2009 IEEE International Conference on Networking, Architecture, and Storage

Identification and Authentication in Large-scale Storage Systems

Zhongying Niu*, Ke Zhou*, Hong Jiang†, Tianming Yang* and Wei Yan*
*School of Computer, Huazhong Uni. of Sci. & Tech.
Wuhan National Laboratory for Optoelectronics, Wuhan, 430074, China
Email: niuzhy@gmail.com, k.zhou@hust.edu.cn, {tmyang, wyan}@smail.hust.edu.cn
†Department of Computer Science and Engineering
University of Nebraska-Lincoln, Lincoln, NE 68588, USA
Email: jiang@cse.unl.edu

***Abstract*—Large-scale storage systems may service millions of clients. Effectively identifying and authenticating these clients can not only prevent illegal accesses but also eliminate unnecessary redundancies of access control. However, existing security solutions have largely ignored the identification and authentication issues or separately consider them from access control mechanisms. As a result, these solutions either have incurred a high cost of access control or led to new insurmountable problems.**

In this paper, we discuss the important consideration in designing an appropriate authentication scheme for large-scale storage systems. First, we discuss potential authentication techniques for large-scale storage systems. The looming crisis of PKI, a widely used technology for authentication in today's information security area is discussed and potential alternative technologies to PKI are suggested. Second, by merging authentication into access control we develop a decentralized security architecture for large-scale storage systems, designed to improve the scalability of authentication and reduce the overhead of access control. Finally, we experimentally evaluate how much overhead can be caused when various cryptographic algorithms used in an authentication protocol are applied to a large-scale storage system.

***Keywords*-authentication; large-scale storage; decentralized security; access control;**

I. Introduction

Large-scale storage systems may service millions of clients. The identification and authentication technologies for such a system must be scalable and provide a simple authentication process because of the large number of clients and concurrent accesses of both random I/O and high data throughput [1]. Effectively identifying and authenticating clients can not only prevent illegal accesses but also eliminate unnecessary redundancies of access control. However, existing security solutions for large-scale storage systems have largely ignored the identification and authentication issues or separately consider them from access control mechanisms. As a result, these solutions either have incurred a high cost of access control or led to new insurmountable problems.

For example, existing security solutions, such as capability based [2], [3], [4], [5], [6], [7], [8], [9], [10], [11] and identity key [12], [13], for large-scale storage systems are strongly tied to a centralized authorization server. Authentication in these systems is centrally performed in the authorization server and completely independent of access control, which are achieved via a tripartite security protocol. To prove a client's identity to the storage device, the authorization server has to grant the client a capability or identity key, which is cryptographically hardened via a complex key hierarchy between the authorization server and storage device. However, the security protocol and key hierarchy are vulnerable to security attacks and incurs additional cost of access control. To authenticate and authorize clients, the authorization server needs to store and maintain an enormous amount of user identities and authentication information; however centralized authentication is not scalable with the ever increasing number of users and goes against the growing trend of global sharing of data across organizational boundaries [14]. For those public storage platforms, such as large-scale Internet services, it is difficult or even impractical to know each client's identifier and store their identities in advance because these clients may come from different organizations and with multiple distinctive identities each.

To address these shortcomings, it is necessary to consider identity management and access control as a whole and thus design a scalable identification and authentication scheme for large-scale storage systems. In this paper, we first discuss three potential certificate-based authentication techniques: public key infrastructure (PKI), identity-based encryption (IBE) and combined public key (CPK). Their advantages and disadvantages in addressing the authentication issue are discussed. Second, by merging authentication into access control we develop a decentralized security architecture for large-scale storage systems, designed to improve the scalability of authentication and reduce the overhead of access control. Finally, we experimentally evaluate the overhead caused by the cryptographic algorithms used in an authentication protocol.

The rest of this paper is organized as follows. Section II compares the three potential authentication techniques: PKI, IBE and CPK. The decentralized security architecture is proposed in Section III. Various cryptographic overhead caused by an authentication protocol is evaluated in Section IV and

978-0-7695-3741-2/09 $25.00 © 2009 IEEE
DOI 10.1109/NAS.2009.72

the conclusions are given in Section V.

