Privacy-Preserving Enforcement of Spatially Aware RBAC

Michael S. Kirkpatrick, *Member*, *IEEE Computer Society*,
Gabriel Ghinita, *Member*, *IEEE*, and Elisa Bertino, *Fellow*, *IEEE*

Abstract—Several models for incorporating spatial constraints into role-based access control (RBAC) have been proposed, and researchers are now focusing on the challenge of ensuring such policies are enforced correctly. However, existing approaches have a major shortcoming, as they assume the server is trustworthy and require complete disclosure of sensitive location information by the user. In this work, we propose a novel framework and a set of protocols to solve this problem. Specifically, in our scheme, a user provides a service provider with role and location tokens along with a request. The service provider consults with a role authority and a location authority to verify the tokens and evaluate the policy. However, none of the servers learn the requesting user's identity, role, or location. In this paper, we define the protocols and the policy enforcement scheme, and present a formal proof of a number of security properties.

Index Terms—RBAC, privacy, security, access control, applied cryptography.

1 Introduction

Role-based access control (RBAC) has become the industry standard for authorization, and it is widely deployed as it provides organizations with a simplified mechanism for granting privileged access to sensitive resources. Although RBAC systems traditionally consider only identity attributes, such as job title, the emergence of location-based applications has led to the enrichment of the model with spatial features. A number of RBAC extensions incorporate location constraints into access control policies [1], [2], [3], [4], providing organizations with the ability to control resource access based on the physical coordinates of the requesting users.

Geo-spatial RBAC (GEO-RBAC) has a large number of applications, in both military and civilian applications. Consider a military application, where physical presence in a secured room is required for a principal to access a confidential document. Such a protection model can prevent personnel from unintentionally exposing secrets in an environment where principals without the appropriate clearance are present. Alternatively, a policy may state that strictly confidential documents must only be accessed in a room that has been checked thoroughly to ensure there are no unauthorized surveillance devices present. GEO-RBAC addresses these needs and supports permission manipulation at a fine granularity.

- *M.S. Kirkpatrick is with the Department of Computer Science, James Madison University, 701 Carrier Drive, MSC 4103, Harrisonburg, VA 22807. E-mail: kirkpams@jmu.edu.*
- *G. Ghinita is with the Department of Computer Science, University of Massachusetts, 100 Morrissey Blvd., Boston, MA 02125. E-mail: gabriel.ghinita@umb.edu.*
- *E. Bertino is with the Department of Computer Science, Purdue University, 305 N. University Street, West Lafayette, IN 47907-2107. E-mail: bertino@cs.purdue.edu.*

Manuscript received 14 Apr. 2011; revised 26 Aug. 2011; accepted 28 Oct. 2011; published online 7 Dec. 2011.
For information on obtaining reprints of this article, please send e-mail to: tdsc@computer.org, and reference IEEECS Log Number TDSCSI-2011-04-0111.
Digital Object Identifier no. 10.1109/TDSC.2011.62.

Civilian applications also benefit from authorization models and enforcement systems for location-based access control. For instance, medical personnel are often provided with mobile devices that allow them to access medical records of patients. However, these devices may be stolen by a malicious adversary who may be able to break the device password and access the records of high-profile patients (e.g., politicians, celebrities). Furthermore, some employees may not be trustworthy, and may attempt to access the record of such patients in a fraudulent manner. Restricting access to the perimeter of the healthcare provider, where physical surveillance and auditing mechanisms are in place to prevent unauthorized access, can prevent such detrimental leaks of sensitive medical records.

Existing work in GEO-RBAC has primarily focused on the problem of defining an access control model. While the challenge of creating enforcement architectures has received some focus, existing solutions require disclosure of the requesting user's logical location or physical coordinates. Previous work [5] has acknowledged the severe privacy threats that may occur as a result of disclosing fine-grained location information with high frequency. This problem becomes even more serious in a decentralized, loosely coupled environment, such as cloud computing, where services (e.g., data storage) are outsourced to an external provider. Thus, it is no longer the case that all components involved in the authorization process can be fully trusted to protect the privacy of the principals. Even though authorization and auditing mechanisms may prevent unauthorized disclosure of sensitive records and reduce the impact of a leak, similar protection mechanisms for the privacy of the requesting principals' locations do not exist.

Even when a centralized authorization infrastructure exists and all GEO-RBAC components reside within the administrative control of a single organization, it is possible for a rogue administrator to collect and use principals' locations for malicious purposes (e.g., stalking, blackmail). Furthermore, the presence of malware at the service provider (SP) can also be a threat to users. Therefore, we argue that

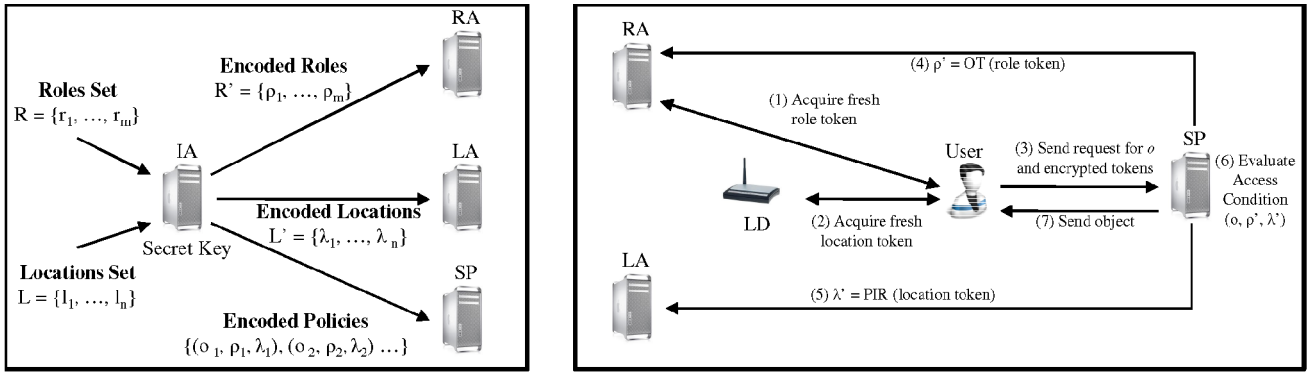

Fig. 1. Overview of privacy-preserving GEO-RBAC.

privacy-preserving access control enforcement is an important desideratum that should be supported by future GEO-RBAC systems.

