International Journal on Emerging Technologies (Special Issue on NCRIET-2015) **6**(2): 182-187(2015)

ISSN No. (Print) : 0975-8364
ISSN No. (Online) : 2249-3255

Key-aggregate crypto system for scalable data sharing in cloud storage

Prof. Vishal More* and Prof. Amit Kumar Singh**
**Department of CSE, BKIT, Bhalki VTU, Belgaum, Karnataka, INDIA*
***Department of ECE, BKIT, Bhalki VTU, Belgaum, Karnataka,*

(Corresponding author: Prof. Vishal More)

(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Data sharing is an important functionality in cloud storage. In this paper, we show how to securely, efficiently, and flexibly share data with others in cloud storage. We describe new public-key cryptosystems that produce constant-size ciphertexts such that efficient delegation of decryption rights for any set of ciphertexts are possible. The novelty is that one can aggregate any set of secret keys and make them as compact as a single key, but encompassing the power of all the keys being aggregated. In other words, the secret key holder can release a constant-size aggregate key for flexible choices of ciphertext set in cloud storage, but the other encrypted files outside the set remain confidential. This compact aggregate key can be conveniently sent to others or be stored in a smart card with very limited secure storage. We provide formal security analysis of our schemes in the standard model. We also describe other application of our schemes. In particular, our schemes give the first public-key patient-controlled encryption for flexible hierarchy, which was yet to be known.

I. INTRODUCTION

CLOUD storage is gaining popularity recently. In enterprise settings, we see the rise in demand for data outsourcing, which assists in the strategic management of corporate data. It is also used as a core technology behind many online services for personal applications. Nowadays, it is easy to apply for free accounts for email, photo album, file sharing and/or remote access, with storage size more than 30 GB (or a few dollars for more than 2 TB). Together with the current wireless technology, users can access almost all of their files and emails by a mobile phone in any corner of the world. Considering data privacy, a traditional way to ensure it is to rely on the server to enforce the access control after authentication (e.g., [1]), which means any unexpected privilege escalation will expose all data. In a shared-tenancycloud computing environment, things become even worse. Data from different clients can be hosted on separate virtual machines (VMs) but reside on a single physical machine. Data in a target VM could be stolen by instantiating another VM coresident with the target one [2]. Regarding availability of files, there are a series of cryptographic schemes which go as far as allowing a third-party auditor to check the availability of files on behalf of the data owner without leaking anything about the data [3], or without compromising the data owners anonymity [4].

Likewise, cloud users probably will not hold the strong belief that the cloud server is doing a good job in terms of confidentiality. A cryptographic solution, for example, [5], with proven security relied on number-theoretic assumptions is more desirable, whenever the user is not perfectly happy with trusting the security of the VM or the honesty of the technical staff. These users are motivated to encrypt their data with their own keys before uploading them to the server. Data sharing is an important functionality in cloud storage.

1.2 Literature Survey

1 “Privacy- Preserving Public Auditing for Secure Cloud Storage,”
Using cloud storage, users can remotely store their data and enjoy the on-demand high-quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in cloud computing a formidable task, especially for users with constrained computing resources. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity.

Thus, enabling public audit ability for cloud storage is of critical importance so that users can resort to a third-party auditor (TPA) to check the integrity of outsourced data and be worry free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities toward user data privacy, and introduce no additional online burden to user. In this paper, we propose a secure cloud storage system supporting privacy-preserving public auditing. We further extend our result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient. Our preliminary experiment conducted on Amazon EC2 instance further demonstrates the fast performance of the design.

2” Storing Shared Data on the Cloud via Security-Mediator,”

Nowadays, many organizations outsource data storage to the cloud such that a member (owner) of an organization can easily share data with other members (users). Due to the existence of security concerns in the cloud, both owners and users are suggested to verify the integrity of cloud data with Provable Data Possession (PDP) before further utilization on data. However, previous methods either unnecessarily reveal the identity of a data owner to the untrusted cloud or any public verifiers, or introduce significant overheads on verification metadata to preserve anonymity. In this paper, we propose a simple and efficient publicly verifiable approach to ensure cloud data integrity without sacrificing the anonymity of data owners nor requiring significant verification metadata. Specifically, we introduce a security-mediator (SEM), which is able to generate verification metadata (i.e., signatures) on outsourced data for data owners. Our approach decouples the anonymity protection mechanism from the PDP. Thus, an organization can employ its own anonymous authentication mechanism, and the cloud is oblivious to that since it only deals with typical PDP-metadata, Consequently, there is no extra storage overhead when compared with existing non-anonymous PDP solutions. The distinctive features of our scheme also include data privacy, such that the SEM does not learn anything about the data to be uploaded to the cloud at all, which is able to minimize the requirement of trust on the SEM. In addition, we can also extend our scheme to work with the multi-SEM model, which can avoid the potential single point of failure existing in the single-SEM scenario. Security analyses prove our scheme is secure, and experiment results demonstrate our scheme is efficient.

