A Comprehensive Security Model for Networking Applications

Eric Chan-Tin, Tingting Chen and Subhash Kak
Computer Science Department
Oklahoma State University
Stillwater, OK, USA
{chantin, tingtic, subhashk}@cs.okstate.edu

Abstract—**The Internet is currently being used by millions of users for web browsing, data storage, social networks, communications, VOIP, e-commerce, and other applications that are enabled by wireless networks, cloud computing, distributed systems, and cellphone networks. The architecture for these applications is not secure as shown by all the recent widely-publicized attacks. We propose a comprehensive security model for networking applications that includes new key distribution and management techniques and a realistic trust model.**

Keywords – security, key management, key distribution, trust, privacy

I. Introduction

As the Internet, the cloud, and smartphones have become popular [1], there is increasing anxiety about their security due to continuing attacks on major companies. The stock exchanges in New York and London were hacked on a couple of occasions. In early 2011, RSA computers were attacked undermining the reputation of RSA's popular SecurID security service. This attack was linked to the penetration of computer networks at some of RSA's most powerful defense-contractor clients — among them, Lockheed Martin, Northrop Grumman, and L-3 Communications. It begs the question that if major companies can be so "easily" hacked and information stolen, what can the rest of us hope in terms of security and protecting vulnerable information on our computers or servers? Although it has not been disclosed how the attacks took place, they were most likely due to security vulnerabilities in the software used or mismanagement of user accounts. Many attacks focused on data disclosures, for example, stealing of credit card numbers or social security numbers from retail vendors, or the disclosure of private US diplomatic cables by WikiLeaks. One of the reasons for these massive data disclosures is that many people had access to them, authentication controls were lax, and there was no protection mechanism in place to avoid the dissemination of the data. Clearly, the current security mechanisms are insufficient in protecting the widely disseminated data on the Internet.

An additional problem is that in many computing scenarios such as clouds the line between the secure inside and the insecure outside is blurred and therefore one needs new ways of preventing malicious inside attacks by using a trusted cloud platform.

We propose a comprehensive security model that incorporates new approaches of distribution, management as a service [2][3], together with a trust model. Our comprehensive security model can be applied to wireless networks, cloud computing, distributed systems, and cellular networks. Our position is that a *new key distribution and management technique with a realistic trust model is required for the Internet to keep growing and being usable*. Although the contrasting requirements of confidentiality and verifiability preclude a general solution for the network security problem, satisfactory applications-specific solutions, guaranteeing adequate security, privacy, and reliability constitute a trustworthy computing framework for many critical applications in the information society. Our comprehensive security framework incorporates techniques of key dispersal, cryptography, and a trust model in a generalized networking environment. The novelty of this approach lies in our use of new algorithms and trust models that provide a basis for systematic analysis of security in the cloud. The analysis of security will be based on the formal model in which the adversary can interact with legitimate users and what the adversary needs to do in order to break the system will be specified. We advocate the use of the random oracle model in which we assume that the hash functions are totally random about whose internal workings the adversary has no information. This provides a basis of provable security that is essential to the development of a secure and trustworthy Internet. Our model can be easily deployed without changing the current Internet architecture.

One of the components of our comprehensive security model is the use of information dispersal algorithms (IDA**)**. It divides up data and stores it over many different servers in a manner so that no single part of it reveals the data. New information dispersal techniques together with efficient reconstruction algorithms that will perform well in the presence of erratic servers and adversaries are needed. The new techniques consider a general model of risk distribution that is appropriate for situations related to a wide variety of applications. Mathematically, one needs verifiable information dispersal techniques so that a client can distribute the data among a set of servers, of which a certain subset might be faulty or compromised, in such a way that the client can always recover the stored data correctly, independent of the behavior of the faulty servers, and not have the data compromised to the adversary.

978-1-4673-1544-9/12/$31.00 ©2012 IEEE

Along with an efficient, secure, and practical key distribution mechanism, the user needs to ensure that the service providers are following the rules and service of agreements. For example, a malicious server can remove the data it claims to store for users. Therefore, users need a way to verify that the server(s) is storing the data without having to download all the data. An efficient proof of possession [4], possibly a proof of possession on data stored on multiple servers, is needed. Moreover, the authenticity and integrity of the data is required. Key distribution ensures that keys are managed properly, and those keys are used to encrypt data, for signature (authenticity), and to ensure integrity (MAC).