In this work, we propose a framework for privacy-preserving GEO-RBAC (Priv-GEO-RBAC) that enforces location-based authorization without revealing the identity attributes or physical coordinates of requesting users to the policy enforcement point (PEP). Our approach uses a combination of cryptographic techniques and separation of functionality. The trust assumption in our model is reduced to one component, which generates all cryptographic secrets required for evaluation but *does not participate* in the online operations of the PEP; therefore, this component can be effectively shielded from attacks. The protocols we define ensure that the other components that participate in the online enforcement cannot learn the user's identity, role, or location, even under very powerful adversarial assumptions. Our work reconciles competing goals of security and privacy by supporting fine-grained spatially aware RBAC while simultaneously preserving the privacy of requesting users.

There are clear parallels between our work and access control mechanisms built on attribute-based encryption [6], [7]. That is, location and role are used as attributes for policy evaluation. However, existing attribute-based schemes are built on the premise that attributes (e.g., a driver's license number or date of birth) are fairly persistent for a user, and the user simply needs a credential that certifies that the user possesses the respective attributes. In GEO-RBAC, roles and locations are inherently transient. While one user may activate a role for a short time, he will eventually be replaced by another user; additionally, users are assumed to be mobile. Given the high frequency of attribute changes, temporary credentials that are dynamically and automatically generated are mandatory for usability. To the best of our knowledge, ours is the first work to address the challenge of *attribute-based access control with* ***transient credentials***. Based on the results of this work, we believe that future work in this direction would be valuable.

Fig. 1 gives an overview of the proposed approach: in an offline phase (Fig. 1a), executed only when the policies or the set of roles change, a trusted *Identity Authority (IA)* takes as input the set of all roles and locations (as we discuss later in Section 2, we use a discrete location space model) and encodes them using a secret mapping that never leaves the *IA*. The encoded roles, locations, and policies are then delivered to the *Role Authority (RA)*, *Location Authority (LA)*, and *Service Provider*, respectively. The SP stores the protected objects and ultimately makes the decision of whether to grant access or not. Based on the encoded values, the *RA*, *LA*, and *SP* are not able to link real roles or locations with the access request.

To prevent users from caching tokens for former locations, as well as other replay attacks, we incorporate a time-based code generation approach that causes the credentials to expire. In the online phase (Fig. 1b), the user initiates a role session with the *RA*, acquiring a role token, then presents the role token to a *Location Device (LD)*, acquiring a location token that is bound to the role token (steps 1 and 2). The *LD* could be an RF-based sensor or a card reader present at the entrance to a room. The user then sends the credentials (which do not reveal the role and location) to the *SP* (step 3), who subsequently performs an oblivious transfer (OT) and a private information retrieval (PIR) protocol with the *RA* and *LA*, respectively, thus retrieving data to evaluate the access control policy (steps 4-5). The private retrieval steps are necessary in order to protect the access pattern of the principals: the *RA* and the *LA* do not learn anything about the encrypted credentials that are being used in the current access. In step 6 the *SP* evaluates the access condition on *ciphertext credentials only*, and if the condition is satisfied, it grants access in step 7.

Observe that our protocols are not simply straightforward applications of OT and PIR, as our design goals are complex and seemingly contradictory. First, the information stored with the *RA* and *LA* must be completely independent of the access control policies, as these authorities serve multiple *SPs*. Second, each *SP* must be able to use the information retrieved to evaluate its own policies, *but without learning which policy was satisfied*. Thus, the technical challenge of this work is to combine the individual protocol components to achieve these policy goals, while mitigating information leakage that would facilitate attacks over repeated protocol executions. The specific contributions of this work are:

- We propose an architecture for privacy-preserving GEO-RBAC that allows enforcement of authorization decisions without disclosure of sensitive user

attributes to the service provider. To the best of our knowledge, this is the first such framework.

- We devise a family of cryptographic protocols for authorization enforcement that ensure that service providers cannot link the request to a specific policy and, thus, cannot determine the user's role and/or location. The *RA* and *LA* are also prevented from learning this information. Our approach also preserves client privacy over repeated access requests.
- We present a formal analysis of our scheme using a protocol composition logic (PCL) specification, proving a number of properties under standard cryptographic assumptions.

The rest of the paper is organized as follows: in Section 2 we introduce necessary background materials. We describe the cryptographic protocols for authorization in Section 3, perform a formal security analysis in Section 4, and provide a summary of the key security properties in Section 5. We give an overview of related work in Section 6 and conclude in Section 7.

2 Background Material

2.1 GEO-RBAC

We consider a geospatial role-based access control model (GEO-RBAC) [1], [2], [3], [4] where access control conditions are specified as a combination of predicates on the nonspatial principal credentials (i.e., conventional roles), as well as spatial characteristics, such as the principals' location. In GEO-RBAC, a user initiates a *role session* by activating a traditional role. When making the request, the user's *spatial role* consists of the combination of the role and the current location. Traditional roles are generally defined by a *role hierarchy*. Although our approach for privacy-preserving GEO-RBAC does support role hierarchies, there is a preprocessing step that we require. Specifically, due to the fact that we use encoded values to evaluate access control policy decisions, we can only perform equality tests, and not parent-child relationships in a role hierarchy. Therefore, our approach dictates that when an access control policy for an object is specified, all rules must be expressed directly using leaf nodes of the role hierarchy. For instance, if there exists a role *lecturer* which subsumes two other roles *part-time lecturer* and *full-time lecturer*, every time there is a rule specifying a condition based on role lecturer, the rule is duplicated into two new rules, corresponding to the part-time and full-time counterparts. Note that the preprocessing is trivial and incurs minimal storage overhead for moderately sized hierarchies.

