SecRBAC: Secure data in the Clouds

Dr. K. Rameshwaraiah*1  G. Shiva Prasad*2

*1 Professor Department of Computer Science and Engineering  *2 M.Tech
Nalla Narasimha Reddy Educational Society’s Group of Institution

Abstract—Most current security solutions are based on perimeter security. However, Cloud computing breaks the organization perimeters. When data resides in the cloud, they reside outside the organizational bounds. This leads users to a loss of control over their data and raises reasonable security concerns that slow down the adoption of Cloud computing. Is the Cloud service provider accessing the data? Is it legitimately applying the access control policy defined by the user? This paper presents a data-centric access control solution with enriched role-based expressiveness in which security is focused on protecting user data regardless the Cloud service provider that holds it. Novel identity-based and proxy re-encryption techniques are used to protect the authorization model. Data is encrypted and authorization rules are cryptographically protected to preserve user data against the service provider access or misbehavior. The authorization model provides high expressiveness with role hierarchy and resource hierarchy support. The solution takes advantage of the logic formalism provided by Semantic Web technologies, which enables advanced rule management like semantic conflict detection. A proof of concept implementation has been developed and a working prototypical deployment of the proposal has been integrated within Google services.

Keywords—Data-centric security, Cloud computing, Role-based access control, Authorization.

I. INTRODUCTION

Security is one of the main user concerns for the adoption of Cloud computing. Moving data to the Cloud usually implies relying on the Cloud Service Provider (CSP) for data protection. Although this is usually managed based on legal or Service Level Agreements (SLA), the CSP could potentially access the data or even provide it to third parties. Moreover, one should trust the CSP to legitimately apply the access control rules defined by the data owner for other users. The problem becomes even more complex in Inter-cloud scenarios where data may flow from one CSP to another. Users may lose control on their data. Even the trust on the federated CSPs is outside the control of the data owner. This situation leads to rethink about data security approaches and to move to a data-centric approach where data are self-protected whenever they reside.

Encryption is the most widely used method to protect data in the Cloud. In fact, the Cloud Security Alliance security guidance recommends data to be protected at rest, in motion and in use [1]. Encrypting data avoids undesired accesses. However, it entails new issues related to access control management. A rule-based approach would be desirable to provide expressiveness. But this supposes a big challenge for a data-centric approach since data has no computation capabilities by itself. It is not able to enforce or compute any access control rule or policy. This raises the issue of policy decision for a self-protected data package: who should evaluate the rules upon an access request? The first choice would be to have them evaluated by the CSP, but it could potentially bypass the rules. Another option would be to have rules evaluated by the data owner, but this implies that either data could not be shared or the owner should be online to take a decision for each access request. To overcome the aforementioned issues, several proposals [2] [3] [4] try to provide data-centric solutions based on novel cryptographic mechanisms applying Attribute-based Encryption (ABE) [5]. These solutions are based on Attribute-based Access Control (ABAC), in which privileges are granted to users according to a set of attributes. There is a long standing debate in the IT community about whether Role-based Access Control (RBAC) [6] or ABAC is a better model for authorization [7] [8] [9]. Without entering into this debate, both approaches have their own pros and cons. To the best of our knowledge, there is no data-centric approach providing an RBAC model for access control in which data is encrypted and self-protected. The proposal in this paper supposes a first solution for a data-centric RBAC approach, offering an alternative to the ABAC model. An RBAC approach would be closer to current access control methods, resulting more natural to apply for access control enforcement than ABE-based mechanisms. In terms of expressiveness, it is said that ABAC supersedes RBAC since roles can be represented as attributes. However, when it comes to data-centric approaches in which data is encrypted, ABAC solutions are constrained by the expressiveness of ABE schemes.
The cryptographic operations used in ABE usually restrict the level of expressiveness for access control rules. For instance, role hierarchy and object hierarchy capabilities cannot be achieved by current ABE schemes. Moreover, they usually lack some combination with a user-centric approach for the access control policy, where common authorization-related elements like definition of users or role assignments could be shared by different pieces of data from the same data owner.

