Volume 3, Issue 12, December 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Cryptographic Cloud Storage & Networking

Rohit S. Bhore, Sejal B. Bharkhada, Ashwini N. Malik, Prof. Anuja K Pande

Dept:CE,DBNCOET

Amravati University, India

Abstract*— Nowadays Data Security is a major field in Networking. Data security has been a leading issue in the Information Technology arena because as users we don't want anyone to hinder our privacy and as developers we don't want anyone to use our work as their own. Data Security does not only mean password protection, data hiding or adding additional firewalls it also means having complete information about your data i.e. where is your data kept and who all view it. The Cryptographic Cloud Storage and Networking has two basic parts i.e. Cryptography and second one is Cloud or Network Storage. Cryptography is the practice and study of techniques for secure communication in the presence of third parties More generally, it is about constructing and analysing protocols that overcome the influence of adversaries and which are related to various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation. And the term "Cloud" is analogical to "Internet". The cloud computing is Internet based computing where virtual shared server provides software, infrastructure, platform, devices and other resources. We consider the problem of building a secure cloud storage service on top of a public cloud infrastructure where the service provider is not completely trusted by the customer. We describe, at a high level, several architectures that combine recent and non-standard cryptographic primitives in order to achieve our goal.

***Keywords*— Cloud Computing, Cryptography, Security, Data Integrity.**

I. INTRODUCTION

The term "Cloud" is analogical to "Internet". The cloud computing is Internet based computing where virtual shared server provides software, infrastructure, platform, devices and other resources. All information that a digitized system has to offer is providing as a services in the cloud computing model. User can access these services available on the Internet Cloud without having any previous know-how managing the resources involved. Thus, user can concentrate more on their core business processes rather than spend time and gaining knowledge on resources that needed to manage on cloud. Clouds computing is a colloquial expression used to describe a variety of different types of computing concepts that involve a large number of computers that are connected through are all time communication network.

Data outsourcing to cloud storage servers is raising trend among many firms and users owing to its economic advantages This essentially means that the owner (client) of the data moves its data to a third party cloud storage server which is supposed to - presumably for a fee -faithfully store the data with it and provide it back to the owner whenever required. As data generation is far outpacing data storage it proves costly for small firms to frequently update their hardware whenever additional data is created. Also maintaining the storages can be a difficult task. Storage outsourcing of data to a cloud storage helps such firms by reducing the costs of storage, maintenance and personnel. It can also assure are liable storage of important data by keeping multiple copies of the data thereby reducing the chance of losing data by hardware failures. Storing of user data in the cloud despite its advantages has many interesting security concerns which need to be extensively investigated for making it a reliable solution to the problem of avoiding local storage of data. Many problems like data authentication and integrity (i.e., how to efficiently and securely ensure that the cloud storage server returns correct and complete results in response to its clients' queries, outsourcing encrypted data and associated difficult problems dealing with querying over encrypted domain were discussed in research literature. In this paper we deal with the problem of implementing a protocol for obtaining a proof of data possession in the cloud sometimes referred to as Proof of irretrievability (POR).This problem tries to obtain and verify a proof that the data that is stored by a user at a remote data storage in the cloud (called cloud storage archives or simply archives) is not modified by the archive and thereby the integrity of the data is assured. Such kinds of proofs are very much helpful in peer-to-peer storage systems, network file systems, long-term archives, web-service object stores, and database systems. Such verification systems prevent the cloud storage archives from modifying the data stored at it without the consent of the data owner by using frequent checks on the storage archives. Such checks must allow the data owner to efficiently, frequently, quickly and securely verify that the cloud archive is not cheating the owner. Cheating, in this context, means that the storage archive might delete some of the data or may modify some of the data. It must be noted that the storage server might not be malicious; instead, it might be simply unreliable and lose or inadvertently corrupt the hosted data. But the data integrity schemes that are to be developed need to be equally applicable for malicious as well as unreliable cloud storage servers. Any such proofs of data possession schemes do not, by itself, protect the data from corruption by the archive. It just allows detection of tampering or deletion of a remotely located file at an unreliable cloud storage

