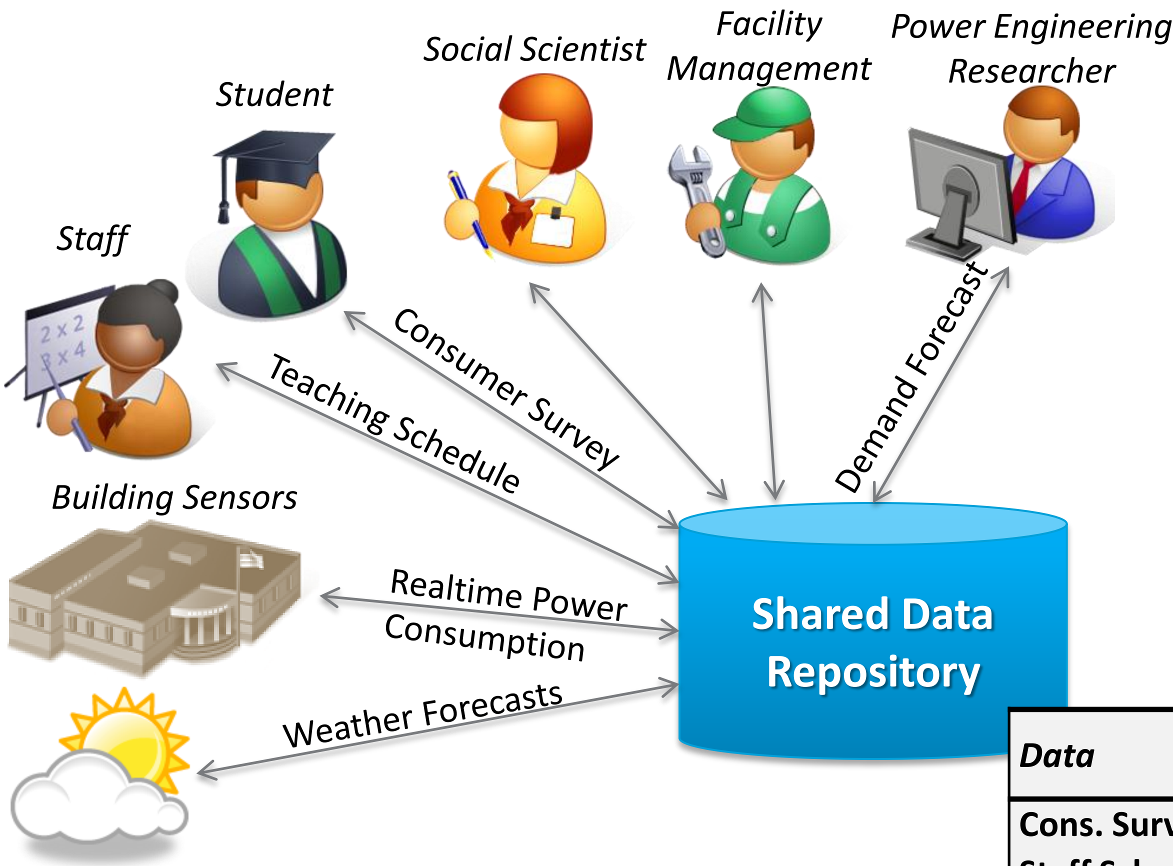
Designing a *Secure Storage Repository* for Sharing Scientific Datasets using Public Clouds

Alok Kumbhare,
Yogesh Simmhan & Viktor Prasanna
Computer Science Department
***University of Southern California**, Los Angeles*

Introduction

- Data sharing - key tenet of scientific computing.
- Tremendous increase in data producers and consumers (e.g. human/electronic sensors)
- Repositories on Cloud provide viable solution but expands potential for **data leakage.**
- Varying degrees of restrictions over data access (e.g. HIPAA)
- Ability to provide Data Owners with **verifiable control** over their data is important.