Trust, which is an important notion in data access and dissemination, forms part of this paper. Users need to “trust” other users and the service providers and the servers need to “trust” the users and other servers. We analyze various trust issues and propose a practical trust model and trust management scheme for use in the Internet.

The multiple applications that could benefit from such key distribution and management techniques are entities sharing data, such as federated intrusion detection systems and monitoring of the Internet, anti-virus signatures, wireless home networks, cellular networks, and in cloud services. As with any “public” information, care must be taken to protect the privacy of the users updating and accessing the data regardless of whether public keys or secret keys are used. Users could potentially use an anonymous system like Tor to access the cloud; however this only makes them anonymous on the Internet. The cloud provider could still know that person A has accessed key X. Moreover, the publisher might want to restrict access to sensitive information to only certain users, for example, if the publisher was sharing its network logs; it would first need to anonymize the logs (by removing identifiable information), then anonymize that it was the publisher (by using pseudonyms). Thus, a privacy-preserving key distribution and management scheme is required.

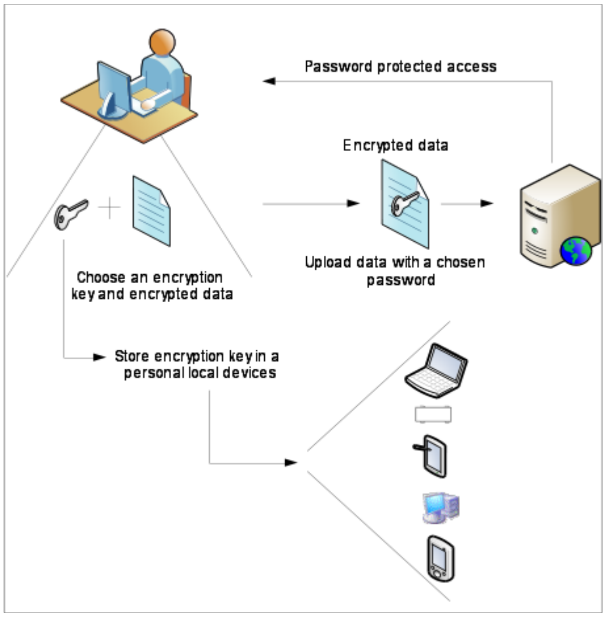
Password protected access
Encrypted data
Choose an encryption key and encrypted data
Upload data with a chosen password
Store encryption key in a personal local devices

Figure 1: Conventional Data Security

We explain a possible new information dispersal algorithm in Section II. Section III describes a realistic trust model, and Section IV outlines the design of a privacy-preserving key distribution and management scheme. We conclude with future directions in Section V.

II. Information Dispersal Algorithms

Various services ranging from data storage to providing users with access to proprietary application software on a pay-per-use basis have emerged. Users can obtain access to supercomputers, clusters of processors and other resources that are suitable for high performance analytics, high volume transactions and other applications in engineering, mathematical sciences, finance/business, medical, pharmaceutical research and so on. Storage providers such as Amazon S3, Google documents, and RackSpace offer virtually unlimited storage for free or very low cost. Storage services are being used as a backup for Apple users, federated intrusion detection systems, centralized anti-virus signatures, RAID backup, and for any user wanting to share data or perform experiments on top of virtual machines.

The concept of remote servers storing massive amounts of data is creating the need for solutions to several basic scientific problems related to reliability and integrity. Figure 1 illustrates how data security is currently achieved. A user chooses an encryption key and encrypts his data using the secret key. The user then uploads the data and optionally the key (with a chosen passphrase) onto a storage server. The issue with this approach is that one storage server contains both the data and the secret key. The server needs to be trusted to behave honestly, and to be secure against attacks from various adversaries to steal any data.

One way to deal with risk and potential failure of the system is to include the adversary in the security model. Traditionally, the idea of explicit security by instituting firewalls has been used but it is not effective in dealing with natural or man-made disasters. The idea of information dispersal (IDA) is an example of implicit security in which the data is sliced and stored over many different servers, which is illustrated in Figure 2. Its direct application is to survivability of the system and protection against data loss due to disasters and intrusions as was the motivation in survivable storage systems of Delta-4 [5],[6], Publius [7], Intermemory [8], etc. IDA systems make the self-healing of data possible, and if the data on a device becomes corrupted or destroyed, an automated process can detect this and recalculate all data contained in the missing or corrupted slice by inspecting available slices.

