Persistent Data Security in Public Clouds

Sara Ibn El Ahrache
LabTIC, ENSAT
Tangier, Morocco

Hassan Badir
LabTIC, ENSAT
Tangier, Morocco

Abderrahmane Sbihi
LabTIC, ENSAT
Tangier, Morocco

ABSTRACT
Recently, there has been increasing confidence for a favorable usage of big data drawn out from the huge amount of information deposited in a cloud computing system. Data kept on such systems can be retrieved through the network at the user's convenience. However, the data that users send include private information, and therefore, information leakage from these data is now a major social problem. The usage of secret sharing schemes for cloud computing have lately been approved to be relevant in which users deal out their data to several servers. Notably, in a (k,n) threshold scheme, data security is assured if and only if all through the whole life of the secret the opponent cannot compromise more than k of the n servers. In fact, a number of secret sharing algorithms have been suggested to deal with these security issues. However, a limitation of these methods is that first they do not consider long term data storage and second they assume that data tempering only occurs at retrieval time, after the distribution of the shares has been correctly done. In this paper these two problems are addressed by presenting a novel scheme to ensure a perpetual secure data storage and retrieval.

General Terms
Cloud Computing, distributed storage, security.

Keywords
cloud computing; data security; secret sharing; blinding; distributed digital signature; threshold cryptography.

1. INTRODUCTION
Cloud computing is the delivery of computing and storage capacity as a service to a heterogeneous community of end-recipients. Cloud computing entrusts services with a user's data, software and computation over a network. As cloud computing is achieving increased popularity, concerns are being voiced about the security issues introduced through adoption of this new model. The effectiveness and efficiency of traditional protection mechanisms are being reconsidered as the characteristics of this innovative deployment model can differ widely from those of traditional architectures.

The relative security of cloud computing services is a contentious issue that may be delaying its adoption. Physical control of the Private Cloud equipment is more secure than having the equipment off site and under someone else's control. Physical control and the ability to visually inspect the data links and access ports is required in order to ensure data links are not compromised. Issues barring the adoption of cloud computing are due in large part to the private and public sectors' unease surrounding the external management of security-based services. It is the very nature of cloud computing-based services, private or public, that promote external management of provided services.

This delivers great incentive to cloud computing service providers to prioritize building and maintaining strong management of secure services. Security issues have been categorized into sensitive data access, data segregation, privacy, bug exploitation, recovery, accountability, malicious insiders, management console security, account control, and multi-tenancy issues. Solutions to various cloud security issues vary, from cryptography, particularly public key infrastructure (PKI), to use of multiple cloud providers, standardization of APIs, and improving virtual machine support and legal support.

Secret sharing schemes are now being considered for cloud systems, because they have the following features.

- They can distribute data to multiple servers and are resistant to system failure caused by natural disasters or human error.
- They can never leak information even if the number of shares is below the threshold and the system is the least computationally secure.

A well-known principle in the analog world is the term reduced trust, meaning that in order to keep a secret, the less knowledge or power each entity has the better; this is the basic philosophy upon which secret sharing schemes reside.

Secret sharing schemes protect the secrecy and integrity of information by distributing the information over different locations. For sensitive data these schemes constitute a fundamental protection tool, forcing the adversary to attack multiple locations in order to learn or destroy the information. In particular, in a (k + 1, n) threshold scheme, an adversary needs to compromise more than k locations in order to learn the secret, and corrupt at least (n-k) shares in order to destroy the information However, a limitation of these methods is that they assume that data corruption only occurs at retrieval time, after the distribution of the shares has been correctly done.

In this paper a specific problem is addressed which is that of guarantying that the process of storing the data is correct even when some of the participants (servers) fail. This is accomplished through the application of cryptographic techniques, such as the distributed digital signatures, distributed key management via threshold cryptography. First the different techniques used to build the model are reviewed. Then, a detailed description is given unfolding the key characteristics to create a cloud computing system capable of preventing an adversary from learning or destroying the secret throughout its lifetime. And finally, it will be demonstrated that the contributed approach is more performing than existing ones.