This paper presents SecRBAC, a data-centric access control solution for self-protected data that can run in untrusted CSPs and provides extended Role-Based Access Control expressiveness. Hierarchies are supported by the authorization model, providing more expressiveness to the rules by enabling the definition of simple but powerful rules that apply to several users and resources thanks to privilege propagation through roles and hierarchies. Policy rule specifications are based on Semantic Web technologies that enable enriched rule definitions and advanced policy management features like conflict detection. A data-centric approach is used for data self-protection, where novel cryptographic techniques such as Proxy Re-Encryption Encryption (PRE) [10], Identity-Based Encryption (IBE) [11] and Identity-Based Proxy Re-Encryption (IBPRE) [12] are used. They allow to re-encrypt data from one key to another without getting access and to use identities in cryptographic operations. These techniques are used to protect both the data and the authorization model. Each piece of data is ciphered with its own encryption key linked to the authorization model and rules are cryptographically protected to preserve data against the service provider access or misbehavior when evaluating the rules. It also combines a user-centric approach for authorization rules, where the data owner can define a unified access control policy for his data. The solution enables a rule-based approach for authorization in Cloud systems where rules are under control of the data owner and access control computation is delegated to the CSP, but making it unable to grant access to unauthorized parties. The main contributions of the proposed solution are:

- Data-centric solution with data protection for the Cloud Service Provider to be unable to access it.
- Rule-based approach for authorization where rules are under control of the data owner.
- High expressiveness for authorization rules applying the RBAC scheme with role hierarchy and resource hierarchy (Hierarchical RBAC or hRBAC).
- Access control computation delegated to the CSP, but being unable to grant access to unauthorized parties.
- Secure key distribution mechanism and PKI compatibility for using standard X.509 certificates and keys. The rest of this paper is structured as follows. Section 2 presents some state of the art in authorization approaches for Cloud. Section 3 introduces the cryptographic techniques used in this proposal. The authorization model with enriched RBAC expressiveness is formalized in Section 4. The data-centric model is specified in Section 5.1. Section 6 provides some deployment guidelines. Statistical results and implementation details are exposed in Section 7. Finally, the paper concludes with some remarks and statements of direction in Section 8.

II. RELATED WORK

Different approaches can be found in the literature to retain control over authorization in Cloud computing. In [13] authors propose to keep the authorization decisions taken by the data owner. The access model is not published to the Cloud but kept secure on the data owner premises. However, in this approach the CSP becomes a mere storage system and the data owner should be online to process access requests from users. Another approach from [14] deals with this issue by enabling a plug-in mechanism in the CSP that allows data owners to deploy their own security modules. This permits to control the authorization mechanisms used within a CSP. However, it does not establish how the authorization model should be protected, so the CSP could potentially infer information and access the data. Moreover, this approach does not cover Inter-cloud scenarios, since the plug-in module should be deployed to different CSPs. Additionally, these approaches do not protect data with encryption methods. In the proposed SecRBAC solution, data encryption is used to prevent the CSP to access the data or to release it bypassing the authorization mechanism. However, applying data encryption implies additional challenges related to authorization expressiveness. Following a straightforward approach, one can include data in a package encrypted for the intended users. This is usually done when sending a file or document to a specific receiver and ensures that only the receiver with the appropriate key is able to decrypt it. From an authorization point of view, this can be seen as a simple rule where only the user with privilege to access the data will be able to decrypt it (i.e. the one owning the key). However, no access control expressiveness is provided by this approach. Only that simple rule can be enforced and just one single rule can apply to each data package. Thus, multiple encrypted copies should be created in order to deliver the same data to different receivers. To cope with...
these issues, SecRBAC follows a data-centric approach that is able to cryptographically protect the data while providing access control capabilities. Several data-centric approaches, mostly based on Attribute-based Encryption (ABE) [5], have arisen for data protection in the Cloud [4]. In ABE, the encrypted ciphertext is labeled with a set of attributes by the data owner. Users also have a set of attributes defined in their private keys. They would be able to access data (i.e., decrypt it) or not depending on the match between ciphertext and key attributes. The set of attributes needed by a user to decrypt the data is defined by an access structure, which is specified as a tree with AND and OR nodes. There are two main approaches for ABE depending on where the access structure resides: Key-Policy ABE (KP-ABE) [5] and Ciphertext-Policy ABE (CP-ABE) [3]. In KP-ABE, the access structure or policy is defined within the private keys of users. This allows to encrypt data labeled with attributes and then control the access to such data by delivering the appropriate keys to users. However, in this case the policy is really defined by the key issuer instead of the encryptor of data, i.e., the data owner. So, the data owner should trust the key issuer for this to properly generate an adequate access policy. To solve this issue, CP-ABE proposes to include the access structure within the ciphertext, which is under control of the data owner. Then, the key issuer just asserts the attributes of users by including them in private keys. However, either in KP-ABE or CP-ABE, the expressiveness of the access control policy is limited to combinations of AND-ed and OR-ed attributes. The data-centric solution presented in this paper goes a step forward in terms of expressiveness, providing a rule-based approach following the RBAC scheme that is not tied to the limitations of current ABE approaches.