II. Potential Solutions

Certificate-based authentication has been widely used in many distributed systems, such as Taos operating system [15], SPKI/SDSI [16], [17] and CRISIS [18], to address the issues of decentralized and cross-domain user authentication. In this section we discuss three potential certificate-based solutions: public key infrastructure (PKI), identity-based encryption (IBE) and combined public key (CPK) [19]. A detailed description and comparison of these solutions will help provide insight into the design decision on the use of an appropriate authentication scheme for large-scale storage systems.

A. Authentication based on PKI

Public key infrastructure (PKI) technology has been available for approximately 30 years and first presents the notion of a third-party authentication and meets the requirement for user identification, authentication, and non-repudiation in cyber security. With PKI, one can maintain her keys and certificates in security and conveniently encrypt data and sign messages. Applications based on PKI include secure web browser, secure e-mail, e-business, e-government, e-banking, and so on. PKI has been imagined to be a magic security elixir, where you can just add a drop to your system and it will become secure. However, there are some counterviews to PKI [20]. Ellison and Scheier [21] consider that the effect of PKI is overtouted, though it can be used as a solution to many security problems. Rivest [22] raises doubts about certificate revocation lists (CRLs), one common approach to revoking certificates. Clarke [23] indicates that the conventional PKI, built around ISO standard X.509 [24], is inherently ineffectual and privacy-invasive. Sha and Bai [25] discuss the deficiencies of PKI and the shortcoming of the current standard based on certificate, such as X.509, PGP [26] and SPKI/SDSI [27], [16], [17]. In sum, there are many inherent deficiencies of PKI and its abilities have been overtouted for a long time.

1) **Illogical trust.** In his article, “Über Sinn und Bedeutung” of 1892, Gottlob Frege[1] (1845-1925), makes a distinction between the actual thing a linguistic expression such as “the morning star” denote or refer to, and the “mode of presentation” or cognitive content associated with the expression in virtue of which the thing is determined or picked out. The former was called the **reference** (*Bedeutung*) of the expression, and the latter was called the **sense** (*Sinn*) of the expression. In the Fregean terminology, a single object can be determined or picked out in virtue of the sense associated with the designation of this object, and the reference of the designation is the object itself. In PKI systems, a certificate binds a public key to a user name, which has been claimed to be the key-holder’s name by a certificate authority (CA). CA is often defined as “trusted”. We usually consider that the reference of “trusted” is CA, but rarely consider the sense of “trusted”, “Who gave the CA the authority to grant such authorizations? Who made it trusted? [21]”. Is it just because PKI vendors tell us “The CA is trusted”?

Many current PKI implementations employ a hierarchical model of trust wherein each layer of CAs needs to be attested to by some superior layer. “Conventional PKI therefore depends on one third party that is partly but not entirely trusted, which in turn depends on another such partly but not entirely trusted third party, which needs to be attested to by some further superior-layer. This results in an unholy spiral up to some mythical authority in which everyone is assumed to have ultimate trust. [23]” As a result, from the superior layer to the inferior trust fast decays down the chain of trust. There is a joke about the trust chain of PKI, “A mother trusts her son, and the son trusts his wife. However, will the mother trust her daughter-in-law?”

2) **Agency expansion and increasing traffic.** The on-line certificate database and hierarchical CA are two basic components of PKI. In fact the ability of a single on-line certificate database is limited and, as a result, a single CA can service only a limited number of users. According to statistics, the number of users that a single CA can service is no more than one thousand [19]. PKI increases the scale of key management by adding CAs, which in turn gives rise to the issue of agency expansion and increasing network traffic.

3) **Unwarranted key revocation.** One common approach to revoking certificates is to issue certificate revocation lists (CRLs). Each such list specifies what unexpired certificates have been revoked, and when the next CRL will be issued. In order to confirm the validity of a certificate, the acceptor must query CAs for corresponding CRLs. As a result, one has to rummage all over the Internet to see if the certificate that she accepted is still OK. But it is not an assured, secure service.

4) **Private key compromise.** Secure digital signature and PKI assume that the holder of a private key will be able to ensure its security. In most cases the secret keys are so long that they can’t be remembered by a normal person, thus the secret keys are usually stored on a host computer under the protection of user password, or in a portable digital memory device in the form of a smart card or truly attack-resistant device. However, with the former scheme the secret keys are subject to attack by viruses and other malicious programs, and with the latter scheme the secret keys can be abused by an infected driving computer.

Since the usage of private keys is relatively small and there are plenty of more attractive attack targets, the private keys have not become a particular target of the attackers so far and there are still very few products available to

[1] A German logician, mathematician and philosopher who played a crucial role in the emergence of modern logic and analytic philosophy.