2. BACKGROUND
In the distributed cloud model, users store their data using a shared pool of resources provided by cloud service providers. Obviously, there are evident security concerns. Currently, a rapidly growing interest in the application of secret sharing schemes to secure data storage and retrieval has been noticed. Essentially, these schemes exhibit numerous appealing properties such as a perfect security, extendibility, and flexibility. The notion of secret sharing schemes was

discovered in 1979 by Shamir and Blakely separately. The fundamental functioning of a (k,n) scheme is to deal out the data to be saved, D, among n servers, in such a way that the retrieval of D is possible even in the presence of only k participants.

A Secret sharing scheme has extremely appealing characteristics: it is symmetric to all servers, it does not require a central authority, and no cryptographic keys are used. Alternatively, this set of extremely appealing characteristics is obtained at the cost of narrowing down the types of flaws the algorithm can handle, namely, by expecting that the available shares were correctly deposited and are never altered.

Various propositions to reproduce corrupted data were exhibited by specialists. Nonetheless, an impediment of these strategies is that they expect that data corruption happens at recovery time, after the distribution of the shares has been correctly done. Moreover, they do not take into consideration that the adversary has the entire life-time during which the data is stored to mount these attacks. In other words, it has been proved that if an attacker is given enough time he can compromise more than k servers. Therefore, for data stored for long period of time the security given by conventional secret sharing scheme might be insufficient. Researchers suggested multitude of solutions to this problem all founded on the notion of periodically refreshing the data shares. However, these methods have been proved to have many limitations that make them impractical in real world situations. In this paper a novel scheme that shall ensure a perpetual secure data storage and retrieval is presented.

3. RELATED WORK

To ensure and boost the security of data a user is willing to store on a cloud computing environment, Shamir's secret sharing scheme is used. Shamir's secret sharing scheme is called a perfect scheme simply because the size of the produced shares is exactly the same regardless of the number of the participant servers [1]. This perfect scheme has been used extensively by many researchers to produce various security approaches [2-4]. As a matter of fact, Kurihara et. al [3] have proposed a new (k,n) threshold scheme using only XOR operations in both the distribution and reconstruction phases; the authors not only presented the new scheme, but also demonstrated that it outperforms Shamir's scheme. Lin, C et. al [4] made two modifications to Shamir's scheme; the first adjustments consists of allowing the servers to store their share as well as the (x,y) coordinates, while the second adjustment consists of using a polynomial of degree higher than the threshold to generate data shares. Ito, M. et. al [2] suggests another version of the secret sharing scheme where they proved that their scheme can produce a secret sharing scheme from any access structure. In [5] the authors propose a multi linear secret sharing scheme which uses super polynomial bounds on the share size. In [6] the authors suggest a hierarchical secret sharing scheme where the data is partitioned among a group of servers into levels.

Security is an essential worry in cloud design as distributed storage is defenseless against security dangers. Ensuring data security and key management are the two most essential ones. The authors of the works [7–12] opted to research the secure data storage and key management in various cloud systems. Alsolami [8] present Cloudstash that shall ensure security of the distributed storage. It applies secret sharing algorithm straightforwardly on the data. To beat the impediments of a single area cloud, Intercloud has been presented in [9].This model performs symmetric encryption on the data and fragments the key into shares by secret sharing algorithm.

All these cited works expect that the distribution of the shares has been correctly done. Additionally, they overlook the fact that the opponent has the entire life-time during which the data is stored to gradually mount small undetectable dangerous attacks.