USC Campus Electricity Micro-Grid



Data	Social Sci.	Power Engr.	Facility Mgmt.	Univ. Staff	Extern
Cons. Survey	O	R*	-	-	-
Staff Schedule	R*	R*	-	O	-
Consumption	R	R	O	-	-
Weather	R	R	O	R	W
Campus Power Forecast	R	W	O	R	-

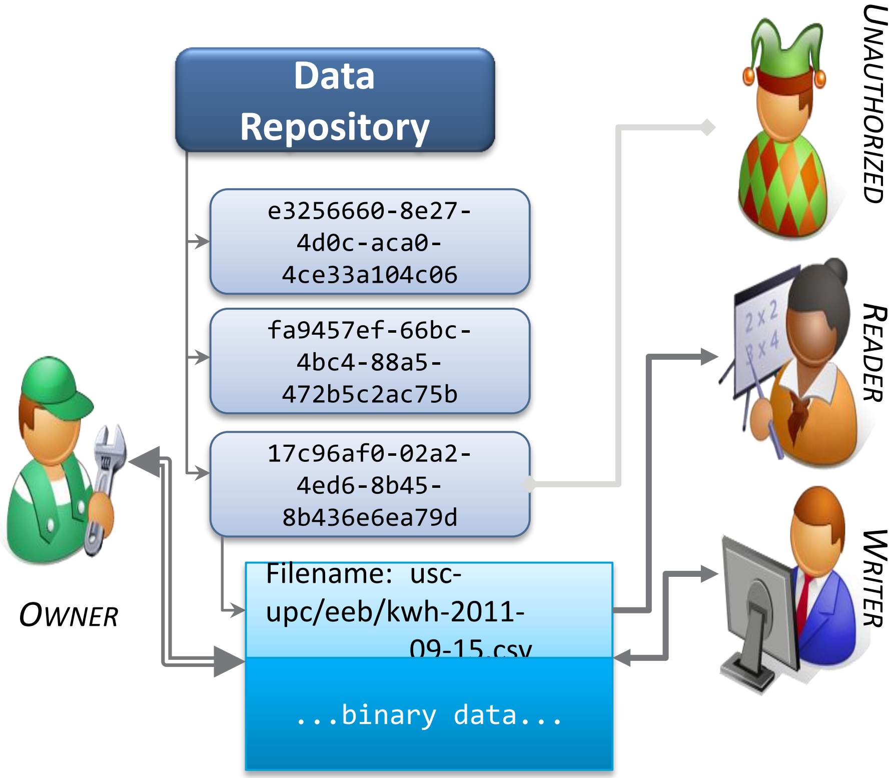
Contributions

- Characterize security and privacy **requirements** for data storage and sharing, using the smart power grid domain as motivation.
- **Cryptonite:** An integrated system designed for a shared, secure Data Repository on Cloud.

Security Requirements

- Data storage security
- Metadata storage security
- Owner controlled data sharing
- Data Integrity and Audit
- Masking ACL & Access Patterns
- Secure Search