Different proposals have been also developed to try to alleviate ABE expressiveness limitations. Authors in [15] propose a solution based on CP-ABE with support for sets of attributes called Ciphertext Policy Attribute Set Based Encryption (CP-ASBE). Attributes are organized in a recursive set structure and access policies can be defined upon a single set or combining attributes from multiple sets. This enables the definition of compound attributes and specification of policies that affect the attributes of a set. An approach named Hierarchical Attribute-based Encryption is presented in [16]. It uses a hierarchical generation of keys to achieve fine-grain access control, scalability and delegation. However, this approach implies that attributes should be managed by the same root domain authority. In [17], authors extend CP-ASBE with a hierarchical structure to users in order to improve scalability and flexibility. This approach provides a hierarchical solution for users within a domain, which is achieved by a hierarchical key structure. Another approach is Flexible and Efficient Access Control Scheme (FEACS) [2]. It is based on KP-ABE and provides an access control structure represented by a formula involving AND, OR and NOT, enabling more expressiveness for KP-ABE.

The aforementioned ABE-based solutions proposed for solving access control in Cloud computing are based on the Attribute-based Access Control (ABAC) model. As commented in Section 1, both ABAC and RBAC models have their own advantages and disadvantages [7] [9]. On one hand, RBAC may require the definition of a large number of roles for fine-grain authorization (role explosion problem in RBAC). ABAC is also easier to set up without need to make an effort on role analysis as needed for RBAC. On another hand, ABAC may result in a large number of rules since a system with n attributes would have up to 2n possible rule combinations (rule explosion problem in ABAC). ABAC separates authorization rules from user attributes, making it difficult to determine permissions available to a particular user, while RBAC is deterministic and user privileges can be easily determined by the data owner.

Moreover, the cryptographic operations used in ABE approaches usually restrict the level of expressiveness provided by the access control rules. Concretely, role hierarchy and object hierarchy capabilities provided by SecRBAC cannot be achieved by current ABE schemes. Moreover, private keys in ABE should contain the attributes of the user, which tightens the keys to permissions in the access control policy. In SecRBAC, user keys only identify their holders and they are not tied to the authorization model. That is, user privileges are completely independent of their private key. Finally, no user-centric approach for authorization rules is provided by current ABE solutions. In SecRBAC, a single access policy defined by the data owner is able to protect more than one piece of data, resulting in a user-centric approach for rule management. Additionally, the proposed solution provides support for the ontological representation of the authorization model, providing additional reasoning mechanisms to cope with issues such as detection of conflicts between different authorization rules.
III. PROXY RE-ENCRYPTION AND IDENTITY-BASED ENCRYPTION