Table I
A CHARACTERISTIC COMPARISON OF PKI, IBE AND CPK SYSTEMS.

System	*Certificate*	*CA Support*	*Key Management*				
			Generation	*Storage*	*Scale*	*Revocation*	*Protection*
PKI	CA Cert	Yes	Decentralized	On-line certificate database	10^3	CRL	Media, password
IBE	ID Cert	No	Centralized	On-line public parameter server	10^3	Identity	Media, password
CPK	ID Cert	No	Centralized	Build-in chip	10^{48}	Identity	Media, password active parameter

enable users to safeguard their private keys. But as private digital signature keys attract more attention, it is reasonable to expect that more attacks will be made to these keys.

5) **Privacy invasiveness.** Current digital signatures and PKI represent a wide range of privacy-threatening. One can trail a person's e-activity by the logs of CRL, because it has become normal to check the CRL as part of the processing of a transaction. Registration at CAs inevitably exposes sensitive personal data. In order to sufficiently describe the person who holds the digital signature, it requires people to provide detail personal information that they may wish to keep private, such as addresses, date-of-birth etc. Furthermore, the issue of certificates in public may accelerate revealing those information.

B. Authentication based on IBE

In 1984 Adi Shamir [28] first proposed the idea of an identity-based cryptosystem in which the public key can be an arbitrary string. However, the fist practical identity encryption scheme was not introduced until 2001 by Boneh and Franklin [29]. Identity-based encryption (IBE) schemes employ the identification of a user's identity as a public key or derive the public key from the user's identity. According to Frege, in the context of IBE the reference of "identity" is public key, however the sense of "identity" is self-identity. As a result, "Identity-based encryption schemes enable any pair of users to communicate securely and to verify each other's signatures without exchanging private or public keys, without keeping key directories, and without using the services of a third party" (Shamir, 1984). In contrast to PKI, IBE can work without the support of a CA hierarchy, thereby eliminating the increasing demands on bandwidth incurred by constantly expanded CA facilities. Because of these inherent advantages, IBE is considered a potential alternative technology to PKI.

C. Authentication based on CPK

Although IBE addresses the crisis of trust that confounds PKI and can work without the support of a CA hierarchy, IBE systems still rely on a large number of public parameters to define their operations, and a user of an IBE system needs to obtain these public parameters before any IBE operation can be carried out. So IBE systems can not work without an on-line public parameter server (PPS), which provides IBE public parameters and policy information for an IBE private key generator (PKG). As a result, IBE is not a true two-party authentication process (i.e., devoid of a third party such as CA in PKI).

Nan [19] presents a combined public key (CPK) algorithm, one of the IBE variants, that does not require a third-party CA hierarchy. But unlike IBE, CPK holds only a small number of public parameters, which can be stored in a tiny chip. Thus CPK will be able to obtain the necessary public parameters for a public key from the chip without an on-line PPS. As a result, CPK fully implements peer-to-peer authentication.

The main idea behind CPK is to use a small amount of seeds to produce an almost limitless amount of keys in order to meet the almost limitless demand for keys. The CPK algorithm is based on the discrete logarithm problem (DLP). It constructs public and private key matrices according to the hardness of DLP, and maps the identity of an entity onto the sequence of the row and column coordinate in these matrices. According to the sequence, the CPK algorithm picks out and combines the matrix elements to generate an enormous number of public/private key pairs. For example, a 32×32 matrix with 192-bit keys that occupies 24KB of memory can generate $32^{32} = 10^{48}$ keys. Thus the size of a public key matrix is small enough to be stored in any accessible media, such as ATM and POS machine, or even be distributed to a user in a smart card. Moreover, CPK first introduces agent and active parameter technologies into key protection in order to withstand the colluded attacks. Besides the protection of traditional physical media and password, the user's private key in a chip is also under the protection of the active parameters.