2.2 Spatial Modeling

We assume a bounded and discrete geographical space with respect to which access control policies are defined and enforced. Our space model, which is similar to the work of Jensen et al. [8], [9], divides the reference space into $n-1$ bounded protected areas (pa), whereas the remainder of the space constitutes an additional nth area. Fig. 2 illustrates a typical representation of the space, with six protected areas labeled $pa_1, \ldots, pa_6$. The remainder of the space is marked as T. Observe that protected areas are

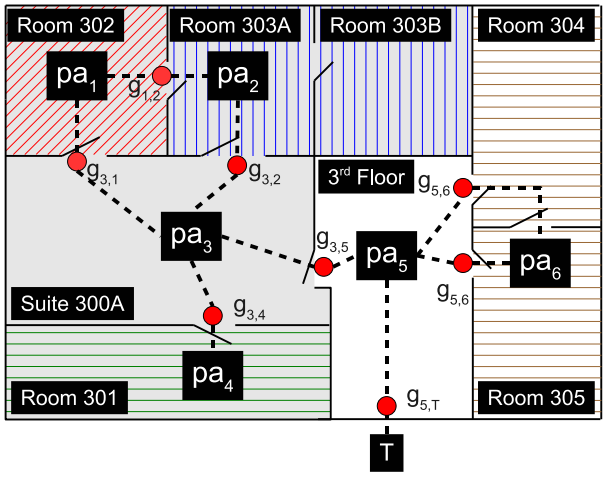

Fig. 2. Spatial model of protected areas in an indoor environment.

logically defined based on topological relations, and do not necessarily indicate geometric properties (e.g., there is no implication that protected areas are equivalent in size or shape). Internally, protected areas are modeled as nodes in a graph, while edges indicate the ability to pass from one such protected area to another (e.g., there is a single edge between the nodes representing pa_5 and pa_6).

In order to ensure that the user is situated in the required location at the access time and that the user remains in the protected area for the duration of the access, it is important to monitor changes in position. The model we adopt accommodates this requirement by enforcing physical proximity controls both when entering and exiting a protected area. Upon exiting a pa, the user must present his or her physical access credentials (e.g., a magnetic card) to a device that guards the pa (indicated in Fig. 2 by $g_{i,j}$). This way, the movement of the user outside the pa can be immediately detected.

To ensure freshness of reported user locations, we employ a time-slice-based expiration mechanism of credentials, that dictates that the user must reacquire a token from the *LD* in the enclosing pa at a regular time interval. In practice, this is implemented by having the *LDs* and the *LA* share pseudorandom sequence of codes, with the help of a common random number generator seed. Combined with the forced reauthentication upon pa exiting, the expiration mechanism ensures that the user's location and path are constantly available for access control enforcement.

2.3 OT and PIR

Both Oblivious Transfer [10] and Private Information Retrieval [11] protocols allow a client to fetch the value of a data item x_i from a remote ordered set $x_1 \ldots x_n$, while preventing the server from learning the value of i. However, the two concepts have different security and performance requirements. In the case of OT, the server requires the additional restriction that the client is able to access only the single item requested. To accomplish this goal, each item is encrypted with a distinct key, and the client retrieves the entire encrypted database. Then, the two parties repeatedly execute a *1-out-of-2* OT protocol to reveal only the desired key. Thus, the client is only able to decrypt

the single item. PIR does not place this additional restriction on the client's access capabilities. Instead, PIR allows the client to read a limited number of records while achieving sublinear (i.e., $O(n)$) communication cost.

In our framework, *RA* stores information about individual users' role sessions. Thus, *SP* uses OT to retrieve this information, which is restricted on a need-to-know basis. On the other hand, *LA* stores public location information; consequently, *SP* uses PIR to retrieve this data more efficiently.

3 Proposed Framework

3.1 Preliminaries

Let $\mathbf{R}$ denote the set of traditional RBAC roles and $\mathbf{L}$ be the set of locations defined by protected areas. In addition, $\mathbf{O}$ denotes the set of objects to be protected, $\mathbf{A}$ is the set of actions that can be performed on objects, and $\mathbf{S}$ denotes the set of subjects that can make requests. Our assumption is that $|\mathbf{A}| \ll |\mathbf{R}| \ll |\mathbf{L}|$. That is, there are very few types of actions (e.g., "read" or "write"), a moderate number of roles (i.e., every person can activate several overlapping organizational roles, such as "accountant," "administrator," or "manager"), and many locations (e.g., rooms, floors, suites, buildings). As $|\mathbf{O}|$ and $|\mathbf{S}|$ do not directly impact our protocol design, we place no assumptions on the size of these sets.