server. To ensure file robustness other kind of techniques like data redundancy across multiple systems can be maintained. While developing proofs for data possession at untrusted cloud storage servers we are often limited by the resources at the cloud server as well as at the client. Given that the data sizes are large and are stored at remote servers, accessing the entire file can be expensive in I/O costs to the storage server. Also transmitting the file across the network to the client can consume heavy bandwidths. Since growth in storage capacity has far outpaced the growth in data access as well as network bandwidth, accessing and transmitting the entire archive even occasionally greatly limits the scalability of the network resources. Furthermore, the I/O to establish the data proof interferes with the on-demand bandwidth of the server used for normal storage and retrieving purpose. The problem is further complicated by the fact that the owner of the data may be a small device, like a PDA (personal digital assist) or a mobile phone, which have limited CPU power, battery power and communication bandwidth. Hence a data integrity proof that has to be developed needs to take the above limitations into consideration. The scheme should be able to produce a proof without the need for the server to access the entire file or the client retrieving the entire file from the server. Also the scheme should minimize the local computation at the client as well as the bandwidth consumed at the client.

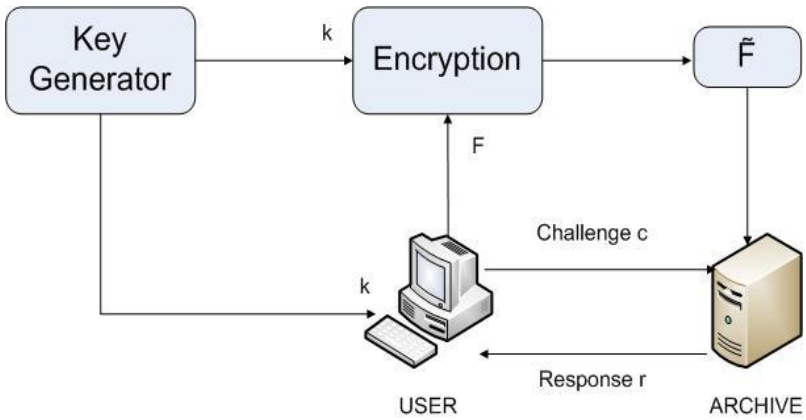

Fig 1: Overview of Cryptographic System

II. CLOUD COMPUTING FEATURES

A. *Why Cloud computing?*

Cloud computing is an industry transformation. Cloud computing enables businesses, of all sizes to deliver IT as a service, offering new possibilities to focus more on business success and less on operational costs and maintenance. There are many advantage a user can leverage from cloud computing. They are listed as follows,

- Cloud computing user avoids capital expenditure on building up an infrastructure to support their application. Instead, they pay the provider only the amount they consume.
- The user need not invest on the maintenance of the Infrastructure of the application. The provider maintains the infrastructure for the user.
- The user can access the multiple data servers from any location at a go.
- Enhancement of the application is easy, as the user need not worry about the infrastructure enhancement.
- Cloud computing is an eco-friendly incentive which will replace the hardware components with services.

B. *Cloud computing Features*

Cloud Computing brings features that distinguish it from classical resource and service provisioning environments, they are as follows,

- *Highly Scalable*–Cloud computing provides resources and services for users on demand. The resources are scalable over several data centres.
- *Less capital expenditure* -Cloud computing does not require upfront investment. No capital expenditure is required. Users may pay and use or pay for services and capacity as they need them.
- *Higher resource Utilization* - Cloud computing can guarantee QoS for users in terms of hardware or CPU performance, bandwidth, and memory capacity.
- *Disaster recovery and Back up*
- *Device and Location Independence*

C. *Cloud Computing Service Models*

- ***Software as a Service (SaaS)*** - It is a model of software deployment whereby the provider licenses an application to the customers for use as a service on demand. The capability provided to the End users is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web enabled e-mail). The end users does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user specific application configuration settings. Today SaaS is offered by companies such as Google, Sales force, Microsoft, Zoho, etc.

- ***Platform as a Service (PaaS)*** - It is the delivery of computing platform and solution stack as a service. The capability provided to the end users is to deploy onto the cloud infrastructure user created or acquired applications created using programming languages and tools supported by the provider. The end user does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage. PaaS providers offer predefined combination of OS and application servers, such as WAMP platform (Windows, Apache, MySQL and PHP), LAMP platform (Linux, Apache, MySQL and PHP), and XAMP(X-cross platform) limited to J2EE, and Ruby etc. Google App Engine, Salesforce.com, etc. are some of the popular PaaS examples.
- ***Infrastructure as a Service (IaaS)*** - It is the delivery of computer infrastructure (typically a platform virtualization environment) as a service. The capability provided to the end users is to provision processing, storage, networks, and other fundamental computing resources where the end user is able to deploy and run arbitrary software, which can include operating systems and applications. The user does not manage or control the underlying cloud infrastructure but it has control over operating systems, storage, deployed applications, and possibly limited control of select networking components. Some of the common examples are Amazon, Go Grid, 3tera, etc.