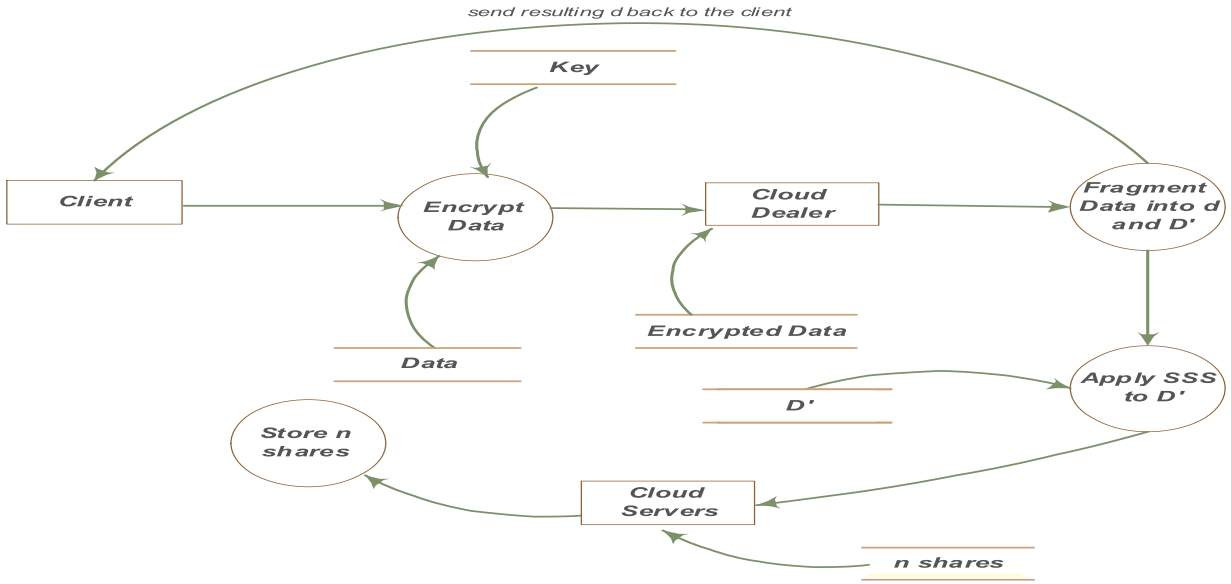

Fig. 1 The Dealing Phase

4. THE CONTRIBUTION

4.1 Long Term Security Scheme

As any scheme that uses the secret sharing algorithm, there are two phases: the dealing phase and the reconstruction phase. In the dealing phase a client who is already registered with a cloud computing service provider decides to deposit data in the cloud. In this phase, Shamir's secret sharing scheme is used. It allows the dealer to distribute the data to n servers, such that at least k servers are required to reconstruct the data.

The protocol is information theoretically secure, i.e., any fewer than k servers cannot gain any information about the data by themselves. Likewise, the same protocol is used for the reconstruction phase which is when this client decides to download his data from the cloud. Both operations are done upon client's request.

4.1.1 Dealing Phase

As demonstrated by figure 1, the scheme is composed of the user willing to store his data on a cloud service provider, the dealer or the cloud service provider gateway and the servers where the data is to be stored. Obviously, the number and location of the participating servers is transparent to the user. Although it is assumed that the communication channel between the client and the cloud gateway is secure, the client first encrypts the data before sending it to the dealer. When the dealer receives the data it extracts from it a one connection token (the algorithm used for this operation is to be discussed in upcoming publications). The token is a small subset of the original data to be safely kept by the client for future data retrieval. The dealer then applies the secret sharing algorithm to D' (data after the extraction of the token) and sends the n shares to the n participating servers. The strength of this algorithm resides in the token kept by the client. Not only an opponent needs to compromise at least k servers but even if he manages to attack all n servers he would still get a meaningless data because of the missing puzzle kept by the client.

4.1.2 Reconstruction Phase

When the client decides to retrieve his data, he asks the dealer to recover the necessary parts from the servers using secret sharing algorithm and then forwards them to the client. The client then adds the missing puzzle to the data and decrypts it. At this stage, the dealer regenerates a new token as the old one has became obsolete. Note that the dealing is done per connection in contrast to conventional proactive secret sharing schemes where the dealing is done periodically.

4.2 Correct Data Deposit Proof

The main contribution of this approach to give the user a proof (in the form of an acknowledgement) that his data has correctly been deposited at the cloud service provider by using distributed digital signatures. Besides, simple protocols are combined to enable correct and secure data storage while limiting extra space and processing overhead. The approach outlined in this paper relies partially on the work of Juan A. Garay et. al [13] where the authors consider the Information Dispersal Algorithm for distributed secure storage.

The decisive goal of this approach is data confidentiality. On one hand, this is achieved by applying Shamir's secret sharing scheme to distribute the data among n servers in such a way that any collusion of up to t servers should not be able to learn anything about the data. However, since all processing is done at the cloud service provider perimeter, data should as well travel confidentially between the user and the cloud gateway. To this end, the decision taken was to use cryptography with threshold cryptography and blinding techniques.

4.2.1 Model Description

The architecture of the system considered in this paper is similar to the architecture of any conventional cloud computing environment with an entry point and many servers among which data is to be distributed. In this model, the participating servers intercommunicate in order to distribute the data. Yet, users wishing to securely store their data only interact with the cloud interface. It is assumed that at most t servers can fail.