By coding and dispersing information, the reliability, security and efficiency of data storage can be vastly improved by IDA systems over traditional copy and parity-based systems. Copy-based systems offer reliability by mirroring data on *n* storage devices, and up to *n-1* of the devices can fail without data loss. But this comes at a high cost which is *n* times greater than storing a single copy. Parity-based systems, commonly used in RAID [9], typically allow at most two storage devices to fail without data loss. Parity-based systems are not as wasteful as copy-based systems but they are also equally reliable.

A special benefit of IDA schemes over copy-based systems is the inherent security advantages of dispersing information since the data is not stored at any single location. Each stored slice is a small and unique fraction of the data. An IDA system can be configured to disperse data to *n* number of devices which can sustain up to *n-k* failures without loss. Even if an attacker gained access to the devices, the system remains secure until at least *k* devices are compromised. Maintaining multiple copies creates multiple points of attack, further decreasing the reliability of the system. IDA systems are efficient since the storage overhead is low. They offer the high reliability of copy-based storage and the high efficiency of parity-based systems.

IDA systems are not only reliable, secure, and less expensive than backup and archive systems; they are the obvious choice for geographically distributed storage. If each of the *p* devices is kept at a different location throughout the world, then the resultant system is effectively a disaster proof backup system. So long as *p-m* locations remain up (where *m* is the number of devices that can fail) , the data would be safe and accessible even in the presence of power failure, earthquakes, floods, and other disasters.

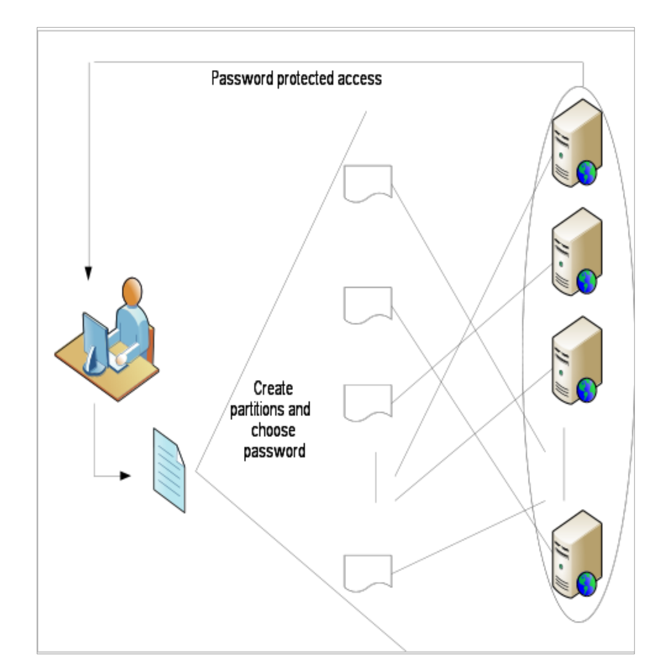
Password protected access
Create partitions and choose password

Figure 2: Security using IDA

Some IDA storage systems use *(p,k,n)* data distribution schemes, where the data is divided into pieces so that fewer than *p* pieces do not reveal any information, *p* or more than *p* but less than *k* pieces reveal some information and at least *k* pieces are required to reveal the entire data. IDA systems using secret sharing scheme [10] use *(k,k,n)* data distribution schemes, which create *n* pieces of the same size as the data and thus are space inefficient. Rabin's [11] information dispersal scheme is a *(1,k,n)* data distribution scheme that simultaneously encodes *k* values into *n* pieces and each piece reveals partial information about the encoded values. Other schemes with general values of *p*, *k* and *n* are known as implicit security schemes [12],[13] that provide considerable flexibility and efficiency.

An additional layer of security may be provided by using an encryption key to encrypt the data before dividing it into pieces [39]. The encryption key is then stored using a threshold secret sharing scheme of type *(k,k,n)* and the encrypted data is divided into *n* pieces using *(1,k,n)* information dispersal scheme. Such a hybrid scheme balances security and space efficiency.

III. TRUST

The Internet and related storage servers have their unique characteristics in terms of the system structure and computing scenarios, such that trust management issues [14],[15] need to be investigated specially for these systems. On the Internet, complicated roles and identities increase the complexity in understanding and managing the trust relationship. In particular, we recognize at least the following unique types of trust relationships in cloud computing systems.

1) Trust between service providers and the clients is the most common trust relationship. It is easy to observe that the service providers need to be trustworthy to the clients so that they are willing to use the services (e.g., storing data in a server or using data processing services). Moreover, since the data or processing requests from a client may be malicious to the service providers, this type of trust relationship should be mutual.