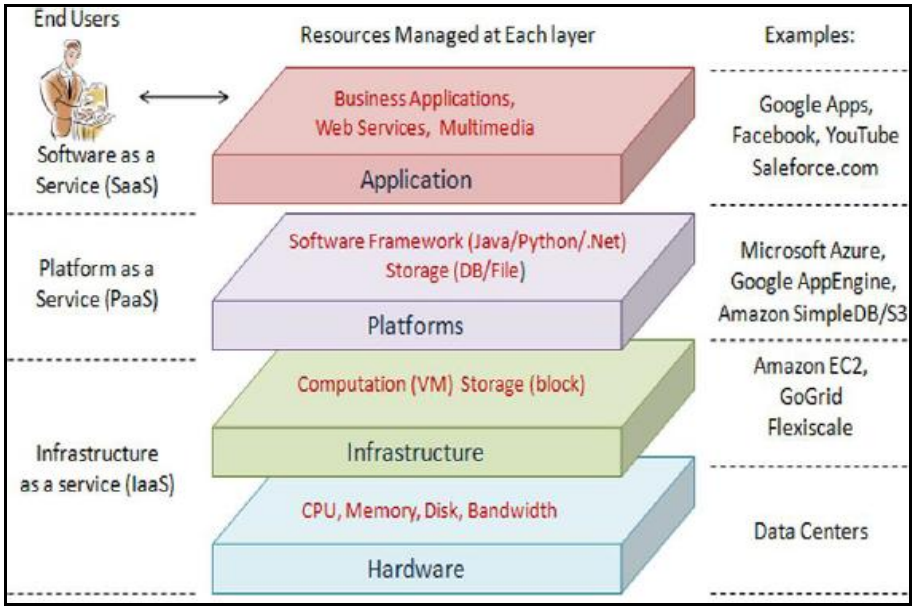

Fig 2: Cloud Computing Service Models

III. ARCHITECTURE OF A CRYPTOGRAPHIC STORAGE SERVICE

We now describe, at a high level, a possible architecture for a cryptographic storage service. At its core, the architecture consists of three components: a data processor (DP), that processes data before it is sent to the cloud; a data verier (DV), that checks whether the data in the cloud has been tampered with; and a token generator (TG), that generates tokens that enable the cloud storage provider to retrieve segments of customer data; and a credential generator that implements an access control policy by issuing credentials to the various parties in the system (these credentials will enable the parties to decrypt encrypted _les according to the policy). We describe designs for both consumer and enterprise scenarios.

A. *Consumer Architecture*

Consider three parties: a user Alice that stores her data in the cloud; a user Bob with whom Alice wants to share data and a cloud storage provider that stores Alice's data. To use the service, Alice and Bob begin by downloading a client application that consists of a data processor, a data verier and a token generator. Upon its rest execution, Alice's application generates a cryptographic key. We will refer to this key as a master key and assume it is stored locally on Alice's system and that it is kept secret from the cloud storage provider. Whenever Alice wishes to upload data to the cloud, the data processor is invoked. It attaches some metadata (e.g., current time, size, keywords etc.) and encrypts and encodes the data and metadata with a variety of cryptographic primitives. Whenever Alice wants to verify the integrity of her data, the data verier is invoked. The latter uses Alice's master key to interact with the cloud storage provider and ascertain the integrity of the data. When Alice wants to retrieve data (e.g., all _les tagged with keyword \urgent") the token generator is invoked to create a token. The token is sent to the cloud storage provider who uses it to retrieve the appropriate (encrypted) _les which it returns to Alice. Alice then uses the decryption key to decrypt the _les. Data sharing between Alice and Bob proceeds in a similar fashion. Whenever she wishes to share data with Bob, the application invokes the token generator to create an appropriate token, and the credential generator to generate a credential for Bob. Both the token and credential are sent to Bob who, in turn, sends the token to the provider. The latter uses the token to retrieve and return the appropriate encrypted documents which Bob decrypts using his

credential. This process is illustrated in Figure 1. We note that in order to achieve the security properties we seek, it is important that the client-side application and, in particular, the core components be either open-source or implemented or verier by someone other than the cloud service provider.

