Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage

**Literature survey**

**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 auditability 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 towards 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.

In this paper, we propose a privacy-preserving public auditing system for data storage security in Cloud Computing. We utilize the homomorphic linear authenticator and random masking to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users’ fear of their outsourced data leakage. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, we further extend our privacy-preserving public auditing protocol into a multi-user setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. Extensive analysis shows that our schemes are provably secure and highly efficient.

**Storing Shared Data on the Cloud via Security-Mediator**

Nowadays, many organizations outsource data storage to the cloud such that a member of an organization (data owner) 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 of 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 for preserving anonymity. In this paper, we propose a simple, efficient, and publicly verifiable approach to ensure cloud data integrity without sacrificing the anonymity of data owners nor requiring significant overhead. 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, the identity of the data owner is not revealed to the cloud, and there is no extra storage overhead unlike existing 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, and thus the trust on the SEM is minimized. In addition, we extend our scheme to work with the multi-SEM model, which can avoid the potential single point of failure. Security analyses prove that our scheme is secure, and experiment results demonstrate that our scheme is efficient.

In this paper, we introduce what we believe is the right approach to achieve anonymity in storing data to the cloud with publicly-verifiable data-integrity in mind. Our approach decouples the anonymous protection mechanism from the provable data possession mechanism via the use of security mediator. Our solution not only minimizes the computation and bandwidth requirement of this mediator, but also minimizes the trust placed on it in terms of data privacy and identity privacy. The efficiency of our system is also empirically demonstrated.

**Aggregate and Verifiably Encrypted Signatures from Bilinear Maps**

An aggregate signature scheme is a digital signature that supports aggregation: Given n signatures on n distinct messages from n distinct users, it is possible to aggregate all these signatures into a single short signature. This single signature (and the n original messages) will convince the verifier that the n users did indeed sign the n original messages (i.e., user I signed message Mi for i = 1; : : : ; n). In this paper we introduce the concept of an aggregate signature, present security models for such signatures, and give several applications for aggregate signatures. We construct an efficient aggregate signature from a recent short signature scheme based on bilinear maps due to Boneh, Lynn, and Shacham. Aggregate signatures are useful for reducing the size of certificate chains (by aggregating all signatures in the chain) and for reducing message size in secure routing protocols such as SBGP. We also show that aggregate signatures give rise to verifiably encrypted signatures. Such signatures enable the verifier to test that a given ciphertext C is the encryption of a signature on a given message M. Verifiably encrypted signatures are used in contract-signing protocols. Finally, we show that similar ideas can be used to extend the short signature scheme to give simple ring signatures.

We introduced the concept of aggregate signatures and constructed an efficient aggregate signature scheme based on bilinear maps. Key generation, aggregation, and verification require no interaction. We proved security of the system in a model that gives the adversary his choice of public keys and messages to forge. F or security, we introduced the additional constraint that an aggregate signature is valid only if it is an aggregation of signatures on distinct messages. This constraint is satisfied naturally for the applications we have in mind. More generally, the constraint can be satisfied by pre pending the public key to the message prior to signing. We gave several applications for aggregate signatures. F or example, they can be used to reduce the size of certificate chains and reduce communication bandwidth in protocols such as SBGP. We also showed that our specific aggregate signature scheme gives verifiably encrypted signatures. Previous signature constructions using bilinear maps only required a gap Diffie Hellman group (i.e., DDH easy, but CDH hard). The signature constructions in this paper require the extra structure provided by the bilinear map. These constructions are an example where a bilinear map provides more power than a generic gap Diffie-Hellman group.

**Dynamic and Efficient Key Management for Access Hierarchies**

The problem of key management in an access hierarchy has elicited much interest in the literature. The hierarchy is modeled as a set of partially ordered classes (represented as a directed graph), and a user who obtains access (i.e., a key) to a certain class can also obtain access to all descendant classes of her class through key derivation. Our solution to the above problem has the following properties: (i) only hash functions are used for a node to derive a descendant’s key from its own key; (ii) the space complexity of the public information is the same as that of storing the hierarchy; (iii) the private information at a class consists of a single key associated with that class; (iv) updates (revocations, additions, etc.) are handled locally in the hierarchy; (v) the scheme is provably secure against collusion; and (vi) key derivation by a node of its descendant’s key is bounded by the number of bit operations linear in 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. Moreover, for trees (and other “recursively decomposable” hierarchies), we are the first to achieve a worst- and average-case number of bit operations for key derivation that is exponentially better than the depth of a balanced hierarchy (double-exponentially better if the hierarchy is unbalanced, i.e., “tall and skinny”); this is achieved with only a constant increase in the space for the hierarchy. We also show how with simple modifications our scheme can handle extensions proposed by Crampton of the standard hierarchies to “limited depth” and reverse inheritance [13]. The security of our scheme relies only on the use of pseudo-random functions.