2) Trust between different service providers may be difficult to be observed in the current implementations. We thus need dedicated research efforts to investigate this particular type of trust relationship.

3) Trust between different clients is yet another type of trust relationship that we need to manage. Data from different clients may be stored in the same server. Although various data protection methods have been proposed [16],[17] to separate the data from different clients, clients may still have the concern that whether other clients on the same server are their competitors and whether they have the incentive to attack the data. Therefore, this type of trust relationship also needs to be thoroughly exploited.

These trust relationships are not isolated; they may connect and intersect in certain scenarios, and form a long trust chain. It brings more challenges in investigating the trust issues.

We notice that traditional trust models are mainly identity-centric trust, that is, the system administrator or peer users evaluate and maintain a trust value for each identity in the system. Thus, one needs to develop new trust models incorporating the unique characteristics of these systems. For example, trust should just be modeled in a service-centric way. In particular, in the service-centric data trust model, each service or service request should be evaluated whether it is trustworthy. Service-centric trust is particularly useful in preventing the attacks that utilize the services. In particular, one should investigate how to compute the trustworthiness of a service $\mathbf{s_p}$ generated by a service provider p , by using both relatively static information on trustworthiness such as general service type $\tau(s)$, and dynamically changing information captured by the current system security status χ_t and the specific metrics $\ell(s_p)$ that measure the requested service. We combine these as the arguments into the function to compute the trustworthiness of s_p as follows.

$$T(s_p) = F(\tau(s), f(\chi_t, \ell(s_p)))$$

In addition to evaluating the trustworthiness of a service request from a client, it is also needed to build corresponding trust models to enable the trust assessment for the services offered by the servers, to protect the clients. As another novel trust model, one needs to investigate the data-centric trust in this type of environment. Data-centric trust provides a way for clients to measure whether it is trustworthy to store their data in a public server, with the presence of data from others.

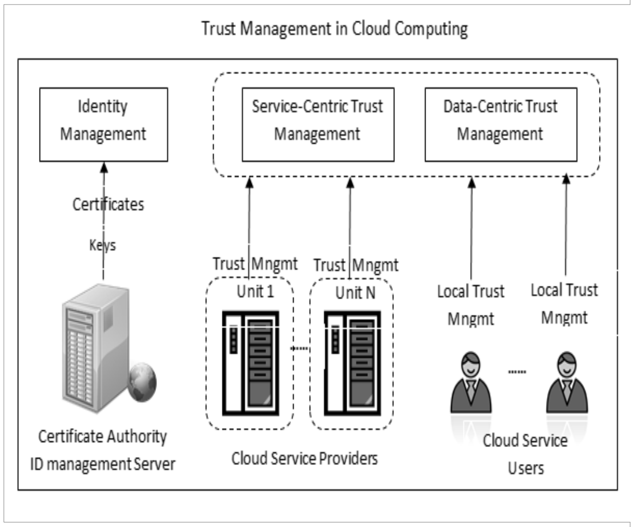

Figure 3: Trust Management

As the second step of trust management, trust evaluation mechanisms based on the new trust models should be designed. In particular, a trust evaluation system for the Internet shall be built, by leveraging previous works on reputation systems in other types of distributed systems. We note that although some general ideas in existing reputation systems (e.g., monitoring and event reporting) can continue to serve, new designs of trust evaluation mechanisms are needed. For instance, to prevent botnet attacks, a new trust evaluation mechanism that can detect and punish the distributed misbehavior is needed. The system architecture of the trust management system is shown in Figure 3.

IV. PRIVACY-PRESERVING KEY DISTRIBUTION

In a privacy-preserving key distribution mechanism, we need to make sure that 1) data sent to the server(s) is secure against both passive and active adversaries, 2) data stored in the server(s) is secure and inaccessible by unauthorized users, and 3) data is secure against corrupted or colluding servers.

Although it is straightforward for a user to encrypt its data and store it in the storage server, it is not very scalable [2][3][18]. Only the user can decrypt its data using the same device it was encrypted on (unless the user copies its key to other devices such as its smart phone, car, etc..., which is not convenient for the average user). The user could also specify multiple servers to replicate its data on to ensure high availability, but this is not practical; a more automated and intuitive method of storing data is required. Moreover, the user cannot share the contents of its data with other users. Depending on the encryption software used, it might also be cumbersome to re-encrypt the whole data if a small change is made.