B. ***An Enterprise Architecture***

In the enterprise scenario we consider an enterprise Mega Corp that stores its data in the cloud; a business partner Corp with whom Mega Corp wants to share data; and a cloud storage provider that stores Mega Corp's data.

To use the service, Mega Corp deploys dedicated machines within its network. Depending on the particular scenario, these dedicated machines will run various core components. Since these components make use of a master secret key, it is important that they be adequately protected and, in particular, that the master key be kept secret from the cloud storage provider and Partner Corp. If this is too costly in terms of resources or expertise, management of the dedicated machines (or speci_c components) can alternatively be outsourced to a trusted entity. In the case of a medium-sized enterprise with enough resources and expertise, the dedicated machines include a data processor, a data verier, a token generator and a credential generator.

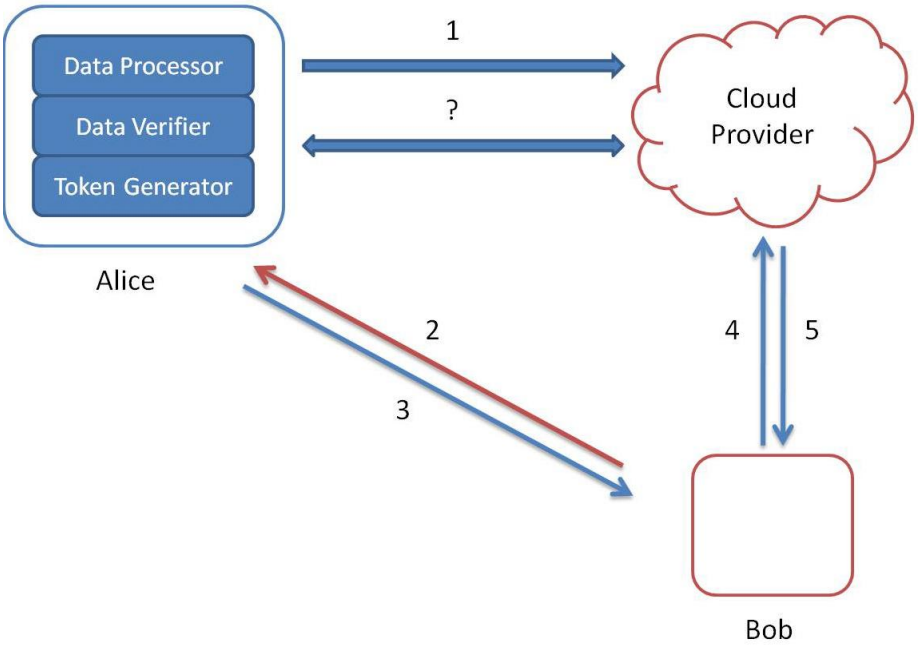

Fig 3: verify the integrity of the data.

Figure 3: (1) Alice's data processor prepares the data before sending it to the cloud; (2) Bob asks Alice for permission to search for a keyword; (3) Alice's token and credential generators send a token for the keyword and a credential back to Bob; (4) Bob sends the token to the cloud; (5) the cloud uses the token to find the appropriate encrypted documents and returns them to Bob. (?) At any point in time, Alice's data verier can verify the integrity of the data.

To begin, each Mega Corp and Partner Corp employee receives a credential from the credential generator. These credentials will reflect some relevant information about the employees such as their organization or team or role. Whenever a Mega Corp employee generates data that needs to be stored in the cloud, it sends the data together with an associated decryption policy to the dedicated machine for processing. The decryption policy species the type of credentials necessary to decrypt the data (e.g., only members of a particular team). To retrieve data from the cloud (e.g., all _les generated by a particular employee), an employee requests an appropriate token from the dedicated machine. The employee then sends the token to the cloud provider who uses it to find and return the appropriate encrypted _les which the employee decrypts using his credentials. Whenever Mega Corp wants to verify the integrity of the data, the dedicated machine's data verier is invoked. The latter uses the master secret key to interact with the storage provider and ascertain the integrity of the data. Now consider the case where a Partner Corp employee needs access to Mega Corp's data. The employee authenticates itself to Mega Corp's dedicated machine and sends it a keyword. The latter veriest that the particular search is allowed for this Partner Corp employee. If so, the dedicated ma- chine returns an appropriate token which the employee uses to recover the appropriate (encrypted) files from the service provider. It then uses its credentials to decrypt the _le. This process is illustrated in Figure 4. Similarly to the consumer architecture, it is imperative that all components be either open-source or implemented by someone other than the cloud service provider.