In summary, we give the first solution to the problem of access control in an arbitrary hierarchy G with the following properties: 1. Only hash functions are used for a node to derive a descendant’s key from its own key; 2. The space complexity of the public information is the same as that of storing graph G; 3. The derivation by a node of a descendant’s access key requires O(`) bit operations, where ` is the length of the path between the nodes, for arbitrary hierarchies and log log n or less for trees; 4. Updates are handled locally and do not “propagate” to descendants or ancestors of the affected part of G; 5. The scheme is resistant to collusion in that no subset of nodes can conspire to gain access to any node access to which they cannot legally obtain; 6. The private information at a node consists of a single key associated with that node. We also provided simple modifications to our scheme that allow to handle Crampton’s extensions of the standard hierarchies to “limited depth” and reverse inheritance [13], and gave shortcut schemes that permit to significantly reduce key derivation time for tree hierarchies.

**Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records**

We explore the challenge of preserving patients' privacy in electronic health record systems. We argue that security in such systems should be enforced via encryption as well as access control. Furthermore, we argue for approaches that enable patients to generate and store encryption keys, so that the patients' privacy is protected should the host data center be compromised. The standard argument against such an approach is that encryption would interfere with the functionality of the system. However, we show that we can build an efficient system that allows patients both to share partial access rights with others, and to perform searches over their records. We formalize the requirements of a Patient Controlled Encryption scheme, and give several instantiations, based on existing cryptographic primitives and protocols, each achieving a different set of properties.

We have presented several schemes for Patient Controlled Encryption, each appropriate for a different setting. Table 1 summarizes the advantages and disadvantages of these schemes. For a concrete design, we suggest that one follow the set-up described in Section 3. We conclude that it is possible and practical to achieve secure and private EMR while maintaining efficiency and functionality, including searchability and delegation.

**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, cipher texts are labeled with sets of attributes and private keys are associated with access structures that control which cipher texts a user is able to decrypt. We demonstrate the applicability of our construction to sharing of audit-log information and broadcast encryption. Our construction supports delegation of private keys which subsumes Hierarchical Identity-Based Encryption (HIBE).

Achieving CCA-Security and HIBE from Delegation We briefly outline how we can achieve efficient CCA-2 security and realize the Hierarchical Identity-Based Encryption by applying delegation techniques to the large universe construction. To achieve CCA-2 security an encyrptor will chooses a set ° of attributes to encrypt the message under and then generate a public/private key pair for a one time signature scheme. We let VK denote the bit string representation of the public key and let 0 be the set ° [VK. The encryptor encrypts the ciphertext under the attributes ° 0 and then signs the ciphertext with the private key and attaches the signature and the public key description. Suppose a user has a key for access structure X wishes to decrypt. The user first checks that the ciphertext is signed under VK and rejects the ciphertext otherwise. Then it creates an new key for the access structure of \X AND CCA : VK". By similar arguments to those in Canetti, Halevi, and Katz this gives chosen-ciphertex security. We can also use other methods to achieve greater efficiency.

**Cryptographic Solution to a Problem of Access Control in a Hierarchy**

A scheme based on cryptography is proposed for access control in a system where hierarchy is represented by a partially ordered set (or poset). Straightforward implementation of the scheme requires users highly placed in the hierarchy to store a large number of cryptographic keys. A time- versus-storage trade-off is then described for addressing this key management problem. Categories and Subject DescriPtors: D.4.6 [Operating Systems]: Security and Protection--access controls; authentication; cryptographic controls; information flow controls; E.3 [Data]: Data Encryption--Data Encryption Standard (DES); public-key cryptosystems.

We have described a scheme based on cryptography for controlling access to data in an organization where hierarchy is represented by a partially ordered set. The scheme enables a member of the organization at some level of the hierarchy to derive from his own cryptographic key the keys of members below him in the hierarchy, and consequently to have access to information enciphered under those keys. One interesting feature of the scheme is that the protection it offers against illegal disclosure depends neither on the physical security of the storage medium where the information resides nor on the trustworthiness of the people managing it. Furthermore, this protection does not apply only to files that are stored in a central computer memory, but also to messages broadcast on a communication network using telephone lines or radio waves. Anyone with the proper receiving equipment can intercept the message but has access to the information it contains only if in possession of the right key. These two properties, which distinguish the scheme from other solutions to the problem of access control in a hierarchy, are clearly due to the use of cryptography. Another important property of the scheme is that it provides security against two or more users of the system collaborating to compute a key to which they are not entitled. It is not difficult to conceive of examples of hierarchies where such access control is required. As an illustration, consider the personnel of a chain of department stores. Employees are grouped by their rank into classes forming a poset. Here the piece of information to be broadcast may, for instance, be the date of release on the market of the latest brand of a particular product. The problem arises in this case if top management desires to make these data available to all employees at or above the level of store manager, say, but not lower. Another example is a hospital where only doctors with a certain degree of seniority may have access to some personal information in a patient's medical record. Similar situations abound in other areas, particularly in the government and the military, and are easily envisaged. In all these cases a scheme such as ours may contribute a convenient solution to the problem of access control in a hierarchy. Finally, the scheme could also be useful in a secure distributed system where hosts, operating at different security levels, communicate via an untrusted communication sub network (csn). Encrypted messages are broadcast into the csn without concern for misrouting by untrusted csn software since unintended recipients would be unable to decrypt them. Such a use does not, of course, address problems of information flows that are based on some type of message stream modulation.