Based on this model, one can design a new key distribution scheme which is easy for the average user to use, and allows sharing of information on distributed systems, such as cloud, or peer-to-peer systems. The user would still encrypt its data, but will also store the key to the distributed system. To prevent servers collusion, the keys can be secret-shared, or broken up using a threshold secret-sharing scheme such as IDA. This allows the user to access his files anywhere. This also allows the user to share his data with his family and/or friends. A practical and easy-to-use key distribution scheme is required. Some issues with current key distribution mechanisms are that authentication and anonymity of the sharer and those who access the data are important. Thus, a privacy-preserving key distribution scheme needs to be investigated.

Our threat model is that none of the servers can be trusted. Moreover, they could collude, but we assume that the majority of servers are honest. If all the servers are malicious and collude, then there can be no secure and private key dispersal technique. Moreover, we assume that an adversary can monitor all traffic the user sends, and all traffic that the servers receive or send. The adversary could also perform Sybil attacks to try to gain access to the data, for example, if it had control over a botnet.

Many applications could potentially benefit from the proposed secure and privacy-preserving key distribution scheme. Such examples include wireless networks, cellular networks, cloud computing, smart grid, peer-to-peer systems, and other distributed systems. The broader impact of this research is that regular users can easily and anonymously share data. Moreover, this could be used in large enterprises or government entities.

There has been a lot of work on key distribution and key dispersal techniques [19],[20],[21]. Moreover, there has been previous research performed for privacy-preserving techniques, such as privacy-preserving data aggregation [22] and privacy-preserving data mining [23]. However, there has been very little work done on privacy-preserving key distribution. The scheme should be easy-to-use and efficient.

V. FUTURE DIRECTIONS

The schematic of our proposed comprehensive security model for Internet applications is shown in Figure 4. It includes the Information Dispersal Algorithm with the privacy-preserving key distribution, along with trust relationships among all the entities.

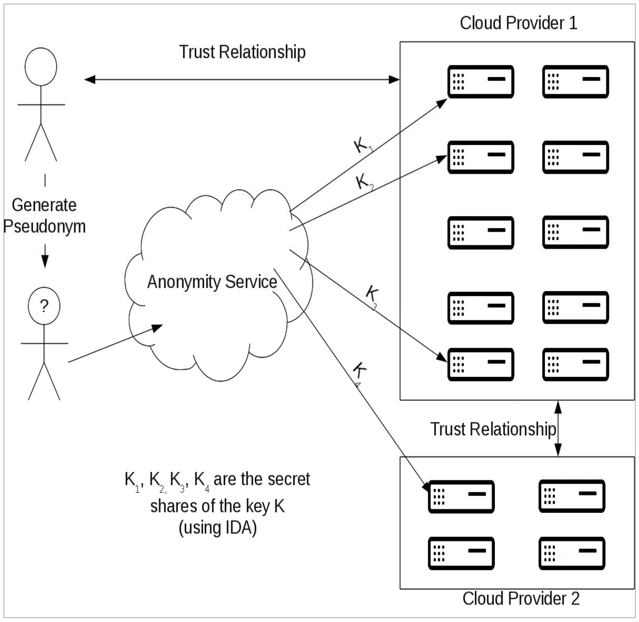

Figure 4: Schematic of the proposed design

Applications that can benefit from this security model are wireless networks – how data can be shared in an ad-hoc network and how trust can be leveraged in these networks; cellular networks – how other phones and base stations can be trusted and how phones can interact with the current Internet architecture to store files, such as pictures, music online; cloud computing – where data can be stored at multiple cloud servers and the cloud service providers' trust, with regards to each other and the users, needs to be investigated.

Since current storage techniques are insufficient in terms of security in today's Internet, we propose several avenues of research which would strengthen the security on the Internet. New information dispersal algorithms (IDA) will allow efficient storage of data at multiple servers, by dividing up the data or the key into smaller pieces. This will ensure the availability of the data and the security and privacy of the data by preventing the servers from decrypting the data. A realistic trust management model is also described, showing how servers and clients can trust each other. What trust actually means on the Internet needs to be explored and quantified. Finally, we propose the need for a privacy-preserving key distribution that allows the secret keys to be stored on the same servers that the data is stored at. The key distribution scheme also provides anonymity to the users, as the servers cannot link the data with the actual users.

REFERENCES

[1] Bowers, K., Juels, A., and Oprea, A. HAIL: A High-Availability and Integrity Layer for Cloud Storage. In *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS'09)* 2009.