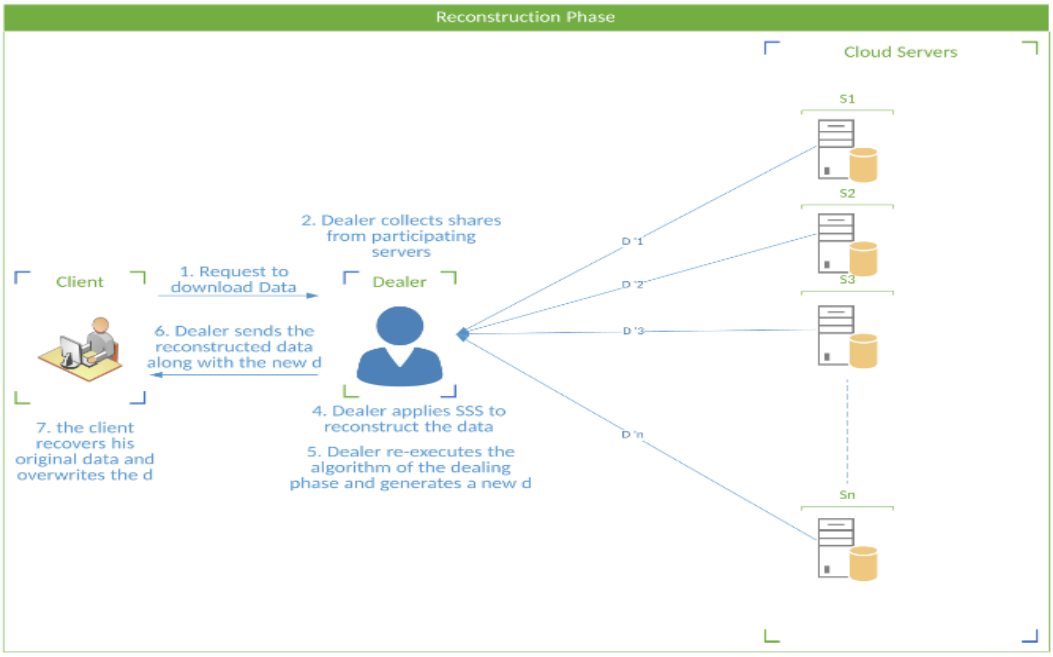

Fig. 2 The Reconstruction Phase

The following table summarizes the primitives used in the different protocols:

Table 1. Cryptographic Primitives

Primitive	Explanation
PCu, PRu	Public and private keys of user U
PCus, PRus	Public and private signing keys of user U
SKvi	Server Vi's share of private key SKv
H()	A strong one way hash function
Eu	Public key encryption using Pcu
Su()	Digital signature with respect to PRu
Sv,……. ,Svn()	Distributed digital signature with respect to SKvi
Ơvi()	Partial digital signature with respect to SKvi
Ek	Symmetric key based encryption algorithm

A user willing to store his data have two pairs of public/private keys: one is used for signing purposes and the second one for public key encryption decryption operations. As a matter of fact, the following question should be raised: how are the keys used to encrypt the data securely stored? i.e., the user has to securely guard PRu as loss of this private key would automatically cause losing all the encrypted data. To this end, this model stores PRu at the servers in a distributed manner.

Bellow is the protocol used in the proposed model. As a matter of fact, it can be decomposed into three main stages:

a) Transfer: the user sends data to the cloud gateway and gets an acknowledgement.

b) Partition: data is actually partitioned and distributed among the n servers.

c) Reconstruction: the user asks the cloud gateway to get the data back.

Figure 3 shows the flow of the first part "Transfer".

In the first part of this model, the user first generates a random key DK (like a DES key) and uses it to encrypt the data D, encrypts DK using his public key PCu, signs the encrypted data D' using his private key PRu and sends the data she wants to store to the cloud gateway, together with her signature on the file under her private authentication key PRu. The user keeps a hash of the data H(D') for future control. After reception of the user's request, the cloud gateway forwards it to the participating servers. Every server receiving a (valid) message from the cloud gateway transfers this message to every other server. Servers receiving at least one valid message store D' as a valid request from user U. Servers then use their share of the secret key to generate a partial signature on D', and send this message to the cloud gateway. After Receipt, the cloud gateway computes the distributed digital signature on D' and U, and sends it to the user. The finally user verifies the signature to ensure that the data was correctly transferred.