3.2 Principals

Let $\mathbf{P}$ denote the set of all possible principals as follows:

- **Client**—the principal representing the subject making the request. For each such C, there is a unique subject $s \in \mathbf{S}$, but not necessarily vice versa. For instance, the same user may simultaneously initiate sessions with multiple devices (e.g., a smartphone and a laptop); we treat these sessions independently as multiple clients. In our framework, we will refer to the client's public key $pk(C)$, which could be linked to either the user or the device according to the needs of the deployed system.[1]
- **Location device**—a small, proximity-based embedded device that identifies a protected area. In practice, as protected areas may be large, we assume each location $l \in \mathbf{L}$ would have multiple such devices for ease-of-use. However, we treat these devices collectively as a single principal *LD*.
- **Role authority**—a centralized RBAC server that is tasked with authenticating users, as well as creating and maintaining session identifiers.
- **Location authority**—a centralized server that maintains information on the set of *LDs*.
- **Service provider**—a server that controls access to a protected resource. We assume that there are multiple such servers, and the client knows which *SP* to contact when retrieving a desired service.
- **Identity authority**—a trusted third party that establishes and maintains information on the identities of all users and location devices. *IA* is responsible for encoding policies for each *SP*, however, it does not maintain this information after the setup phase. Moreover, *IA* is not involved in any of the runtime protocols, and only exists as a trusted entity for establishing identity credentials.

3.3 Cryptographic Primitives and Notation

Let (Gen, Enc, Dec) denote an encryption scheme that provides indistinguishable encryptions under chosen plaintext attacks (IND-CPA-secure)[2] such that $\mathrm{Gen}(1^n)$ denotes a probabilistic key generation algorithm with security parameter 1^n, $\mathrm{Enc}_k(\cdot)$ denotes encryption using the key k, while $\mathrm{Dec}_k(\cdot)$ denotes the corresponding decryption routine. As key generation and encryption are probabilistic, we adopt the standard convention to denote whether or not assignment is deterministic:

$$k \leftarrow \mathrm{Gen}(1^n)$$
$$c \leftarrow \mathrm{Enc}_k(m)$$
$$m := \mathrm{Dec}_k(c).$$

Our convention is to use the same notation for both public and private key encryption, as the context clearly identifies which is needed. We denote the public and secret keys of principal p as $pk(p)$ and $sk(p)$, respectively, whereas symmetric keys are written as K_p for some identifier p. In the public key case, we require commutative encryption.[3] Specifically, given $c \leftarrow \mathrm{Enc}_{pk(A)}(\mathrm{Enc}_{pk(B)}(m))$,

$$m := \mathrm{Dec}_{sk(A)}(\mathrm{Dec}_{sk(B)}(c)) = \mathrm{Dec}_{sk(B)}(\mathrm{Dec}_{sk(A)}(c)).$$

In addition to encryption, we denote a collision-resistant hash function as $\mathrm{H}(\cdot)$, and $\mathrm{Auth}(\cdot)$ denotes an interactive authentication protocol. The details of $\mathrm{Auth}(\cdot)$ are orthogonal to our scheme and may be selected as desired; the arguments indicate the entity to authenticate. We also deploy two privacy-preserving schemes, as described in Section 2. We denote oblivious transfer as $\mathrm{OT}(i)$, where i indicates the index of the record to be retrieved. Similarly, $\mathrm{PIR}(i)$ denotes private information retrieval for the index i and the items $X(i)$.

Our protocol relies on subtle facets of the RSA assumption that warrant explicit discussion. Recall that the RSA assumption states that, given $N = pq$ (p and q are large primes), e coprime with $\phi(N)$, and some y, it is intractable to find x such that $x^e \equiv y \bmod N$. The critical point is that this implies the difficulty of computing multiplicative inverses modulo $\phi(N)$ when $\phi(N)$ is unknown. Otherwise, the RSA assumption would crumble, as one could efficiently compute d such that $ed \equiv 1 \bmod \phi(N)$ (i.e., $d = e^{-1} \bmod \phi(N)$) and $y^d = (x^e)^d = x^{ed} = x^{1 \bmod \phi(N)} \equiv x \bmod N$. Thus, the RSA assumption implies that *computing multiplicative inverses modulo $\phi(N)$ is intractable when the factorization of N is unknown.* Note that,

1. Observe that using the same key for multiple sessions (e.g., if the user creates sessions on multiple devices with the same public key) could be a privacy concern, as these sessions are linkable. A straightforward solution would be to prohibit users from using the same key for multiple sessions.

2. We stress here the importance that encryption must be IND-CPA-secure. Specifically, IND-CPA provides a probabilistic guarantee that encrypting the same message with the same key multiple times produces *distinct* ciphertexts. Consequently, given $c_1 \leftarrow \mathrm{Enc}_k(m_1)$ and $c_2 \leftarrow \mathrm{Enc}_k(m_2)$, an observer cannot determine $m_1 \stackrel{?}{=} m_2$ without decrypting the messages, even if the encryption key k is known. See [12] for further discussion.

3. Commutativity is often conflated with homomorphic encryption. This is natural, as well-known homomorhpic cryptosystems, such as RSA, are commutative. However, the two properties are distinct, and we only require commutativity, not homomorphic properties.

when we write $\iota^{x^{-1}}$, the implication is that the exponent is the multiplicative inverse of x modulo $\phi(N)$, and the exponentiation is evaluated modulo N. Thus,

$$\iota^{x^{-1} \bmod \phi(N)} = \iota^{x^{-1}} \bmod N.$$

3.4 Identity Establishment

Our role and location authentication scheme relies on cryptographic properties of the group $\mathbf{Z}_N^*$ such that the RSA assumption is satisfied.[4] The value N is made public to all principals. *IA* selects a number of unique role identifiers as follows: First, $\iota \in \mathbf{Z}_N^*$ serves as the basis of our identification scheme. Next, *IA* selects role identifiers $\rho \in \mathbf{Z}_N^*$ such that $\sqrt{N} \leq \rho \leq \phi(N)$. *IA* provides *RA* with ι, ρ, and $\phi(N)$, so that *RA* can compute multiplicative inverses as needed. *IA* also ensures that *RA* knows the mapping from ρ to the corresponding RBAC role. These values must be kept secret to *RA*.