Entities and Roles

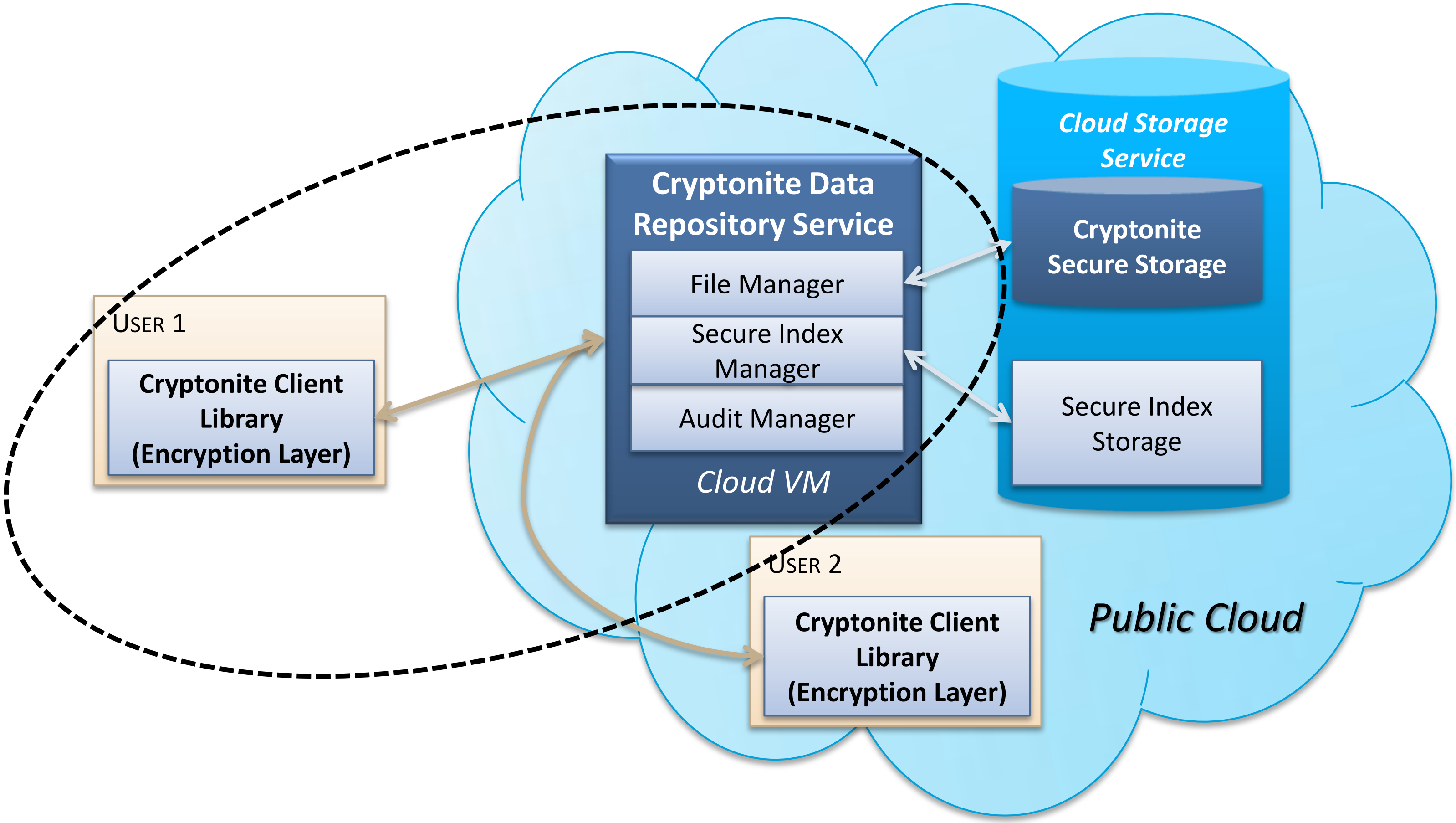


- Users
  - A community of users who require a secure storage repository for data storage and sharing.
- Cloud Storage Service Provider
  - Provides the required persistent scalable storage space.
  - Trusted with 'availability' (SLA)
  - Not trusted with data security
- Secure Data Repository
  - Shared data repository – ***Cryptonite***
  - Not trusted with plain text data
  - Partially trusted to perform requested operations (all operations should be verifiable)

Supported Operations

- PUT
  - Create/Update a file in the repository.
- GET
  - Retrieve an encrypted file
- GRANT(F, U, A)
  - Grant specific access permission (A) to a specific user (U) for a specific file (F)
- REVOKE(F,U,[A])
  - Revoke all/specific access permissions (A) from a user (U) for the given file (F)
- SEARCH
  - Search for files in the repository satisfying a specific query based on the files meta-data properties (e.g. filename, keywords, description etc.)

Cryptonite Architecture



Cryptographic Techniques - I

- Public Key Infrastructure(PKI)
- Digital Signatures
- Broadcast Encryption
  - allows a user to encrypt their data such that it can be deci $K_{encr}^{shared} = f(K_{U_1}^{pub}, K_{U_2}^{pub}, ...)$ lar subset of users.

$$F = encrypt(D, K_{encr}^{shared})$$

$$D = decrypt(F, K_{U_i}^{pri})$$

Cryptographic Techniques - II

- Lazy Revocation
  - A strategy of ***read*** access revocation, in which a file is not re-encrypted unless the file's contents change.
  - ***Key Rotation*** is used to enable forward Secrecy
- Searchable Encryption
  - Allows a user to search within an encrypted file given an appropriate “TrapDoor”.
  - Without decrypting the entire file
  - Without revealing its contents to the searching entity
  - Current SE techniques lack ability to have fine grained access control over index entries and revocation strategies.