After the data has been correctly transferred, the servers are now ready to partition the data using a recursive secret sharing scheme adapted to this model. Figure 4 shows the flow of the second part "Partition".

Algorithm 1: Transfer Phase

Input: Data D, a symmetric key based algorithm e, public key PC_u, private key PR_u, private signing key PR_us

1. The user generates a random key D_k ;
2. The user encrypts D into D' using D_k and a symmetric key based algorithm e;
3. The user encrypts key D_k using public key encryption using PC_u;
4. The user signs D' using PR_u;
5. The user calculates hash of D', H(D');
6. The user sends D' along with its signature;
7. Cloud Gateway forwards user's request to all servers;
8. Every server echoes the request to every other server;
9. Servers receiving at least one valid message store D' as a valid request from user U;
10. Each server computes part of the signature using its share of PR_us;
11. Each server sends its part of the signature to the Cloud Gateway;
12. The cloud gateway computes the distributed signature by multiplying the partial signatures and send it to the user;
13. The user verifies the signature to ensure that data was correctly transferred;

Result: The user has a proof that data has been correctly deposited

Fig. 3 The Transfer Algorithm

Algorithm 2: Partition Phase

Input: Data D', user $ID = a_1$

Every participating server will execute the following:

1. Partition D' into n parts ;
2. Generate 1st degree polynomial $\mathcal{F}_1(x) = a_1(x) + d'_1$ where d'_1 is the first part of D';
3. Sample $\mathcal{F}_1(x)$ at two points $\mathcal{F}_1(1) = P_1$ and $\mathcal{F}_1(2) = P_2$ to generate two shares for d'_1 ;
4. Generate a new polynomial $\mathcal{F}_2(x)$ with P_1 and P_2 as coefficients and d'_2 as the free term ;
5. Sample $\mathcal{F}_2(x)$ at three points P_3, P_4 and P_5 to generate three shares for d'_2;
6. Delete P_1 and P_2 because the new shares P_3, P_4 and P_5 encode P_1 and P_2 within them;
7. Repeat the steps 2-6 for d'_3, d'_4, ..., d'_{k-2} by generating $\mathcal{F}_3(x)$, $\mathcal{F}_4(x)$, ..., $\mathcal{F}_{k-2}(x)$, sampling points, and deleting previous ones;
8. At this step n shares should have been generated and that have $k-1$ recursively secrets hidden within them;
9. Compute hash value for every n share;
10. Each server saves its own share and the corresponding hash value;

Output: n shares, n hash values

Fig. 4 The Partition Algorithm

Before starting the execution of the Partition part, every participating server has a copy of the user's data D'. There is no interaction taking place during this phase between any of the actors of this model. Rather, every participating server computes everybody's share of D' using a modified recursive secret sharing scheme as well as the corresponding hashes of the resultant shares, and saves its own share of the file along with all the hash values.

If the user decides to retrieve his data, he simply contacts the cloud gateway and sends him a number of parameters as shown in figure 5.

The user first generates a random integer r that he saves securely; r will be his blinding factor. He then computes b=Eu(r) (RSA could be used at this case). The user then signs b and the data id with his signing key, and sends to the cloud gateway. The cloud gateway transfers this request to each of the participating servers. The servers check that the user signing this request has permission to access the requested data. If so, each server generates Pi a partial decryption of Eu(DK) * b = Eu(DK* r). Each server then sends Di, the hashes, and Pi to the cloud gateway. The cloud gateway reconstructs the data and computes the value P = DK • r from the partial decryptions. The cloud gateways now sends the encrypted file and the blinded key P = (FK. r) to the user. The user obtains the data key DK by factoring out r, and acknowledges receipt of the file.

