. Boneh and M. K. Franklin. Identity-Based Encryption from the Weil Pairing. Advances in Cryptology - Proceedings of CRYPTO 2001 (2001).

[12] Recently discovered weaknesses in MD5 hash protocols come to mind.

[4] R. Butler, D. Engert, I. Foster, C. Kesselman, S. Tuecke, J. Volmer, and V. Welch. National-scale authentication infrastructure. IEEE Computer, 33(12):60-66, 2002.

[5] L. Chen, H.W. Lim, and W. Mao. User-friendly Grid Security Architecture and Protocols. In, Proceedings of the 13th International Workshop on Security Protocols 2005, Cambridge, UK.

[6] T. Dierks and E. Rescorla. RFC 4346: The Transport Layer Security (TLS) Protocol Version 1.1. In RFC4346, Internet Engineering Task Force, April 2006.

[7] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke. A Security Architecture for Computational Grids. Proc. 5th ACM Conference on Computer and Communications Security Conference, pp. 83-92, 1998.

[8] C. Gentry and A. Silverberg. Hierarchical ID-Based Cryptography. In Proceedings of ASIACRYPT 2002: 548-566.

[9] V. Gupta, S. Gupta and S. Chang. Performance Analysis of Elliptic Curve Cryptography for SSL. ACM Workshop on Wireless Security (WiSe), Mobicom 2002, Atlanta, Sept. 2002.

[10] V. Gupta, M. Millard, S. Fung, Y. Zhu, N. Gura, H. Eberle, S. Chang. Sizzle: A Standards-based end-to-end Security Architecture for the Embedded Internet, Third IEEE International Conference on Pervasive Computing and Communication (PerCom 2005), Kauai, Mar. 2005.

[11] X. Huang, L. Chen, L. Huang, M. Li. An Identity-Based Grid Security Infrastructure Model. ISPEC 2005, pp 314-325.

[12] M. Humphrey, M. Thompson, and K.R. Jackson. Security for Grids. Proceedings of the IEEE (Special Issue on Grid Computing), vol 93, No. 3, March 2005. pp. 644 – 652.

[13] N. Koblitz. Elliptic Curve Cryptosystems. In Mathematics of Computation 48, 1987, pp. 203-209.

[14] H.W. Lim and K.G. Paterson. Identity-Based Cryptography for Grid Security. In, H. Stockinger, R. Buyya, and R. Perrott, editors, Proceedings of the 1st IEEE International Conference on e-Science and Grid Computing (e-Science 2005), Melbourne, Australia, pages 395-404. IEEE Computer Society Press, 2005.

[15] H.W. Lim and M.J.B. Robshaw. A Dynamic Key Infrastructure for GRID. In, P.M.A. Sloot, A.G. Hoekstra, T. Priol, A. Reinefeld, and M. Bubak, editors, Proceedings of the European Grid Conference (EGC 2005), Amsterdam, The Netherlands, pages 255-264. Springer-Verlag LNCS 3470, 2005.

[16] W. Mao. An Identity-based Non-interactive Authentication Framework for Computational Grids. HP Labs Bristol, Trusted Systems Laboratory, Technical Report, HPL-2004-96. 2004.

[17] J. Novotny, S. Tuecke, V. Welch,"An Online Credential Repository for the Grid: MyProxy" 10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10 '01), 2001.

[18] D. Page, N. P. Smart and F. Vercauteren. A comparison of MNT curves and supersingular curves. Appl. Algebra Eng., Commun. Comput. 17, 5, Oct. 2006, 379-392.

[19] F. Siebenlist, N. Nagaratnam, V. Welch, and C. Neuman. Security for virtual organizations. In The Grid2: Blueprint for a New Computing Infrastructure, 2004.

[20] S. Tuecke, V. Welch, D. Engert, L. Perlman, and M. Thompson. RFC3820: Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile. In RFC3820. Internet Engineering Task Force, 2004.

[21] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, and S. Tuecke. Security for grid services. In Twelfth International Symposium on High Performance Distributed Computing (HPDC-12). IEEE Computer Society Press, 2003.

[22] V. Welch, I. Foster, C. Kesselman, O. Mulmo, L. Pearlman, S. Tuecke, J. Gawor, S. Meder, and F. Siebenlist. X.509 Proxy Certificates for dynamic delegation. In Proceedings of the 3rd Annual PKI R&D Workshop, 2004.