Data Structures

$S_M = \mathit{sign}(M,\ K^{pri}_{owner})$

$M = \langle UUID,\ SUUID,\ K^{pub}_{sign},\ U_{owner}\rangle$

$S_F = \mathit{sign}(F,\ K^{pri}_{sign})$

$F = \mathit{encrypt}(D,\ K^{sym}_{encr})$

Data File

PKI & Digital Signature

Searchable Encryption

$S_I = \mathit{sign}(I,\ K^{pri}_{owner})$

$I = \mathit{searchable\text{-}encrypt}(\langle P \rightarrow UUID\rangle, K^{shared}_{searchable})$

Index Entry (Encrypted File Metadata)

$\mathit{sign}(SB,\ K^{pri}_{owner})$

SB = IV-StrongBox | U_{owner}

$\mathit{broad\text{-}encrypt}(K^{pri}_{sign},\ K^{shared}_{w\text{-}encr})$

$\mathit{broad\text{-}encrypt}(K^{sym}_{encr},\ K^{shared}_{r\text{-}encr})$

$\mathit{encrypt}(L,\ K^{sym}_{encr})$

L =

File Handle	Init. Vector
UUID-1	IV-1
UUID-2	IV-2
…	

Key sharing with Broadcast encryption

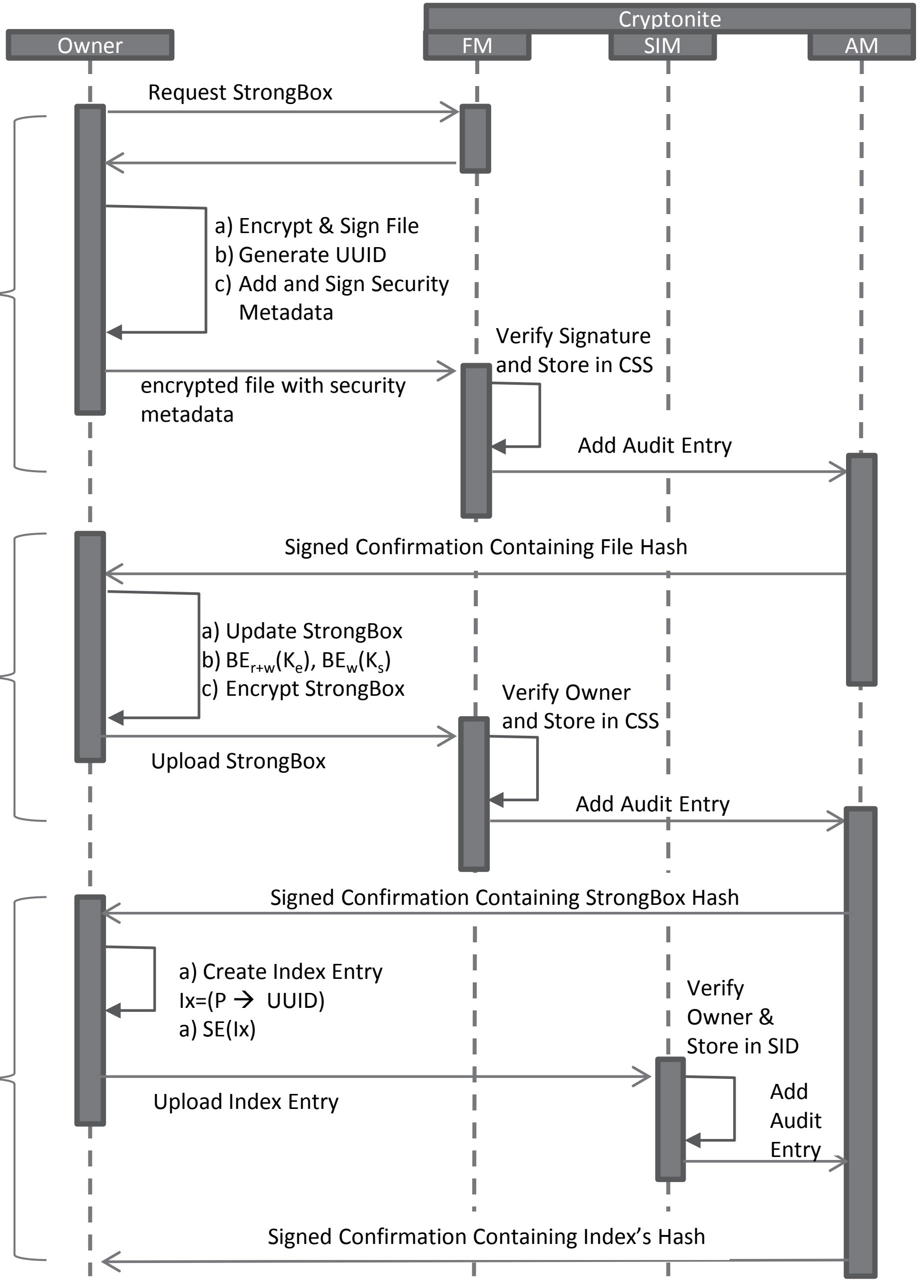
Strong Box

PUT Operation

Data File encryption & Storage.

Update StrongBox. Share File Encryption & Signature Keys using Broadcast Encryption

Encrypt file Metadata (e.g. filname, keywords) using searchable encryption and add to the filesystem index.

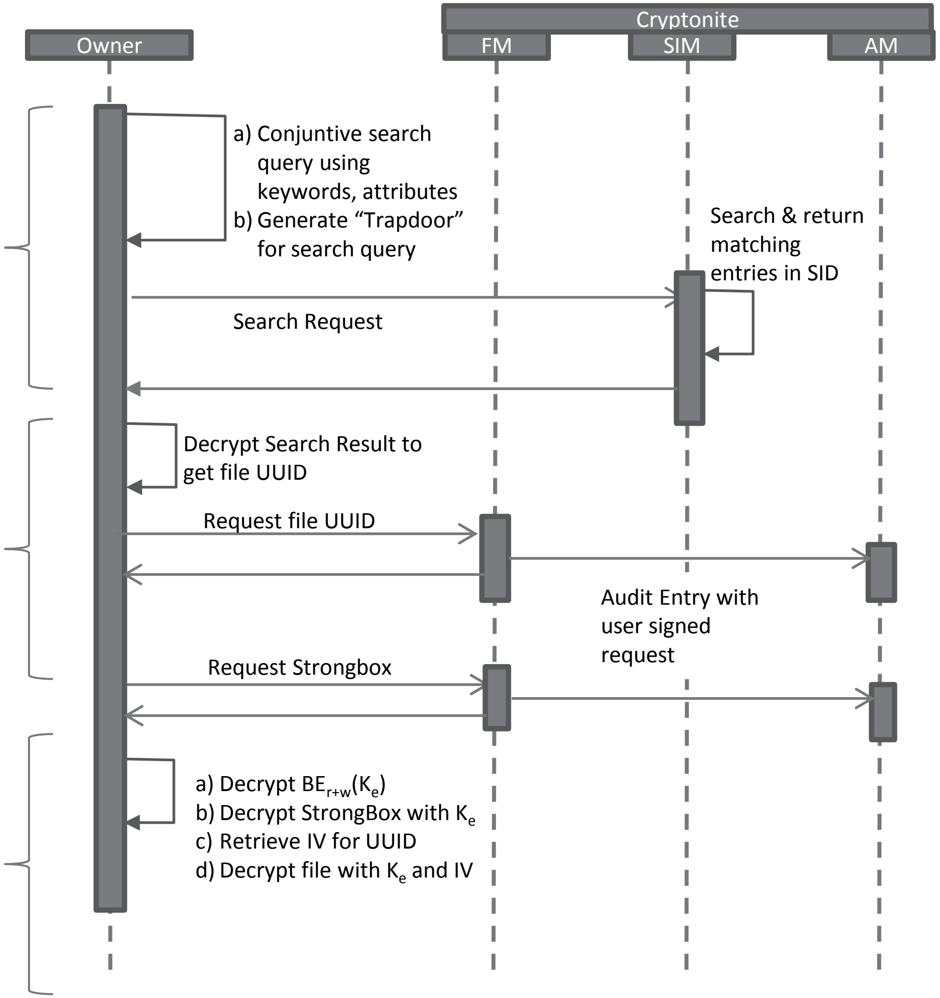


Search & GET

Generate "Trapdoor" for search and decrypt search results.