D. Comparison of PKI, IBE and CPK systems

Table I summarizes the main characteristics and performance properties of key management provided by PKI, IBE and CPK systems. In the table, the *Certificate* column describes the type of certificate used by these authentication systems. CA Cert and ID Cert stand for the authority certificates issued by a certificate authority (CA) and the identity certificates issued by a trusted authority (TA) respectively. The *CA Support* column indicates whether the authentication process needs a third party CA support. The *Key Management* column includes five sub-columns: *Generation*, *Storage*, *Scale*, *Revocation*, and *Protection*, which stand for key generation mode, public key storage, the size that a CA

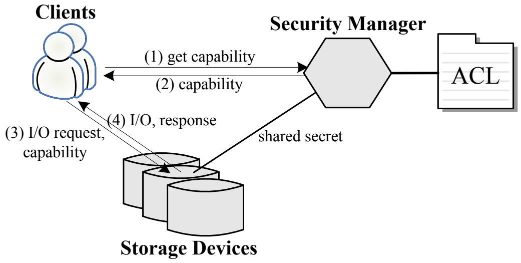

Figure 1. Capability-based security architecture for large-scale storage systems. Clients request capabilities from the authorization server and use them to request I/O from storage devices.

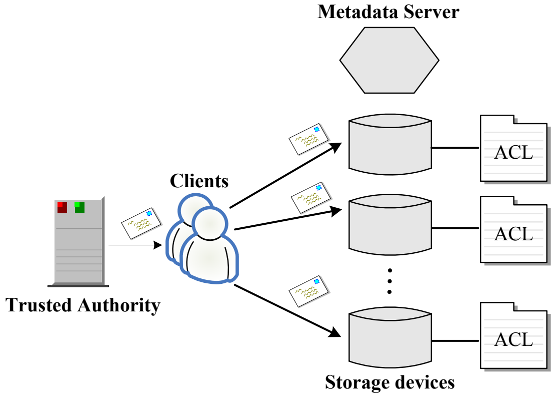

Figure 2. Decentralized security architecture for large-scale storage systms. Storage devices authenticate and authorize clients according to the clients' certificates and local privileges.

or TA can service, key revocation method, and private key protection, respectively. PKI systems revoke any public key certificate by issuing CRLs, while IBE and CPK systems directly revoke a user's identity. It can be seen that CPK addresses the problems that PKI and IBE can not solve.

III. A Decentralized Security Architecture for Large-scale Storage Systems

Recent advances [30], [31], [32], [33], [34], [35], [36] in large-scale storage systems have enabled direct interaction between clients and devices to improve the performance and scalability of the system. However, such a decoupled design of separating metadata from data results in having no implicit knowledge of access privileges and authorizations at the storage devices because the information is now stored at the MDS or security manager. Before accessing the devices, the clients must acquire a capability from the MDS or security manager. A typical security solution based on this architecture is capability-based security shown in Figure 1. Capability-based security solutions have aimed to rapidly authenticate and authorize I/Os but leave user authentication to an existing security infrastructure, such as Kerberos [37]. However, authenticating users and then authorizing I/Os in a centralized authorization server imposes a lot of overhead on the security manager, which also presents a single point of failure and an attractive attack target. If the security manager is down or is subject to denial of service (DoS) attack, the entire system can come to a halt.

A. System design

In this section, we propose a decentralized security architecture shown in Figure 2 for large-scale storage systems, which implements decentralized authentication and authorization by merging existing authentication systems into access control. As the figure shows, in a decentralized security system, each user and component (e.g., the storage device and metadata server) has a unique identifier ID_{id}. These identifers include the designation of the entity as a user or storage device; e.g. $ID_u = (user||identifier)$. User identifiers are certified at regular periods (say monthly) by a TA (trusted authority). A certificate consists of the necessary certificate discriminator ID_{cert}, user identifier ID_u, user public key generated using one of the classic algorithms like RSA, expiration time and other optional items. Thus the certificate can be implemented using identity (ID) cards and is compliant with existing ID card systems. The storage devices authenticate users and authorize requests according to the user's certificate and ACLs stored at the storage devices.

In a decentralized security system, users can directly interact with any device in the network by using a single identity certificate without the service of the centralized security server, thus shortening the data access path and reducing the load on the security manager. Since the user has to prove her identity by a certificate (such as a smart card or passport), which is secured against physical or electronic forgery attempts and the storage devices, that is, data providers themselves determine the users identity and specify what the user is allowed to do by the locally stored privilege information, the security of decentralized security systems has been significantly improved over that of capability-based storage systems that use capabilities to deliver privilege information.

B. Implementation consideration

The trusted authority in a decentralized security system can be the existing federated or enterprise certificate authority, and thus the system can be applicable to most of storage platforms, such as large-scale Internet services, large-scale enterprise data center and federated storage platforms for millions of disjoint, individual customers.

