EFFICIENT AND EXPRESSIVE KEYWORD SEARCH OVER ENCRYPTED DATA IN CLOUD

ZEESHAN SHARIEF

MASTER OF COMPUTER APPPLICATIONS, JAIN-SCMS, BANGALORE-560069

ABSTRACT - Searchable encryption allows a cloud server to conduct keyword search over encrypted data on behalf of the data users without learning the underlying plaintexts. However, most existing searchable encryption schemes only support single or conjunctive keyword search, while a few other schemes that can perform expressive keyword search are computationally inefficient since they are built from bilinear pairings over the composite-order groups. In this paper, we propose an expressive public-key searchable encryption scheme in the prime-order groups, which allows keyword search policies i.e., predicates, access structures to be expressed in conjunctive, disjunctive or any monotonic Boolean formulas and achieves significant performance improvement over existing schemes. We formally define its security and prove that it is selectively secure in the standard model. Also, we implement the proposed scheme using a rapid prototyping tool called Charm and conduct several experiments to evaluate it performance. The results demonstrate that our scheme is much more efficient than the ones built over the composite-order groups.

INDEX TERMS - Searchable encryption, cloud computing, expressiveness, attribute-based encryption.

OVERVIEW

Our expressive SE scheme consists of a trusted trapdoor generation center which publishes a public system parameter and keeps a master key in secret, a cloud server which stores and searches encrypted data on behalf of data users, multiple data owners who upload encrypted data to the cloud, and multiple data users who would like to retrieve encrypted data containing certain keywords. To outsource an encrypted document to the cloud, a data owner appends the encrypted document with keywords encrypted under the public parameter and uploads the combined encrypted document and encrypted keywords to the cloud. To retrieve all the encrypted documents containing keywords satisfying a certain access structure (i.e., predicate or policy) such as ("Illness = Diabetes" AND ("Age = 30" OR "Weight = 150200")), a data user first obtains a trapdoor associated with the access structure from the trapdoor generation center and then sends the trapdoor to the cloud server. The latter will conduct the search and return the corresponding encrypted documents to the data user. The basic idea of our scheme is to modify a key-policy attributed-based encryption (KP-ABE) scheme constructed from bilinear pairing over prime-order groups. Without loss of generality, we will use the large universe KP-ABE scheme selectively secure in the standard model proposed by Rouselakis and Waters in to illustrate our construction during the rest of the paper.

In KP-ABE, a ciphertext is computed with respect to a set of attributes and an access policy is encoded into a user's private key. A ciphertext can be decrypted by a private key only if the set of attributes associated with the ciphertext satisfies the access policy associated with the private key. Access policies in can be very expressive, supporting any monotonic Boolean formulas. At first sight, a KP-ABE scheme can be transformed to an expressive SE scheme by treating attributes as keywords to be searched, by directly transforming the key generation algorithm on attribute access structures to a trapdoor generation algorithm on keyword search predicates, and by using the decryption algorithm to test whether keywords in a ciphertext satisfy the predicate in a trapdoor. However, KP-ABE schemes are not designed to preserve privacy of attributes (keywords) associated with ciphertexts. Specifically, given the public parameter and a ciphertext, the attributes (keywords) in the ciphertext can be discerned by anyone. In the following, to keep our description compact and consistent, we will use access structure, policy and predicate interchangeably. In order to hide keywords in a ciphertext, inspired by the "linear splitting" technique, we firstly split ciphertext components corresponding to every keyword into two randomized complementary components.

Thus, even though the ciphertext still contains information about the keywords, this information is computationally infeasible to obtain from the public parameter and the ciphertext corresponding to every keyword associated with an access structure to match the split components in the ciphertext. In addition to hiding keywords in ciphertexts, we also need to preserve keyword privacy in a trapdoor which contains an access structure as a component. First, to preserve keyword privacy in an access structure, we adopt the method in to divide each keyword into a generic name and a keyword value. Since keyword values are much more sensitive than the generic keyword names, the keyword values in an access structure are not disclosed to the cloud server, whereas a partial hidden access structure with only generic keyword names is included in a trapdoor and sent to the cloud server. Take the keyword access structure ("Illness = Diabetes" AND ("Age = 30" OR "Weight = 150-200")) as an instance, "Illness", "Age" and "Weight" are the generic names whilst "Diabetes", "30" and "200" are the keyword values. Consequently, the partial hidden access structure ("Illness" AND "Age" OR "Weight") is included in the trapdoor. Second, as in all the PEKS schemes, trapdoors are subject to the offline keyword dictionary guessing attacks. That is, anyone who knows a trapdoor and the public parameter may discover the keyword values embedded in the trapdoor by launching exhaustive searching attacks on keyword values. As a remedy to such attacks, we assign a designated cloud server to perform the searching operations. We equip this designated server with a public and private key pair of which the public key will be used in trapdoor generation such that it is computationally infeasible for anyone without knowledge of the privacy key to derive keywords information from the trapdoor. Thus, trapdoors can be delivered to the cloud server over a public channel. We define a security model for expressive SE, which considers all adversarial capabilities of the standard SE security notion. The adversary can learn trapdoors over access structures of its choice, but it should not be able to learn any information about the keyword values in the challenge ciphertext. Note that since the Rouselakis-Waters KP-ABE scheme, which the proposed SE scheme is built upon, is selectively secure, our expressive SE scheme can only be proved to be selectively secure where the adversary must commit the challenge keyword set in advance. Ideally, in the practical applications, search predicates (i.e., policies) should be expressive such that they can be expressed as conjunction, disjunction or any Boolean formulas2 of keywords. In the above cloud-based healthcare system, to find the relationship between diabetes and age or weight, a medical researcher may issue a search query with an access structure.

We assume that the trapdoor generation center is a trusted entity. The cloud server is assumed to be "honest-but curious", i.e., it will honestly follow the protocol, but it is also curious to learn any private information from the data stored in the cloud. Data owners are assumed to honestly store their data, while data users are not trusted, and they can even collude with a malignant cloud server in order to discover private information of other parties. We assume that the trusted trapdoor generation center is equipped with a separate authentication mechanism to verify data users before issuing trapdoors to users. Also, we assume that all adversaries have bounded computational capability, so they cannot break the difficult problems. The scheme uses a rapidly prototyping tool called Charm and conduct extensive experiments to evaluate its performance. Our results confirm that the proposed scheme is sufficiently efficient to be applied in practice. Using a randomness splitting technique, our scheme achieves security against offline keyword dictionary guessing attacks to the ciphertexts. Moreover, to preserve the privacy of keywords against offline keyword dictionary guessing attacks to trapdoors, we divide each keyword into keyword name and keyword value and assign a designated cloud server to conduct search operations in our construction.

REFERENCES

[1] O. Goldreich and R. Ostrovsky, "Software protection and simulation on oblivious rams," J. ACM, vol. 43, no. 3, pp. 431–473, 1996.

[2] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in 2000 IEEE Symposium on Security and Privacy, Berkeley, California, USA, May 14-17, 2000. IEEE Computer Society, 2000, pp.44–55.

[3] E. Goh, "Secure indexes," IACR Cryptology ePrint Archive, vol. 2003, p. 216, 2003.