3. “Dynamic Secure Cloud Storage with Provenance,”

One concern in using cloud storage is that the sensitive data should be confidential to the servers which are outside the trust domain of data owners. Another issue is that the user may want to preserve his/her anonymity in the sharing or accessing of the data (such as in Web 2.0 applications). This paper addresses the problem of building a secure cloud storage system which supports dynamic users and data provenance. Previous system is based on specific constructions and does not offer all of the aforementioned desirable properties. Most importantly, dynamic user is not supported. We study the various features offered by cryptographic anonymous authentication and encryption mechanisms; and instantiate our design with verifier-local revocable group signature and identity-based broadcast encryption with constant size ciphertext and private keys. To realize our concept, we equip the broadcast encryption with the dynamic ciphertext update feature, and give formal security guarantee against adaptive chosen-ciphertext decryption and update attacks.

4. “Dynamic and Efficient Key Management for Access Hierarchies,”

Hierarchies arise in the context of access control whenever the user population can be modeled as a set of partially ordered classes (represented as a directed graph). A user with access privileges for a class obtains access to objects stored at that class and all descendant classes in the hierarchy. The problem of key management for such hierarchies then consists of assigning a key to each class in the hierarchy so that keys for descendant classes can be obtained via efficient key derivation. We propose a solution to this problem with the following properties: (1) the space complexity of the public information is the same as that of storing the hierarchy; (2) the private information at a class consists of a single key associated with that class; (3) updates (i.e., revocations and additions) are handled locally in the hierarchy; (4) the scheme is provably secure against collusion; and (5) each node can derive the key of any of its descendant with a number of symmetric-key operations bounded by the length of the path between the nodes. Whereas many previous schemes had some of these properties, ours is the first that satisfies all of them. The security of our scheme is based on pseudorandom functions, without reliance on the Random Oracle Model. Another substantial contribution of this work is that we are able to lower the key derivation time at the expense of modestly increasing the public storage associated with the hierarchy. The key derivation work for such graphs is then linear in d and the increase in the number of edges is by the factor O (log d□−□1 n) compared to the one-dimensional case.

Finally, by making simple modifications to our scheme, we show how to handle extensions proposed by Crampton [2003] of the standard hierarchies to "limited depth" and reverse inheritance.

5. "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data,"

As more sensitive data is shared and stored by third-party sites on the Internet, there will be a need to encrypt data stored at these sites. One drawback of encrypting data, is that it can be selectively shared only at a coarse-grained level (i.e., giving another party your private key). We develop a new cryptosystem for fine-grained sharing of encrypted data that we call Key-Policy Attribute-Based Encryption (KP-ABE). In our cryptosystem, ciphertexts are labeled with sets of attributes and private keys are associated with access structures that control which ciphertexts a user is able to decrypt.

II. AIM AND OBJECTIVE

The main objective of the any citation analysis is to provide useful information to the scholars in searching of literature and to help the librarian in selecting relevant sources. The specific objectives of the present study are to find out the information sources cited by researchers in the field of Key-aggregate crypto system for scalable data sharing in cloud storage,

(i) To find-out the authorship pattern and degree of collaboration in aggregate key for flexible choices,
(ii) To identify the core cryptosystem,
(iii) To examine the subject wise break up of citations,
(iv) To know the country wise distribution of CLOUD storage
(v) To identify the language wise distribution of journal citations.