As a rule, the public key cryptography is orders of magnitude slower than the shared key cryptography. To reduce the cost of authentication for each I/O, the system can

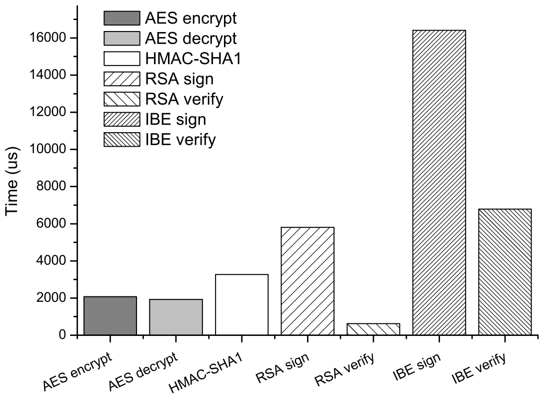

Figure 3. Performance of cryptographic algorithms. Block size is 64KB except for sign & verify, which are done on 160 bits input.

Table II
RAW SPEED OF CRYPTOGRAPHIC OPERATIONS.

	RSA	*IBE*
Hash	-	$11\mu s$
Scalar multiplication	-	$7794\mu s$
Exponentiation	$4675\mu s$	$795\mu s$
paring	-	$6781\mu s$

establish a secret session key between the client and storage device in advance. Then the two parties in a conversation can verify each other's identity by the session key, which can be created in the logging process using an ID-based authenticated key agreement protocol.

Decentralized access control systems demand that the storage devices themselves store access control lists. Keeping ACLs consistent among multiple devices is a critical concern for designing a decentralized security system. There can be many ways to ensure consistency. One popular and efficient way is to store a global replica of the ACLs at a security manager and then all ACL changes will first go to the security manager (master) and then the master will propagate it to other devices (slaves).

IV. CRYPTOGRAPHIC OVERHEAD

We tested the raw speed of the various cryptographic algorithms used by our system on a Linux (kernel version 2.6.12) host, each with one Intel Xeon 3.0 GHz processor and a total of 512 MB DDR-SDRAM physical memory. These algorithms include traditional symmetrical and non-symmetrical cryptographic algorithms as well as up-to-date identity-based cryptographic algorithms; this provides insight into how fast the identity-based algorithms are likely to be when compared to the traditional cryptographic algorithms and how much overhead can be caused when these algorithms are applied to a practical system.

Due to proprietary reasons, we can not obtain an open source library for the CPK algorithm. We used the Hess's identity based signature scheme [38] for the performance evaluation of identity-based signature algorithms, which are implemented using the PBC library [39], and used the OpenSSL crypto library for the AES, HMAC-SHA1 and RSA cryptographic algorithm implementations. The AES and HMAC-SHA1 keys are 128 and 160 bits length respectively, while the RSA and IBE keys are 1024 and 160 bits length respectively. The performance of these cryptographic algorithms is summarized in Figure 3. As the figure shows, the AES and HMAC-SHA1 algorithms with an input of 64KB requires about $2.0ms$ and $3.3ms$ respectively, while with an input of 512 bits, roughly the length of a command, the AES and HMAC-SHA1 algorithms, which provide the confidentiality and integrity protection to messages and commands in an authentication system, incur latencies of only $17\mu s$ and $14\mu s$ respectively.

The most expensive operation by far is signature generation, which takes about $5.8ms$ and $16.4ms$ in the RSA and IBE cryptographies respectively. Compared to signature generation, signature verification is cheaper, which costs about $0.6ms$ and $6.8ms$ in the RSA and IBE cryptographies respectively. For further analysis, we tested the raw speed of cryptographic operations in RSA and IBE signature algorithms. As shown in Table II, the scalar multiplication, exponentiation and paring computation are computationally expensive operations. However, for the RSA algorithm only one exponentiation is required in the signing and verifying steps respectively. Without any precomputation[2], the signing operation in the IBE algorithm requires one exponentiation, one hash function evaluation, one paring computation and two scalar multiplication, while the verifying operation requires one exponentiation, one hash function evaluation and two paring computation. With precomputation, the signing and verifying operations can be optimized; thus the former requires one exponentiation, one hash function evaluation and two scalar multiplication, while the latter requires one paring computation and one hash function evaluation. It should be noted that though the signature takes the most amount of running time, it usually occurs only in the logging process.