Algorithm 3: Reconstruction Phase

Input: a symmetric key based algorithm $\mathcal{E}_u$, public key PC_u

1. Generate a random integer r called the blinding factor and saves it securely ;
2. Compute $\mathcal{E}_u(r) = b$, public key encryption using PC_u ;
3. Sign b and the data D using PR_{us} and send it to the cloud gateway ;
4. Cloud Gateway forwards user's request to all servers ;
5. The servers check that the user sending this request has permission to access this data;
6. Each server V_i generates P_i, a partial decryption of $\mathcal{E}_u(D_k).b = \mathcal{E}_u(D_k.r)$ using d_i the share of PR_u held by V_i;
7. Each server V_i sends D'_i, hash values, P_i to the cloud gateway;
8. The cloud gateway computes $eD_i(D')$ using the hashes and the $D_i s$. It also computes the value $P = D_k.r$ from P_i;
9. The cloud gateway now sends D' and the blinded key $P = (D_k.r)$ to the user;
10. The user gets the data key D_k by factoring out r, and acknowledges receipt of the data;

Result: The user reconstructs successfully the original data

Fig. 5 The Reconstruction Algorithm

5. CONCLUSION AND FUTURE WORK

The corresponding security of a distributed computing plan is a disputable matter that might be abating its reception. Actually, numerous might contend that controlling an on location Private Cloud is more secure than having the assets under another person's power. Be that as it may, this is the embodiment of distributed computing based administrations, private or open, which is the advancement of outer administration of gave administrations.

The security of a cloud computing system is a disputable matter that might be stopping its total spread. Actually, the majority prefers to store and control their data onsite rather than handing it over to external non-trusted parties. However, pushing these tasks outside the perimeters is one essential cloud computing feature.

Secret sharing schemes have proved their security significance when used to secure cloud data storage. In these schemes, clients send out their data to a few servers. However, it has been showed that these schemes have a number of limitations. In this paper two of these issues have been addressed by displaying a novel plan to guarantee a never-ending secure data deposit and recovery. Currently, the presented prototype is being implemented and in the future its performance is going to be compared to existing solutions in order fine-tune it and hard-proof its effectiveness.

6. REFERENCES

[1] Shamir, A.: How to share a secret. Commun. ACM 22(11), 612–613 (1979).

[2] Ito, M., Saito, A., Nishizeki, T.: Secret sharing scheme realizing general access structure. Electron. Commun. Jpn. (Part III: Fundam. Electron. Sci.) 72(9), 56–64 (1989)

[3] Kurihara, J., Kiyomoto, S., Fukushima, K., Tanaka, T.: A New (k,n)-threshold secret sharing scheme and its extension. In: Wu, T.-C., Lei, C.-L., Rijmen, V., Lee, D.-T. (eds.) ISC 2008. LNCS, vol. 5222, pp. 455–470. Springer, Heidelberg (2008)

[4] Lin, C., Harn, L., Ye, D.: Ideal perfect multilevel threshold secret sharing scheme. In: Fifth International Conference on Information Assurance and Security, IAS 2009, vol. 2. IEEE (2009)

[5] KBeimel, A., Ben-Efraim, A., Padr´o, C., Tyomkin, I.: Multi-linear secret-sharing schemes. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 394–418. Springer, Heidelberg (2014)

[6] Tassa, T.: Hierarchical threshold secret sharing. J. Cryptol. 20(2), 237–264 (2007)

[7] Blakley, G.R.: Safeguarding cryptographic keys. In: International Workshop on Managing Requirements Knowledge. IEEE Computer Society (1989).

[8] Alsolami, F., Boult, T.E.: CloudStash: using secret-sharing scheme to secure data, not keys, in multi-clouds. In: 11th International Conference on Information Technology: New Generations, ITNG 2014. IEEE (2014)

[9] Cachin, C., Haas, R., Vukolic, M.: Dependable storage in the intercloud. Research report RZ 3783 (2010)

[10] Alsolami, F., Chow, C.E.: N-Cloud: improving performance and security in cloud storage. In: IEEE 14th International Conference on High Performance Switching and Routing, HPSR 2013. IEEE (2013)

[11] Bessani, A., et al.: DepSky: dependable and secure storage in a cloud-of-clouds. ACM Trans. Storage (TOS) 9(4), Article No. 12 (2013)

[12] Xiong, H., Zhang, X., Zhu, W., Yao, D.: CloudSeal: end-to-end content protection in cloudbased storage and delivery services. In: Rajarajan, M., Piper, F., Wang, H., Kesidis, G. (eds.) SecureComm 2011. LNICST, vol. 96, pp. 491–500. Springer, Heidelberg (2012)

[13] Juan A. Garay, Rosario Gennaro, Charanjit Jutla, Tal Rabin: Secure distributed storage and retrieval. In: Theoretical Computer Science 243 pp. 363-389 (2000)