III. DETAILED DESIGN

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software. The importance can be stated with a single word "Quality". Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer's view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

A. System Study

Feasibility Study. The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

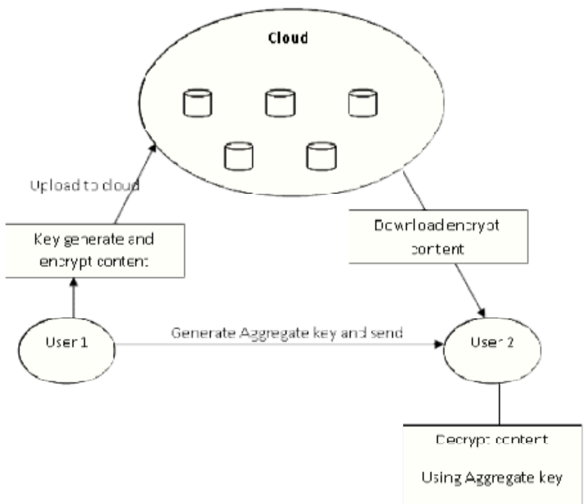
TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it.

IV. BLOCK DIAGRAM



A. Data Flow Diagram

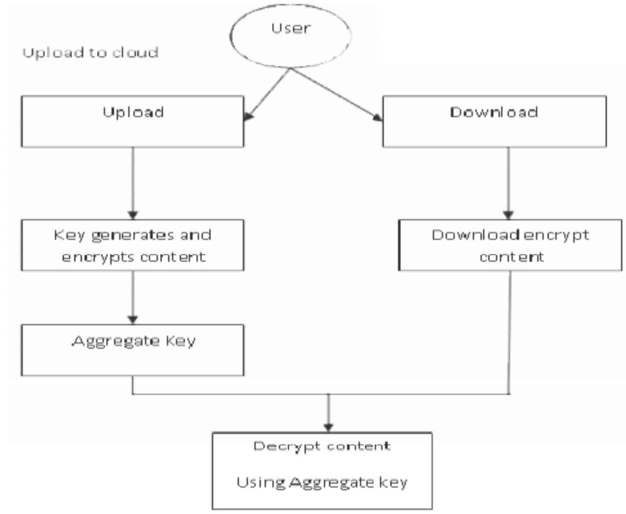
1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

B. UML Diagrams

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.



The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

4.3 GOALS:The Primary goals in the design of the UML are as follows:

1.Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

2.Provide extendibility and specialization mechanisms to extend the core concepts.

3.Be independent of particular programming languages and development process.

4.Provide a formal basis for understanding the modeling language.

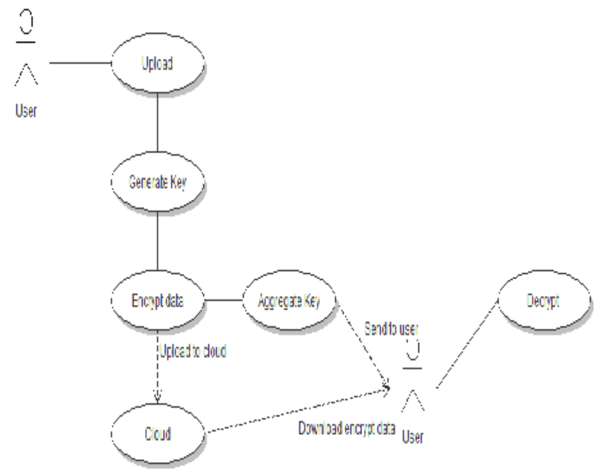
5.Encourage the growth of OO tools market.

6.Support higher level development concepts such as collaborations, frameworks, patterns and components.

7.Integrate best practices.

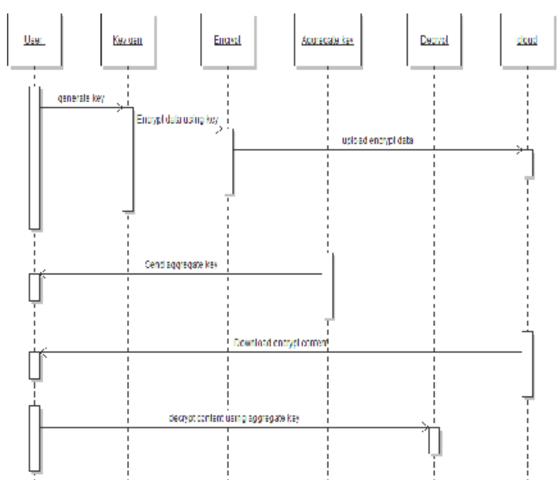
D. Use Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