In the case that Mega Corp is a very large organization and that the prospect of running and maintaining enough dedicated machines to process all employee data is infeasible, consider the following slight variation of the architecture described above. More precisely, in this case the dedicated machines only run data variers, token generators and credential generators while the data processing is distributed to each employee.Note that in this scenario the data processors do not include the master secret key so the confidentiality of the data.

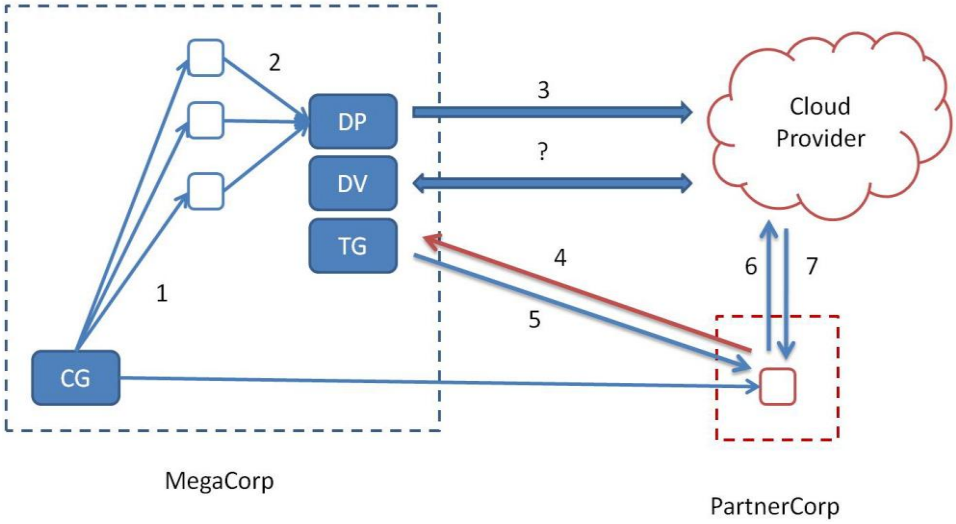

Fig 4: verify the integrity of Mega Corp's data.

Figure 4: (1) Each Mega Corp and Partner Corp employee receives a credential; (2) Mega Corp employees send their data to the dedicated machine; (3) the latter processes the data using the data processor before sending it to the cloud; (4) the Partner Corp employee sends a keyword to Mega Corp's dedicated machine ; (5) the dedicated machine returns a token; (6) the Partner Corp employee sends the token to the cloud; (7) the cloud uses the token to find the appropriate encrypted documents and returns them to the employee. (?) At any point in time, Mega Corp's data verier can verify the integrity of Mega Corp's data. is not affected. The data processors, however, do include some keying material which, if revealed to the service provider, could enable it to compromise the confidentiality of the tokens it receives.

IV. BENEFITS OF A CRYPTOGRAPHIC STORAGE SERVICE

The core properties of a cryptographic storage service are that control of the data is maintained by the customer and the security properties are derived from cryptography, as opposed to legal mechanisms, physical security or access control. Therefore, such a service provides several compelling advantages over other storage services based on public cloud infrastructures. In this section, we recall some of the main concerns with cloud computing as outlined in the Cloud Security Alliances recent report and highlight how these concerns can be mitigated by such architecture.

A. GEOGRAPHIC RESTRICTIONS- Data that is stored in certain legal jurisdictions may be subject to regulations even if it was not collected there. Because it can be difficult to ascertain exactly where one's data is being stored once it is sent to the cloud (i.e., many service providers have data centres deployed throughout the world) some customers may be reluctant to use a public cloud for fear of increasing their legal exposure. In a cryptographic storage service data is only stored in encrypted form so any law that pertains to the stored data has little to no exact on the customer. This reduces legal exposure for the customer and allows the cloud storage provider to make optimal use of its storage infrastructure, thereby reducing costs.