Retrieve file UUID and download

Retrieve Strongbox, obtain file encryption key & IV.
Decrypt the data file.



GRANT & REVOKE

- Changing access permission for a single file
  - Move that file to the corresponding StrongBox and re-encrypt using that StrongBox's file encryption key.
- Removing a user from the StrongBox's group
  - Use Lazy Revocation for removing user with only read access
  - Use Key Rotation to generate new Encryption key and Signature Key pairs
  - Re-encrypt the file whenever some authorized user updates the file.

Discussion

- Cryptonite protects data condentiality by performing client side encryption before data is stored in the repository.
- “Trust but Verify”: Signed acknowledgements let the end user prove unauthorised updates to his data.
- SSUID of Strongbox in plaintext

Related Work

- Commercial Tools
  - Microsoft Azure Storage, Amazon AWS/S3
    - Access Controlled by the providers
    - Providers have enough information to **decrypt the stored data.**
  - Nasuni Cloud Storage
    - Use cloud as storage backend.
    - More user control. But data **sharing granularity** is limited or requires **out-of-band key** exchange.

Related Work

- Secure data storage in Distributed System
  - SiRiUS[14], PLUTUS[17] etc.
  - **Higher level of trust** on the Storage provider
- Data sharing through public Clouds
  - Cryptographic Cloud Storage[18], Cloud-Proof[24]
  - **Lack of file management** capabilities such as Secure Searchable Encryption.

Future work

- Current BE techniques lack support for random access within an encrypted file.
- Write Serialization, Locking mechanism, Random file access to be addressed in future work.
- Next Step: Implementation and Deployment for USC microgrid Smart Grid initiative.

- Thank You!

kumbhare@usc.edu

References

[1] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous ibe,and extensions. In V. Shoup, editor, Advances in Cryptology - CRYPTO 2005, volume 3621 of Lecture Notes in Computer Science, pages 205{222. 2005.}

[2] S. Aman, Y. Simmhan, and V. K. Prasanna. Improving energy use forecast for campus micro-grids using indirect indicators. In International Workshop on Domain Driven Data Mining (DDDM), 2011.

[3] N. Attrapadung, K. Kobara, and H. Imai. Broadcast encryption with short keys and transmissions. In ACM Digital Rights Management Workshop, 2003.

[4] M. Backes, C. Cachin, and A. Oprea. Lazy revocation in cryptographic _le systems. Security in Storage Workshop, International IEEE, 0:1{11, 2005.

[5] J. Baek, R. Safavi-naini, and W. Susilo. Public key encryption with keyword search revisited. In Computational Science and Its Applications, 2008.

[6] F. Bao, R. H. Deng, X. Ding, and Y. Yang. Private query on encrypted data in multi-user settings. In Information Security Practice and Experience, 2008.

[7] C. Cachin, I. Keidar, and A. Shraer. Trusting the cloud. SIGACT News, 40:81{86, June 2009.

[8] Cloud Security Alliance. Security Guidance for Critical Areas of Focus in Cloud Computing V2.1.
Technical report, December 2009.

[9] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky. Searchable symmetric encryption:improved de_nitions and e_cient constructions. In Computer and Communications Security, 2006.

[10] A. Fiat and M. Naor. Broadcast encryption. In D. Stinson, editor, Advances in Cryptology, volume
773 of Lecture Notes in Computer Science, pages 480{491. Springer Berlin / Heidelberg, 1994.

[11] W. E. Freeman and E. L. Miller. Design for a decentralized security system for network attached
storage. In IEEE Symposium on Mass Storage Systems, 2000.

[12] J. A. Garay, J. Staddon, and A. Wool. Long-lived broadcast encryption. In International Cryptology Conference on Advances in Cryptology, pages 333{352, 2000.}

[13] E.-J. Goh. Secure indexes. 2003.