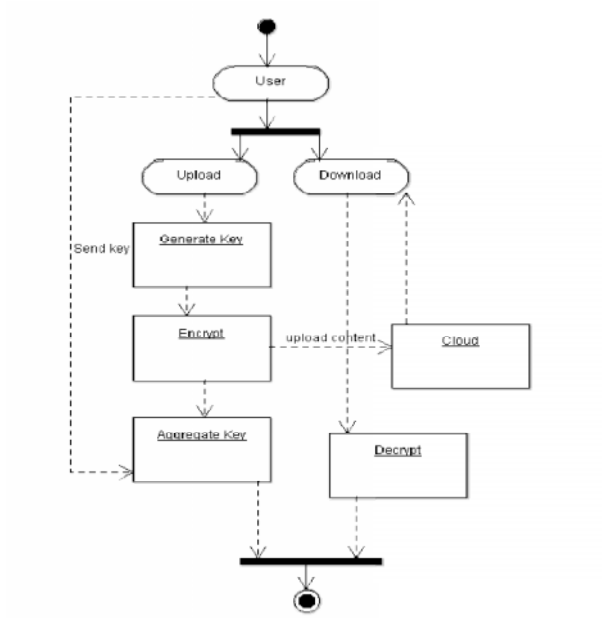
SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



VI. CONCLUSION

How to protect users' data privacy is a central question of cloud storage. With more mathematical tools, cryptographic schemes are getting more versatile and often involve multiple keys for a single application. In this paper, we consider how to "compress" secret keys in public-key cryptosystems which support delegation of secret keys for different cipher text classes in cloud storage. No matter which one among the power set of classes, the delegate can always get an aggregate key of constant size. Our approach is more flexible than hierarchical key assignment which can only save spaces if all key-holders share a similar set of privileges. A limitation in our work is the predefined bound of the number of maximum cipher text classes. In cloud storage, the number of cipher texts usually grows rapidly. So we have to reserve enough cipher text classes for the future extension. Although the parameter can be downloaded with cipher texts, it would be better if its size is independent of the maximum number of cipher text classes. On the other hand, when one carries the delegated keys around in a mobile device without using special trusted hardware, the key is prompt to leakage, designing a leakage-resilient cryptosystem [22], [34] yet allows efficient and flexible key delegation is also an interesting direction.