IA then creates location identifiers $\lambda \in \mathbf{Z}_N^*$ subject to the same constraint that $\sqrt{N} \leq \lambda \leq \phi(N)$. This size constraint ensures that $\rho\lambda \geq N$, which strengthens the security guarantees of our scheme. In crafting the λ values, *IA* must ensure there are no colliding pairs of products $\rho\lambda$. That is, for ρ, $\widehat{\rho}$, λ, $\widehat{\lambda}$ with $\rho \neq \widehat{\rho}$ and $\lambda \neq \widehat{\lambda}$, *IA* ensures that $(\rho\lambda) \neq (\widehat{\rho}\widehat{\lambda})$. If such a collision occurs, *IA* discards one of the values and selects a new element from $\mathbf{Z}_N^*$. Once all values are created, *IA* provides *LA* with $\iota^{-1} \bmod N$, the λ values, and the mapping to the corresponding locations.

Finally, *IA* maintains a persistent mapping for actions to values $\alpha \in \mathbf{Z}_N^*$, also subject to the constraint $\sqrt{N} \leq \alpha \leq \phi(N)$. The mapping of α to actions is kept private to *IA*. Observe that the values ρ and λ do not have any particular sensitivity, unless the mappings from the values to the corresponding roles and locations are known. However, if *SP* knows these values, it provides linkability between policies and requests. In order to minimize this threat, one of the goals of our scheme is to reduce the likelihood that *SP* is able to form such correlations. We will revisit this discussion in Section 4.

3.5 Priv-GEO-RBAC Policies

In Priv-GEO-RBAC, for the object $o \in \mathbf{O}$, service providers define a single **object policy** P_o such that $P_o = <p_{o.1}, \ldots, p_{o.m}>$, where

$$p_{o.i} = <r, l, a> \text{ for some } r \in \mathbf{R}, l \in \mathbf{L}, a \in \mathbf{A}.$$

The policy semantics are based on a white-list, meaning the presence of a policy $p_{o.i}$ grants the permission for role r to perform action a on o while in location l. Absence of such a policy indicates denial of the request. For notation, $p_{o.i}[r]$ denotes the role, $p_{o.i}[l]$ denotes the location, and $p_{o.i}[a]$ denotes the action.

When *SP* establishes or changes the policy for an object o, *SP* contacts *IA* to encode the policies. *IA* first creates a new object identifier $\delta \in \mathbf{Z}_N^*$ such that $\sqrt{N} \leq \delta \leq N$. As with $\rho\lambda$, *IA* ensures that, for all actions, there are no colliding products $\alpha\delta = \widehat{\alpha}\widehat{\delta}$ when $\alpha \neq \widehat{\alpha}$ or $\delta \neq \widehat{\delta}$. Now, consider the policy $p_{o.i} = <r, l, a>$. *IA* finds the identifiers ρ, λ, $\alpha \in \mathbf{Z}_N^*$ for the role, location, and action. The policy is then encoded as the tuple

$$\widehat{p_{o.i}} = <(\rho\lambda)^{-1}(\alpha\delta) \bmod \ \phi(N), \quad \iota^{\alpha\delta} \bmod \ N>,$$

where the $(\rho\lambda)^{-1}$ denotes the multiplicative inverse of $(\rho\lambda)$ modulo $\phi(N)$.

Note the following properties of this encoding. First, as there are no colliding products $\alpha\delta$, there will be no object-action pairs that have the same encoded policies, even if the role-location pairs are the same. This prevents *SP* from linking encoded policies. Next, without knowledge of $\phi(N)$, *SP* cannot compute the multiplicative inverse $((\rho\lambda)^{-1}(\alpha\delta))^{-1} = (\rho\lambda)(\alpha\delta)^{-1} \bmod \phi(N)$. If *SP* could compute this value, it would have

$$(\iota^{\alpha\delta})^{(\rho\lambda)(\alpha\delta)^{-1}} \equiv \iota^{\rho\lambda} \bmod \ N,$$

which would clearly allow linkability for any encoded policy with the same role-location pair. However, *this calculation requires knowledge of* ϕ (N), *which* SP *lacks*. Finally, given the assumption that α and δ are large and *SP* has no knowledge of α, δ, or ι, *SP* cannot link policies across different objects or actions.

IA returns the encoded object policy $\widehat{P_o}$ that consists of the array $<\widehat{p_{o.1}}, \ldots, \widehat{p_{o.m}}>$ in a random order. This shuffling prevents *SP* from determining $\widehat{p_{o.i}} \stackrel{?}{\equiv} p_{o.i}$. Observe, though, that all encoded policies relating to the same action on the same object share the value $\iota^{\alpha\delta}$. Consequently, *SP* is able to group the encoded policies according to the action. Based on this grouping, we can refer to the policy group $\widehat{P_{o.a}} = <\widehat{p_{o.a.1}}, \ldots, \widehat{p_{o.a.q}}>$ for the object o and action a. Note that, as *SP* has no knowledge of ρ, λ, ι, or δ, it can only see the policies as distinct pairs of large integers.

3.6 Protocols

In this section, we define the protocols of our framework. In these definitions, we only explicitly identify encryptions that are required to prevent one of the legitimate principals from learning a protected value. For instance, Protocol 1 sends the symmetric key K_c in the clear, as both *RA* and *C* are authorized to know this key; on the other hand, Protocol $\mathcal{Q}$ sends multiple pieces of data encrypted with a public key to prevent one of the principals from learning the encrypted value. Consequently, system deployments should consider the security threats of the underlying communications channel and add cryptographic protections as needed.