[2] Zhu, S., Setia, S., and Jajodia, S. LEAP: Efficient security mechanisms for large-scale distributed sensor networks. In *Proceedings of 10th ACM conference on Computer and Communications Security*, Washington, DC, pages 62–72, 2003.

[3] Rogaway, P. and Bellare, M. Robust computational secret sharing and a unified account of classical secret-sharing goals. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*. Alexandria, Virginia, pages 172-184, 2007.

[4] Juels, A. and Kaliski B. PORs: Proofs of Retrievability for Large Files. In *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS'07)*, pages 584-597, 2007.

[5] Fray, J. M., Deswarte, Y. and Powell, D. Intrusion-tolerance using fine-grain fragmentation-scattering. In *Proceedings of IEEE Symposium on Security and Privacy,* pages 194-201, 1986.

[6] Deswarte, Y., Blain, L. and Fabre, J. C. Intrusion tolerance in distributed computing systems. Proceedings of 1991 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, pages 110-121, 1991.

[7] Waldman, M., Rubin, A. D. and Cranor, L. F. Publius: a robust, tamper-evident, censorship-resistant, web publishing system. In *Proceedings of USENIX Security Symposium*, pages 59-72, 2000.

[8] Chen, Y., Edler, J., Goldberg, A., Gottlieb, A., Sobti, S., and Yianilos, P. A prototype implementation of archival Intermemory. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, Berkley, CS, pages 28-37, 1999.

[9] Vijayan, S., Selvamani, S. , Vijayan, S. Dual-Crosshatch Disk Array: A Highly Reliable Hybrid-RAID Architecture. *Proceedings of the 1995 International Conference on Parallel Processing: Volume 1*. CRC Press. pp. 1-146, 1995.

[10] Shamir, A. How to share a secret. *Communications of the ACM*, volume 22, issue 11, pages 612-613, 1979.

[11] Rabin, M. O. Efficient dispersal of information for security, load balancing, and fault tolerance. *Journal of the ACM* , volume 36, issue 2, pages 335-348, 1989.

[12] Parakh, A. and Kak, S. Online data storage using implicit security. Information Sciences, vol. 179, pp. 3323-3331, 2009.

[13] Parakh, A. and Kak, S. Space efficient secret sharing for implicit data security. Information Sciences, vol. 181, pp. 335-341, 2011.

[14] Ernesto Damiani , De Capitani Di Vimercati , Stefano Paraboschi , Pierangela Samarati , Fabio Violante, A reputation-based approach for choosing reliable resources in peer-to-peer networks, Proceedings of the 9th ACM conference on Computer and communications security, CCS '02.

[15] Jøsang, A., Ismail, R., and Boyd, C. A survey of trust and reputation systems for online service provision, Decision Support Systems Volume 43, Issue 2, March 2007, Pages 618-644.

[16] Yu, S., Wang, C., Ren, K., and Lou, W. Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing, 2010 Proceedings IEEE INFOCOM.

[17] Goh, E., Shacham, H., Modadugu, N., and Boneh, D. "Sirius: Securing remote untrusted storage," in *Proceedings of NDSS*, 2003.

[18] Rocha, F., Abreau, S., Correia, M., The final frontier: confidentiality and privacy in the cloud. IEEE Computer, vol. 44, September 2011.

[19] Steiner, M., Tsudik, G., and Waidner, M. 1996. Diffie-Hellman key distribution extended to group communication. In *Proceedings of the 3rd ACM conference on Computer and communications security* (CCS '96). ACM, New York, NY, USA, 31-37.

[20] Burmester, M. and Desmedt, Y. A secure and efficient conference key distribution system. In Advances in Cryptology – EUROCRYPT 1994.

[21] Blundo, C., De Santis, A., Herzberg, A., Kutten, S., Vaccaro, U., and Yung, M. Perfectly-Secure Key Distribution for Dynamic Conferences. In CRYPTO 1992.

[22] He, W., Liu, X., Nguyen, H., Nahrstedt, K., and Abdelzaher, T.T. "PDA: Privacy-Preserving Data Aggregation in Wireless Sensor Networks," *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE* , vol., no., pp.2045-2053, 6-12 May 2007.

[23] Vaidya, J. and Clifton, C.. 2003. Privacy-preserving *k*-means clustering over vertically partitioned data. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (KDD '03). ACM, New York, NY, USA, 206-21