V. CONCLUSIONS

This paper discusses the important consideration in designing an appropriate authentication scheme for large-scale storage systems. Three potential authentication techniques: PKI, IBE and CPK are discussed. In sum, IBE addresses the crisis of trust that confounds PKI and is considered a potential alternative technology to PKI, whereas CPK as one of the IBE variants can further solve the issue of large-scale key management. We also develop a decentralized security architecture for large-scale storage systems by merging authentication into access controls, designed to improve the scalability of authentication and reduce the overhead

[2] See Section 2 in [38].

of access control. In a decentralized security system, users can directly interact with any device in the network by using a single identity certificate without the service of the centralized security server, thus shortening the data access path and reducing the load on the security manager. Evaluation on various cryptographic overheads shows that with existing cryptographic techniques achieving a rapid I/O authentication has to resort to a symmetrical cryptographic algorithm. Though IBE is considered a potential alternative technology to PKI, as an up-to-date cryptographic algorithm, IBE needs a more compact design and efficient implementation.

Acknowledgments

This work was supported by National Basic Research Program of China (973 Program) under Grant No.2004CB318201, 863 project 2008AA01A401, National Science Foundation of China under Grant No.60703046 and No.60873028, US National Science Foundation (NSF) under Grant No. CCF-0621526, Changjiang innovative group of Education of China No. IRT0725, the Program for New Century Excellent Talents in University NCET-04-0693 and NCET-06-0650, and Wuhan Project 200750730307.

References

[1] F. Wang, Q. Xin, B. Hong, S. A. Brandt, E. L. Miller, and D. D. E. Long, "File system workload analysis for large scale scientific computing applications," in *Proc. of the Conference on Mass Storage Systems and Technologies*, Apr. 2004.

[2] M. K. Aguilera, M. Ji, M. Lillibridge, J. MacCormick, E. Oertli, D. Andersen, M. Burrows, T. Mann, and C. A. Thekkath, "Block-level security for network-attached disks," in *Proc. of FAST '03*, 2003.

[3] A. Azagury, R. Canetti, M. Factor, S. Halevi, E. Henis, D. Naor, N. Rinetzky, O. Rodeh, and J. Satran, "A two layered approach for securing an object store network," in *Proc of IEEE Security in Storage Workshop*, 2002.

[4] M. Factor, D. Nagle, D. Naor, E. Riedel, and J. Satran, "The OSD security protocol," in *Proc. of 3rd IEEE Security in Storage Workshop*, 2005.

[5] Z. Niu, K. Zhou, D. Feng, H. Jiang, F. Wang, H. Chai, W. Xiao, and C. Li, "Implementing and evaluating security controls for an object-based storage system," in *Proc. of MSST'07*, Sep. 2007.

[6] H. Gobioff, "Security for a high performance commodity storage subsystem," Ph.D. dissertation, Carnegie Mellon University, July 1999.

[7] R. A. Oldfield, A. B. Maccabe, S. Arunagiri, T. Kordenbrock, R. Riesen, L. Ward, and P. Widener, "Lightweight I/O for scientific applications," Sandia National Lab, Tech. Rep. 2006-3057, May 2006.

[8] Y. Zhu and Y. Hu, "Snare: A strong security scheme for network-attached storage," in *Proc. of the 22nd Symp. on Reliable Distributed Systems*, 2003.

[9] C. A. Olson and E. L. Miller, "Secure capabilities for a petabyte-scale object-based distributed file system," in *Proc. of the 1st ACM Workshop on Storage Security and Survivability*, Nov. 2005.

[10] A. W. Leung and E. L. Miller, "Scalable security for large, high performance storage systems," in *Proc. of the 2nd Workshop on Storage Security and Survivability*, 2006.

[11] A. W. Leung, E. L. Miller, and S. Jones, "Scalable security for petascale parallel file systems," in *Proc. of SC07*, Nov. 2007.

[12] B. C. Reed, E. G. Chron, R. C. Burns, and D. D. E. Long, "Authenticating network-attached storage," in *Proc. of Hot Interconnects VII*, Aug. 1999.

[13] V. Kher and Y. Kim, "Decentralized authentication mechanisms for object-based storage devices," in *Proc. of the Second IEEE International Security In Storage Workshop*, 2003.