SecRBAC makes use of cryptography to protect data when moved to the Cloud. Advanced cryptographic techniques are used to protect the authorization model in order to avoid the CSP being able to disclose data without data owner consent. Concretely, the solution is based on Proxy Re-Encryption (PRE). A PRE scheme [10] is a cryptographic scheme that enables an entity called proxy to re-encrypt data from one key to another without being able to decrypt it. That is, given a couple of key pairs α and β, the proxy could re-encrypt a ciphertext $c_α$ encrypted under α public key to another ciphertext $c_β$ that can be decrypted using β private key. Using this kind of cryptography, a user $u_α$ can encrypt a piece of data $m$ using his own public key $pub_α$ to obtain a ciphertext $c_α$. A re-encryption key $rk_{α→β}$ can be generated for a proxy to re-encrypt from $α$ to $β$, thus transforming $c_α$ to another ciphertext $c_β$. Then, another user $u_β$ can use his own private key $priv_β$ to decrypt $c_β$ and obtain the plain piece of data $m$.

Several works in this direction have arisen, resulting in diverse Proxy Re-Encryption schemes with different features. The solution proposed in this paper is not tied to a concrete PRE scheme or implementation. However, not all the available PRE schemes are suitable to achieve the goals of this research. In order to characterize and compare different schemes, [10] provided a set of features relevant to proxy re-encryption. Based on this characterization, the following set of features are required by the Proxy Re-Encryption scheme used for the proposal in this paper:

• Unidirectionality. A unidirectional scheme enables the generation of a re-encryption key $rk_{α→β}$ without allowing re-encryption from $β$ to $α$.
• Non-interactivity. A non-interactive scheme enables a user $u_α$ to construct a re-encryption key $rk_{α→β}$ without the participation of $u_β$ or any other entity.
• Multi-use. A multi-use scheme enables the proxy to perform multiple re-encryption operations on a single ciphertext. That is, to re-encrypt from $c_α$ to $c_β$, from $c_β$ to $c_γ$ and so on.

On the other hand, Identity-Based Encryption (IBE) [11] is a type of public key cryptography in which key pairs for a given entity are generated based on the identity of that entity. Using this kind of cryptography, a piece of data $m$ can be encrypted using the identity $id_α$ of a user $u_α$ to obtain a ciphertext $c_α$. Then, user $u_β$ can use his private key $priv_β$ to decrypt $c_α$ and obtain the plain piece of data $m$. Note that no public key $pub_α$ is used for encryption, but the identity of the user $id_α$ is applied instead. IBE schemes usually require the generation of a master key pair that is used to derive user keys based on their identities. The master public key is publicly known and can be directly employed by users to generate the public key of another user based on his identity. In turn, the master private key should be kept private and users can obtain their private keys from a trusted entity that owns the master private key. This entity is called Private Key Generator (PKG).

As will be seen in following sections, an authorization model is composed of several elements (e.g., subjects, roles, grants, etc). The usage of IBE avoids the need to generate and manage a public and private key pair for every authorization element. Instead, the identities of such elements can be directly used within the cryptographic operations.

In [12], an Identity-Based Proxy Re-Encryption (IBPRE) approach is proposed. It combines both IBE and PRE, allowing a proxy to translate a ciphertext encrypted under a user’s identity into another ciphertext under another user’s identity. In this approach, a Master Secret Key (MSK) is used to generate user secret keys from their identities. These secret keys are equivalent to private keys in IBE. No public keys are needed, since identities are directly used in the cryptographic operations. With this approach, a user $u_α$ can encrypt a piece of data $m$ using its identity $id_α$ to obtain a ciphertext $c_{id_α}$ encrypted under $id_α$. A re-encryption key $rk_{α→β}$ can be generated to re-encrypt from $id_α$ to $id_β$. Then, a proxy can use $rk_{α→β}$ to obtain another ciphertext $c_{id_β}$ under the identity of another user $u_β$. This can then use its own secret key $sk_β$ to obtain the plain piece of data $m$. As for IBE approaches, the MSK should be kept private and users can obtain their secret key from the PKG.