[14] E.-J. Goh, H. Shacham, N. Modadugu, and D. Boneh. SiRiUS: Securing remote untrusted storage. In
Network and Distributed System Security Conference (NDSS), 2003.

[15] D. Halevy and A. Shamir. The lsd broadcast encryption scheme. In International Crytology Conference, pages 47{60, 2002.

[16] B. S. K. Jr. Rsa digital signature scheme. 2005.

[17] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu. Plutus: Scalable secure _le sharing on
untrusted storage. In USENIX Conference on File and Storage Technologies, 2003.

References

[18] S. Kamara and K. Lauter. Cryptographic cloud storage. In Financial Cryptograpy and Data Security Conference, 2010.

[19] U. M. Maurer. Modelling a public-key infrastructure. In European Symposium on Research in Computer Security, 1996.

[20] E. L. Miller, D. D. E. Long, W. E. Freeman, and B. C Reed. Strong security for distributed _le systems. In IEEE International Performance, Computing and Communications Conference (IPCCC), 2001.

[21] P. Mohan, V. N. Padmanabhan, and R. Ramjee. Nericell: rich monitoring of road and tra_c conditions using mobile smartphones. In Conference On Embedded Networked Sensor Systems, 2008.

[22] M. Newborough and P. Augood. Demand-side management opportunities for the uk domestic sector. Generation, Transmission and Distribution, IEE Proceedings-, 146(3):283 {293, may 1999.

[23] B. Plale, J. Alameda, B. Wilhelmson, D. Gannon, S. Hampton, A. Rossi, and K. Droegemeier. Active management of scienti_c data. IEEE Internet Computing, 9:27{34, 2005.

[24] R. A. Popa, J. R. Lorch, D. Molnar, H. J. Wang, and L. Zhuang. Enabling security in cloud storage slas with cloudproof. Technical report, Microsoft Research, May 2010.

[25] E. L. Quinn. Smart metering and privacy: Existing laws and competing policies. SSRN eLibrary, 2009.
[26] R. L. Rivest, K. Fu, and K. E. Fu. Group sharing and random access in cryptographic storage _le systems. Technical report, Master_ Ss thesis, MIT, 1999.

[27] Y. Simmhan, S. Aman, B. Cao, M. Giakkoupis, A. Kumbhare, Q. Zhou, D. Paul, C. Fern, A. Sharma, and V. Prasanna. An informatics approach to demand response optimization in smart grids. Technical report, Computer Science Dept., Univ. of Southern California, 2011.

[28] Y. Simmhan, A. Kumbhare, B. Cao, and V. K. Prasanna. An analysis of security and privacy issues in smart grid software architectures on clouds. In International Cloud Computing Conference (CLOUD). IEEE, 2011.

[29] Y. Simmhan, V. Prasanna, S. Aman, S. Natarajan, W. Yin, and Q. Zhou. Towards data-driven demand-response optimization in a campus microgrid. In ACM Workshop On Embedded Sensing Systems For Energy-E_ciency In Buildings. ACM, 2011.

[30] G. Singh, S. Bharathi, A. Chervenak, E. Deelman, C. Kesselman, M. Manohar, S. Patil, and L. Pearlman. A metadata catalog service for data intensive applications. In Proceedings of the 2003 ACM/IEEE conference on Supercomputing, SC '03, pages 33-, New York, NY, USA, 2003. ACM.

References

[32] P. Stanton. Securing data in storage: A review of current research. CoRR, cs.OS/0409034, 2004.

[33] Y. Tang, P. P. C. Lee, J. C. S. Lui, and R. Perlman. Fade: Secure overlay cloud storage with _le assured deletion. In O. Akan, P. Bellavista, J. Cao, F. Dressler, D. Ferrari, M. Gerla, H. Kobayashi, S. Palazzo, S. Sahni, X. S. Shen, M. Stan, J. Xiaohua, A. Zomaya, G. Coulson, S. Jajodia, and J. Zhou, editors, Security and Privacy in Communication Networks, volume 50. 2010.

[34] W. Treese. Putting it together: the home area network. Networker, 4:11{13, 2000.