B. SUBPOENAS- If an organization becomes the subject of an investigation, law enforcement agencies may request access to its data. If the data is stored in a public cloud, the request may be made to the cloud provider and the latter could even be prevented from notifying the customer. This can have severe consequences for customers. First, it preempts the customer from challenging the request. Second, it can lead to law enforcement having access to data from clients that are not under investigation Such a scenario can occur due to the fact that service providers often store multiple customer's data on the same disks. In a cryptographic storage service, since data is stored in encrypted form and since the customer retains possession of all the keys, any request for the (unencrypted) data must be made directly to the customer.

C. SECURITY BREACHES-Even if a cloud storage provider implements strong security practices there is always the possibility of a security breach. If this occurs the customer may be legally responsible. In a cryptographic storage service data is encrypted and data integrity can be verier at any time.
Therefore, a security breach poses little to no risk for the customer.

D. ELECTRONIC DISCOVERY- Digital information plays an important role in legal proceedings and often organizations are required to preserve and produce records for litigation. Organizations with high levels of litigation may need to keep a copy of large amounts of data on-premise in order to assure its integrity. This can obviously negate the benefits of using a cloud storage service. Since, with a cryptographic storage service, a customer can verify the integrity of its data at any point in time a provider has every incentive to preserve the data's integrity.

E. DATA RETENTION AND DESTRUCTION- In many cases a customer may be responsible for the retention and destruction of the data it has collected. If this data is stored in the cloud, however, it can be difficulty for a customer to ascertain the integrity of the data or to verify whether it was properly discarded. A cryptographic storage service alleviates these concerns since data integrity can be verified and since the information necessary to decrypt data (i.e., the master key) is kept on-premise. Secure data erasure can be electively achieved by just erasing the master KEY.

V. CONCLUSION

Our scheme was developed to reduce the computational and storage overhead of the client as well as to minimize the computational overhead of the cloud storage server. we also minimized the size of the proof of data integrity so as to reduce the network bandwidth consumption. at the client we only store two functions, the bit generator function g, and the function h which is used for encrypting the data. hence the storage at the client is very much minimal compared to all other schemes that were developed. hence this scheme proves advantageous to thin clients like pdas and mobile phones. the operation of encryption of data generally consumes a large computational power. many of the schemes proposed earlier require the archive to perform tasks that need a lot of computational power to generate the proof of data integrity. but in our scheme the archive just need to fetch and send few bits of data to the client.

ACKNOWLEDGEMENT

We acknowledge our senior faculty who have provided us their views in the selection of topic.

REFERENCES

[1] E. Mykletun, M. Narasimha, and G. Tsudik, "Authentication and integrity in outsourced databases," Trans. Storage, vol.2, no.2,pp. 107–138, 2006.

[2] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in SP '00: Proceedings of the 2000 IEEE Symposium on Security and Privacy. Washington, DC, USA: IEEE Computer

[3] Society, 2000, p. 44.

[4] A. Juels and B. S. Kaliski, Jr., "Pors: proofs of retrievability for large files," in CCS '07: Proceedings of the 14th ACM conference on Computer and communications security. New York, NY, USA: ACM, 2007, pp. 584–597.

[5] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in CCS '07: Proceedings of the 14th ACM conference on Computer and communications security. New York, NY, USA: ACM, 2007, pp. 598–609.

[6] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. M. 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 '05, volume 3621 of Lecture Notes in Computer Science, pages 205{222. Springer,2005}.

[7] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song. Provable data possession at untrusted stores. In P. Ning, S. De Capitani di Vimercati, and P. Syverson, editors, ACM Conference on Computer and Communication Security (CCS '07). ACM Press, 2007.

[8] G. Ateniese, S. Kamara, and J. Katz. Proofs of storage from homomorphic identi_cation protocols. In To appear in Advances in Cryptology ASIACRYPT '09, Lecture Notes in Computer Science. Springer, 2009.

[9] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik. Scalable and e_cient provable data possession. In Proceedings of the 4th international conference on Security and privacy in communication netowrks (SecureComm '08), pages 1{10, New York, NY, USA, 2008. ACM.

[10] J. Baek, R. Safavi-Naini, and W. Susilo. On the integration of public key data encryption and public key encryption with keyword search. In International Conference on Information Security (ISC '06), volume 4176 of Lecture Notes in Computer Science. Springer, 2006.

[11] J. Baek, R. Safavi-Naini, and W. Susilo. Public key encryption with keyword search revisited. In International conference on Computational Science *and Its Applications, pages 1249{1259}. Springer-Verlag, 2008.*