Protocol for RBAC session creation. Fig. 3 shows Protocol 1, which used to create a new RBAC session. After authenticating the user (and the user's authority to activate role $r \in \mathbf{R}$), *RA* generates a number of session parameters as follows: The value $x \in \mathbf{Z}_N^*$ is a random positive integer, b denotes a nonce used in the $\mathsf{Auth}(\cdot)$ scheme (known to both *RA* and *C*), and pwd_c is the user's password.[5] *RA* creates a new *record* (identified by index i_r) in its session database with a commitment token σ_r, and marks the record with a boolean value *valid*. Unless the system allows simultaneous activation of multiple roles, *RA* invalidates all other records

4. Recall that $\mathbf{Z}_N^* = \{a \in \{1, \ldots, N-1\} \mid \gcd(a, N) = 1\}$. This ensures that all such $z \in \mathbf{Z}_N^*$ have unique multiplicative inverses.

5. If the particular instantiation of $\mathsf{Auth}(\cdot)$ does not require a nonce b, *RA* can generate one for *C*.

Protocol 1 – creating a new role session

1) $[C \leftrightarrow RA]$ $\mathsf{Auth}(C, r, b)$
   $[RA]$ $\sigma_r := < \iota^{\gamma}(\iota^{\rho})^x = \iota^{\gamma}\iota^{\rho x}, x^{-1} \mod \phi(N) >$
   $[RA]$ $\mathsf{K}_r \leftarrow \mathsf{Gen}(1^n)$
   $[RA]$ $\mathsf{K}_c \leftarrow \mathsf{Gen}(1^n)$
   $[RA]$ $record := \mathsf{Enc}_{\mathsf{K}_r}(\sigma_r \parallel \mathsf{K}_c \parallel valid)$
   $[RA]$ $\tau_r \leftarrow \mathsf{Enc}_{pk(RA)}(i_r \parallel \mathsf{K}_r \parallel \mathsf{H}(b \parallel pwd_c))$
2) $[RA \rightarrow C]$ τ_r, K_c

Protocol 2 – retrieving a proof-of-location

1) $[C \rightarrow LD]$ τ_r
   $[LD]$ $sig := \mathsf{H}(d \parallel code_l[T] \parallel \tau_r)$
   $[LD]$ $\tau_l \leftarrow \mathsf{Enc}_{pk(LA)}(i_l \parallel d \parallel exp \parallel T \parallel sig)$
2) $[LD \rightarrow C]$ τ_l

Fig. 3. Protocols for retrieving tokens τ_r and τ_l.

for this user by flipping the corresponding boolean value in those records. We refer to τ_r as the **role token**. The rationale for encrypting the record with the key K_r is to ensure that the information can only be accessed by a *SP* with the corresponding role token. Alternative approaches [13] have integrated access control with OT, but encrypting the record is sufficient for our framework. The other key K_c is generated to ensure the object can be accessed only by the legitimate C.

Astute readers will note the similarity in structure between $\iota^{\gamma}\iota^{\rho x}$ and the Pedersen commitment scheme [14]. In the Pedersen commitment, for a cyclic group generated by g, the prover aims to commit to $h = g^x$, where x is unknown. To do so, the prover first generates $g^s h^t$ for random values s and t. Later, the prover reveals s and t. This scheme is proven to be *perfectly hiding* of the exponent x. While our goals are different (e.g., we do not require information theoretic security), this structure ensures that, for any two role sessions with the same ρ, it holds that $\iota^{\gamma}\iota^{\rho x} \neq \iota^{\gamma}\iota^{\widehat{\rho x}}$ when $x \neq \widehat{x}$. Furthermore, the obfuscating factor ι^{γ} ensures that knowledge of x^{-1} and $\widehat{x^{-1}}$ are insufficient to determine if the ρ and $\widehat{\rho}$ values are equivalent in two distinct sessions.

Protocol for retrieving a proof-of-location. When the user wishes to make a request, he must retrieve a proof-of-location from a *LD* for the corresponding protected area. This proof must bind the user session to the location at a particular time. To do this, *LD* generates the data shown in Fig. 3, where i_l is the index for *LD* in the location database, d is a nonce, exp denotes an expiration time, $code_l[T]$ is a rolling passcode (i.e., it repeatedly changes after a set time interval) at timestamp T, and the **location token** is τ_l. Additionally, note that the location token τ_l is dependent on the role token τ_r, thus binding the role session to the location.

Protocol for policy enforcement. Fig. 4 shows Protocol $\mathcal{Q}$. Once C has the role and location tokens, C initiates Protocol $\mathcal{Q}$ with the relevant *SP* to request access to perform action $a \in \mathbf{A}$ on object $o \in \mathbf{O}$. For simplicity, we will assume the action is $read$.[6] C sends the tokens to *SP*. *SP* gets *RA* and

6. It is straightforward to add support for $write$ actions by appending MACs as necessary. Note, though, that using a persistent secret key to generate the MAC will allow *SP* to link writes. Instead, τ_r should be augmented with a session public key, and Protocol 1 should send the corresponding session private key to C.