[14] S. Miltchev, J. M. Smith, V. Prevelakis, A. Keromytis, and S. Ioannidis, "Decentralized access control in distributed file systems," *ACM Computing Surveys (CSUR)*, vol. 40, no. 10, August 2008.

[15] E. P. Wobber, M. Abadi, M. Burrows, and B. Lampson, "Authentication in the Taos operating system," *ACM Transactions on Computer Systems*, vol. 12, no. 1, pp. 3–32, 1994.

[16] C. Ellison, B. Frantz, R. Rivest, B. Thomas, and T. Ylonen, *SPKI Certificate Theory*, RFC 2693, IETF, September 1999.

[17] R. L. Rivest and B. Lampson, "SDSI-a simple distributed security infrastructure," September 1996, http://people.csail.mit.edu/rivest/sdsi10.html.

[18] E. Belani, A. Vahdat, T. Anderson, and M. Dahlin, "The CRISIS wide area security architecture," in *Proceedings of the 7th USENIX Security Symposium*, San Antonio, Texas, January 1998.

[19] X. Nan, *Identity Authentication Based on CPK*, 1st ed. Beijing, China: National Defense Industry Press, January 2006, (in Chinese).

[20] T. Moreau, "Thirteen reasons to say 'no' to pubilic key cryptography," CONNOTECH Experts-conseils, Inc., http://www.connotech.com/13REAS.HTM, Draft paper, March 1998.

[21] C. Ellison and B. Schneier, "Ten risks of PKI: what you're not being told about public key infrastructure," *Computer Security Journal*, vol. 16, no. 1, pp. 1–7, Winter 2000.

[22] R. L. Rivest, "Can we eliminate certificate revocations lists?" *Lecture Notes in Computer Science*, vol. 1465, p. 178, 1998.

[23] R. Clarke, "Conventional public key infrastructure: An artefact ill-fitted to the needs of the information society," November 2000, http://www.anu.edu.au/people/Roger.Clarke/II/PKIMisFit.html.

[24] R.Housley, W.Ford, W.Polk, and D.Solo, *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*, RFC 2459, January 1999.

[25] Y. Sha and S. Bai, "On the research and analysis of the main problem of PKI," *Microelectronics & Computer*, no. 6, pp. 18–21, 2002, (in Chinese).

[26] S. Garfinkel, *PGP: Pretty Good Privacy*. O'Reilly & Associates, 1995.

[27] C. Ellison, *SPKI Requirements*, RFC 2692, IETF, September 1999.

[28] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. of CRYPTO 84 on Advances in cryptology*, 1985.

[29] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *Lecture Notes in Computer Science*, 2001.

[30] P. J. Braam, "The Lustre storage architecture," *http://www.lustre.org/documentation.html, Cluster File Systems, Inc.*, Aug. 2004.

[31] S. Ghemawat, H. Gobioff, and S. Leung, "The Google file system," in *Proc. of SOSP'03*, Oct. 2003.

[32] G. A. Gibson, D. F. Nagle, K. Amiri, J. Butler, F. W. Chang, H. Gobioff, C. Hardin, E. Riedel, D. Rochberg, and J. Zelenka, "A cost-effective, high-bandwidth storage architecture," in *Proc. of 8th ASPLOS*, Oct. 1998.

[33] D. Nagle, D. Serenyi, and A. Matthews, "The Panasas activescale storage cluster-delivering scalable high bandwidth storage," in *Proceedings of the ACM/IEEE SC2004 Conference*, Nov. 2004.

[34] O. Rodeh and A. Teperman, "A scalable distributed file system using object disks," in *Proc. of Mass Storage Systems and Technologies Conf.*, 2003.

[35] F. Schmuck and R. Haskin, "GPFS: A shared-disk file system for large computing clusters," in *Proc. of FAST'02*, Jan. 2002.

[36] S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn, "Ceph: A scalable, high-performance distributed file system," in *Proc. of OSDI '06*, 2006.

[37] B. C. Neumann, J. G. Steiner, and J. I. Schiller, "Kerberos: An authentication service for open network systems," in *Proc. of Winter USENIX Conference*, 1988.

[38] F. Hess, "Efficient identity based signature schemes based on pairings," *SAC 2002, LNCS 2595, pp. 310-324, 2003.*

[39] B. Lynn, *PBC Library Version 0.4.12*, http://crypto.stanford.edu/pbc/.