REFERENCES

[1] S.S.M. Chow, Y.J. He, L.C.K. Hui, and S.-M. Yiu, "SPICE – Simple Privacy-Preserving Identity-Management for Cloud Environment," Proc. 10th Int'l Conf. Applied Cryptography and Network Security (ACNS), vol. 7341, pp. 526-543, 2012.
[2] L.Hardesty, Secure Computers Aren't so Secure. MIT press, http://www.physorg.com/news176107396.html, 2009.
[3] C. Wang, S.S.M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Trans. Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013.
[4] B. Wang, S.S.M. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," Proc. IEEE 33rd Int'l Conf. Distributed Computing Systems (ICDCS), 2013.
[5] S.S.M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R.H. Deng, "Dynamic Secure Cloud Storage with Provenance," Cryptography and Security, pp. 442-464, Springer, 2012.
[6] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," Proc. 22nd Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT '03), pp. 416-432, 2003.
[7] M.J. Atallah, M. Blanton, N. Fazio, and K.B. Frikken, "Dynamic and Efficient Key Management for Access Hierarchies," ACM Trans. Information and System Security, vol. 12, no. 3, pp. 18:1-18:43, 2009.
[8] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records," Proc. ACM Workshop Cloud Computing Security (CCSW '09), pp. 103-114, 2009.
[9] F. Guo, Y. Mu, Z. Chen, and L. Xu, "Multi-Identity Single-Key Decryption without Random Oracles," Proc. Information Security and Cryptology (Inscrypt '07), vol. 4990, pp. 384-398, 2007.
[10] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," Proc. 13th ACM Conf. Computer and Comm. Security (CCS '06), pp. 89-98, 2006.
[11] S.G. Akl and P.D. Taylor, "Cryptographic Solution to a Problem of Access Control in a Hierarchy," ACM Trans. Computer Systems, vol. 1, no. 3, pp. 239-248, 1983.
[12] G.C. Chick and S.E. Tavares, "Flexible Access Control with Master Keys," Proc. Advances in Cryptology (CRYPTO '89), vol. **435**, pp. 316-322, 1989.
[13] W.-G. Tzeng, "A Time-Bound Cryptographic Key Assignment Scheme for Access Control in a Hierarchy," IEEE Trans. Knowledge and Data Eng., vol. 14, no. 1, pp. 182-188, Jan./Feb. 2002.
[14] G. Ateniese, A.D. Santis, A.L. Ferrara, and B. Masucci, "Provably-Secure Time-Bound Hierarchical Key Assignment Schemes," J. Cryptology, vol. 25, no. 2, pp. 243-270, 2012.
[15] R.S. Sandhu, "Cryptographic Implementation of a Tree Hierarchy for Access Control," Information Processing Letters, vol. 27, no. 2, pp. 95-98, 1988.
[16] Y. Sun and K.J.R. Liu, "Scalable Hierarchical Access Control in Secure Group Communications," Proc. IEEE INFOCOM '04, 2004.
[17] Q. Zhang and Y. Wang, "A Centralized Key Management Scheme for Hierarchical Access Control," Proc. IEEE Global Telecomm. Conf. (GLOBECOM '04), pp. 2067-2071, 2004.
[18] J. Benaloh, "Key Compression and Its Application to Digital Fingerprinting," technical report, Microsoft Research, 2009.
[19] B. Alomair and R. Poovendran, "Information Theoretically Secure Encryption with Almost Free Authentication," J. Universal Computer Science, vol. 15, no. 15, pp. 2937-2956, 2009.
[20] D. Boneh and M.K. Franklin, "Identity-Based Encryption from the Weil Pairing," Proc. Advances in Cryptology (CRYPTO '01), vol. **2139**, pp. 213-229, 2001.
[21] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," Proc. 22nd Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT '05), vol. 3494, pp. 457-473, 2005.
[22] S.S.M. Chow, Y. Dodis, Y. Rouselakis, and B. Waters, "Practical Leakage-Resilient Identity-Based Encryption from Simple Assumptions," Proc. ACM Conf. Computer and Comm. Security, pp. 152-161, 2010.
[23] F. Guo, Y. Mu, and Z. Chen, "Identity-Based Encryption: How to Decrypt Multiple Ciphertexts Using a Single Decryption Key," Proc. Pairing-Based Cryptography Conf. (Pairing '07), vol. 4575, pp. 392-406, 2007.
[24] M. Chase and S.S.M. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption," Proc. ACM Conf. Computer and Comm. Security, pp. 121-130. 2009,
[25] T. Okamoto and K. Takashima, "Achieving Short Ciphertexts or Short Secret-Keys for Adaptively Secure General Inner-Product Encryption," Proc. 10th Int'l Conf. Cryptology and Network Security (CANS '11), pp. 138-159, 2011.
[26] R. Canetti and S. Hohenberger, "Chosen-Ciphertext Secure Proxy Re-Encryption," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 185-194, 2007.
[27] C.-K. Chu and W.-G. Tzeng, "Identity-Based Proxy Re-encryption without Random Oracles," Proc. Information Security Conf. (ISC '07), vol. **4779**, pp. 189-202, 2007.
[28] C.-K. Chu, J. Weng, S.S.M. Chow, J. Zhou, and R.H. Deng, "Conditional Proxy Broadcast Re-Encryption," Proc. 14th Australasian Conf. Information Security and Privacy (ACISP '09), vol. **5594**, pp. 327342, 2009.
[29] S.S.M. Chow, J. Weng, Y. Yang, and R.H. Deng, "Efficient Unidirectional Proxy Re-Encryption," Proc. Progress in Cryptology (AFRICACRYPT '10), vol. **6055**, pp. 316-332, 2010.
[30] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," ACM Trans. Information and System Security, vol. **9**, no. 1, pp. 1-30, 2006.
[31] D. Boneh, C. Gentry, and B. Waters, "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys," Proc. Advances in Cryptology Conf. (CRYPTO '05), vol. **3621**, pp. 258-275, 2005.
[32] L.B. Oliveira, D. Aranha, E. Morais, F. Daguano, J. Lopez, and R. Dahab, "TinyTate: Computing the Tate Pairing in Resource-Constrained Sensor Nodes," Proc. IEEE Sixth Int'l Symp. Network Computing and Applications (NCA '07), pp. 318-323, 2007.