This IBPRE scheme is the one selected for the authorization solution proposed in this paper. It has been selected because it combines both PRE and IBE. It fulfills the three aforementioned requirements of proxy re-encryption and supports IBE, what allows to use the identities of the authorization elements for cryptographic operations, avoiding the need to generate and manage a key pair for each element. As mentioned before, the proposed solution is not tied to any PRE scheme or implementation. For the purpose of providing a comprehensive and feasible solution, the rest of this paper is based on the IBPRE.
approach and notation. However, the proposal could be applied to use other Proxy Re-Encryption schemes that fulfill the three aforementioned required features. This includes current or future schemes that could improve performance or security. It could be even a pure PRE scheme without combination with IBE, although that could imply the generation and management of extra key pairs. Moreover, some functionality provided by this solution might be lost, like compatibility with PKI, which is supported by IBPRE and avoids the usage of a PKG.

The following set of functions is provided by IBPRE. It constitutes the cryptographic primitives for the proposal:

setup (p, k) → (p, msk) (1) keygen (p, msk, idα ) → skα (2) encrypt (p, idα , m) → cα (3) rkgen (p, skα , idα , idβ ) → rkα→β (4) reencrypt (p, rkα→β , cα ) → cβ (5) decrypt (p, skα , cα ) → m (6)

Details about the cryptographic operations that are performed by these functions can be found in [12]. A brief description of each function follows. (1) Initializes the cryptographic scheme. It takes as input a security parameter k to initialize the cryptographic scheme (e.g. parameters to generate an elliptic curve) and outputs both the Master Secret Key msk and a set of public parameters p that is used as input for the rest of functions. (2) generates Secret Keys. It takes as input the msk and an identity idα ; and outputs the Secret Key skα corresponding to that identity. (3) encrypts data. It takes as input an identity idα and a plain text m; and outputs the encryption of m under the specified identity cα . (4) generates Re-encryption Keys. It takes as input the source and target identities idα and idβ as well as the Secret Key of the source identity skα ; and outputs the Re-encryption Key rkα→β that enables to re-encrypt from idα to idβ . (5) re-encrypts data. It takes as input a ciphertext cα under identity idα and a Re-encryption Key rkα→β ; and outputs the re-encrypted ciphertext cβ under identity idβ . (6) decrypts data. It takes as input a ciphertext cα and its corresponding Secret Key skα ; and outputs the plain text m resulting of decrypting cα .

IV. CONCLUSION

A data-centric authorization solution has been proposed for the secure protection of data in the Cloud. SeeRBAC allows managing authorization following a rule-based approach and provides enriched role-based expressiveness including role and object hierarchies. Access control computations are delegated to the CSP, being this not only unable to access the data, but also unable to release it to unauthorized parties. Advanced cryptographic techniques have been applied to protect the authorization model. A re-encryption key complement each authorization rule as cryptographic token to protect data against CSP misbehavior. The solution is independent of any PRE scheme or implementation as far as three specific features are supported. A concrete IBPRE scheme has been used in this paper in order to provide a comprehensive and feasible solution. A proposal based on Semantic Web technologies has been exposed for the representation and evaluation of the authorization model. It makes use of the semantic features of ontologies and the computational capabilities of reasoners to specify and evaluate the model. This also enables the application of advanced techniques such as conflict detection and resolution methods. Guidelines for deployment in a Cloud Service Provider have been also given, including an hybrid approach compatible with Public Key Cryptography that enables the usage of standard PKI for key management and distribution. A prototypical implementation of the proposal has been also developed and exposed in this paper, together with some experimental results. Future lines of research include the analysis of novel cryptographic techniques that could enable the secure modification and deletion of data in the Cloud. This would allow to extend the privileges of the authorization model with more actions like modify and delete. Another interesting point is the obfuscation of the authorization model for privacy reasons. Although the usage of pseudonyms is proposed, but more advanced obfuscation techniques can be researched to achieve a higher level of privacy.

REFERENCES