Protocol $\mathcal{Q}$ – requesting access to protected resource

1) $[C \rightarrow SP]$ τ_r, τ_l, o, a
2) $[SP \rightarrow RA]$ $e_{sr} \leftarrow \mathsf{Enc}_{pk(SP)}(\tau_r)$
3) $[RA \rightarrow SP]$ $e_s := \mathsf{Dec}_{sk(RA)}(e_{sr})$
   $[SP]$ $(i_r \parallel \mathsf{K}_r \parallel \mathsf{H}(b \parallel pwd_c)) := \mathsf{Dec}_{sk(SP)}(e_s)$
4) $[SP \rightarrow C]$ $z \in \{0,1\}^*$
5) $[C \rightarrow SP]$ $h := \mathsf{H}(z \parallel \mathsf{H}(b \parallel pwd_c))$
6) $[SP \leftrightarrow RA]$ $e_{OT} := \mathsf{OT}(i_r)$
   $[SP]$ $(\sigma_r \parallel \mathsf{K}_c \parallel valid) := \mathsf{Dec}_{\mathsf{K}_r}(e_{OT})$
7) $[SP \rightarrow LA]$ $e_{sl} \leftarrow \mathsf{Enc}_{pk(SP)}(\tau_l)$
8) $[LA \rightarrow SP]$ $e'_s := \mathsf{Dec}_{sk(LA)}(e_{sl})$
   $[SP]$ $(i_l \parallel d \parallel exp \parallel T \parallel sig) := \mathsf{Dec}_{sk(SP)}(e'_s)$
9) $[SP \leftrightarrow LA]$ $(\lambda \parallel code_l[T]) := \mathsf{PIR}(i_l)$
10) $[SP \rightarrow RA]$ $\iota^{-\lambda m}$
11) $[RA \rightarrow SP]$ $(\iota^{-\lambda m})^{\gamma} = \iota^{-\lambda m\gamma}$
    $[SP]$ $v := \mathsf{Eval}(\sigma_r, \lambda, \iota^{-\lambda m\gamma}, exp, sig)$
12) $[SP \rightarrow C]$ $e_o \leftarrow \mathsf{Enc}_{\mathsf{K}_c}([o])$

Fig. 4. Access control enforcement protocol.

LA to perform a blind decryption (i.e., *RA* and *LA* do not learn the decrypted messages). Using this information, *SP* authenticates C with a traditional password, retrieves role session information from *RA* using OT, and retrieves location information from *LA* using PIR. Finally, *SP* evaluates the encoded policies based on the data retrieved from *RA* and *LA*. The evaluation procedure ensures that *SP* only learns *if* a policy is satisfied; for a single request, *SP* cannot determine the user's identity, location, or role. Section 4 examines these properties in detail. In the following definition of $\mathcal{Q}$, z denotes a nonce, while $m \in \mathbf{Z}_N^*$ is selected at random from $\sqrt{N} \leq m \leq N-1$.

The first important point to emphasize in this protocol is the blinded decryption in step 3. That is, $\tau_r \leftarrow \mathrm{Enc}_{pk(RA)}(\cdot)$, so $e_{sr} \leftarrow \mathrm{Enc}_{pk(SP)}(\mathrm{Enc}_{pk(RA)}(\cdot))$. By decrypting this message blindly, *RA* is sending the following to *SP*:

$$\begin{aligned} e_s &:= \mathrm{Dec}_{sk(RA)}(e_{sr}) \\ &= \mathrm{Dec}_{sk(RA)}(\mathrm{Enc}_{pk(SP)}(\mathrm{Enc}_{pk(RA)}(\cdot))) \\ &= \mathrm{Dec}_{sk(RA)}(\mathrm{Enc}_{pk(RA)}(\mathrm{Enc}_{pk(SP)}(\cdot))) \\ &= \mathrm{Enc}_{pk(SP)}(\cdot), \end{aligned}$$

which can then be decrypted by *SP*. However, the IND-CPA encryption by *SP* ensures that *RA* cannot determine the original encrypted message, so *RA* fails to learn which role session is being used. Step 8 uses the same technique for locations.

The need for two separate privacy-preserving schemes, OT and PIR, may not be intuitive. This choice was deliberate, as the needs of the respective portions of the protocol are different. First, the role session information is more sensitive, as it includes the key K_c used to communicate with the user. As such, it is important that the *SP* only retrieves the records for that session. PIR cannot provide this guarantee. Second, the location database contains more information than is required by the protocol. Specifically, each record contains additional information about the protected area. PIR allows *SP* to retrieve only the relevant data. Also, the passcode $code_l[T]$ is only good for a short period of time before it expires. As such, *SP* would

have a very short window of opportunity to exploit knowledge of the passcode for other *LDs*. Furthermore, this leak is *not a security threat*, as we will show in our security analysis. Thus, PIR is the correct choice for retrieving this data from *LA*.

The evaluation procedure consists of checking that the location token has not expired (i.e., current time is prior to exp), validating the location signature sig, and evaluating the policy set $P_{o.a}$. To validate sig, *SP* does a straightforward comparison

$$sig \stackrel{?}{=} \mathrm{H}(d \parallel code_l[T] \parallel \tau_r).$$

To prevent network delay from causing the $code$ value to become outdated, a straightforward adaptation would be for $code_l[T]$ to denote multiple, consecutive codes. Evaluating the policy set requires performing a number of calculations. Given the importance of this evaluation to our protocol, we will describe and evaluate this procedure in detail in the following section.

3.7 Functional Correctness

In this section, we focus on the functional correctness of the policy evaluation. That is, we show that *SP* can evaluate the encoded policies correctly, given the role and location information encoded by σ_r and λ. Consider a request for object $o \in \mathbf{O}$, identified by $\delta \in \mathbf{Z}_N^*$, where the action $a \in \mathbf{A}$ is identified by α. Recall that this uniquely identifies the set $\widehat{P_{o.a}} = <\widehat{p_{o.a.1}}, \ldots, \widehat{p_{o.a.q}}>$. In addition, *SP* retrieved $\sigma_r := <\iota^{\gamma}\iota^{\rho x}, x^{-1} \bmod \ \phi(N)>$ from *RA* during the OT step of the protocol, and *SP* retrieved λ from *LA* during the PIR. Next, *SP* selected a nonce $m \in \mathbf{Z}_N^*$, and used λ and $\iota^{-1} \bmod \ N$ to calculate $\iota^{-\lambda m} \bmod \ N$. *SP* sent this value to *RA*, who responded with $\iota^{-\lambda m\gamma}$, where γ is the persistent obfuscator used by *RA*. As γ is large, it is intractable for *SP* to compute the discrete logarithm and learn γ. Similarly, *RA* cannot determine λ or m. Now, consider an encoded policy $\widehat{p_{o.a.i}} \in \widehat{P_{o.a}}$, which we denote as $<s, \iota^{\alpha\delta}>$. In order to evalute $\widehat{p_{o.a.i}}$, *SP* performs the following calculations:

1. $\left(\iota^{\gamma}\iota^{\rho x}\right)^{\lambda m} = \iota^{\gamma\lambda m}\iota^{\rho x\lambda m}$,
2. $\iota^{-\lambda m\gamma} \cdot \iota^{\gamma\lambda m}\iota^{\rho x\lambda m} \equiv \iota^{0+\rho x\lambda m \bmod \ \phi(N)} = \iota^{\rho x\lambda m}$,
3. $\left(\left(\iota^{\rho x\lambda m}\right)^{s}\right)^{x^{-1}} = \iota^{\rho x\lambda m s x^{-1}} \equiv \iota^{\rho\lambda m s \bmod \ \phi(N)}$, and
4. $\iota^{\rho\lambda m s} \stackrel{?}{=} \left(\iota^{\alpha\delta}\right)^{m}$.

The policy is satisfied if and only if the equality holds. For clarity in the following proofs, we refer to this test as the **policy equation**.

Theorem 1. *Under the assumption that all parties behave honestly, the policy equation is satisfied if and only if the access control policy $p_{o.i}$ is satisfied.*

Proof. There are two cases to consider. First, assume $p_{o.i}$ is satisfied, given the credentials used in the protocol. In that case, $s = (\rho\lambda)^{-1}(\alpha\delta)$ (i.e., the role identified by ρ and the location identified by λ match those in the request). Observe that

$$\begin{aligned}\left(\iota^{\rho\lambda m s}\right)^{(\rho\lambda)^{-1}(\alpha\delta)} &= \iota^{\rho\lambda m(\rho\lambda)^{-1}(\alpha\delta)} \\ &= \left(\iota^{(\rho\lambda)(\rho\lambda)^{-1}}\right)^{m\alpha\delta} \\ &\equiv \left(\iota^{1 \bmod \ \phi(N)}\right)^{m\alpha\delta} \\ &= \iota^{m\alpha\delta}.\end{aligned}$$

The equivalence follows from Euler's theorem, which states that $\iota^{\phi(N)} \equiv 1 \bmod \ N$ if ι and N are coprime. Consequently, the policy equation holds

$$\iota^{\rho\lambda m s} \equiv \iota^{m\alpha\delta} \bmod \ N = \left(\iota^{\alpha\delta}\right)^{m}.$$

Therefore, if the credentials satisfy the original policy, then the policy equation holds. Now, consider the other implication. We must show that, if the policy equation holds, then the original policy is satisfied. We will proceed with a proof by contradiction. Assume the policy equation holds, but the credentials presented do *not* satisfy the policy. Since the policy is not satisfied, $s = (\widehat{\rho}\widehat{\lambda})^{-1}(\alpha\delta)$, where either $\widehat{\rho} \neq \rho$, $\widehat{\lambda} \neq \lambda$, or both. Assume $\widehat{\rho} \neq \rho$ but $\widehat{\lambda} = \lambda$. Then

$$(\rho\lambda)(\widehat{\rho}\lambda)^{-1} = (\rho\widehat{\rho^{-1}})(\lambda\lambda^{-1}) \equiv (\rho\widehat{\rho^{-1}}) \bmod \ \phi(N).$$

Observe that the policy equation will only hold if $\iota^{\rho\widehat{\rho^{-1}}} = \iota$, which can only occur if $\rho\widehat{\rho}^{-1} \equiv 1 \bmod \ \phi(N)$. However, this requires that $\widehat{\rho}^{-1}$ is the inverse of ρ, meaning $\widehat{\rho} = \rho$, which contradicts our assumption. Thus, if $\widehat{\lambda} = \lambda$ and the policy equation holds, then $\widehat{\rho} = \rho$. By the same rationale, if $\widehat{\rho} = \rho$ and the policy equation holds, then $\widehat{\lambda} = \lambda$. Thus, if the policy equation holds and one of the credentials is correct, then the other must be correct, contradicting the assumption that the policy is not satisfied. Now, assume $\widehat{\rho} \neq \rho$ and $\widehat{\lambda} \neq \lambda$. If the policy equation holds, then

$$(\rho\lambda)(\widehat{\rho}\widehat{\lambda})^{-1} \equiv 1 \bmod \ \phi(N).$$

This implies $(\widehat{\rho}\widehat{\lambda})^{-1}$ is the inverse of $(\rho\lambda)$, implying $(\widehat{\rho}\widehat{\lambda}) = (\rho\lambda)$. However, we explicitly prohibited such a collision of products. That is, *IA* ensures that no such pair is created. Thus, if the policy equation holds, then the credentials provided must be correct. □

4 Security Analysis

In this section, we present an analysis of our protocol in two ways. First, we consider the security of our framework under the Dolev-Yao adversarial model [15]. Under this model, an adversary $\mathcal{A}$ is capable of sending and receiving messages, decrypting messages with known keys, storing data, and generating new data. The goals of such an adversary include impersonating a legitimate participant and learning information for a future attack. Second, we evaluate the privacy guarantees of our framework against rational participants. Adversaries in this model participate honestly unless they are able to gain by deviating. Specifically, the participant gains something if he learns information of value about another participant. We start with some preliminary definitions and notation.

4.1 Definitions and Notation

A function f is **negligible** if, for any constant $c > 1$, there exists a constant N such that $\forall n > N$, $f(n) < n^{-c}$. We write negl to indicate a negligible function.[7] Given two values x

7. The N here is not the same as the RSA modulus used throughout Section 3.

$\mathbf{Serv_{Acc}} \equiv (RA, LA)[$ receive $\widehat{C}, \widehat{SP}, (\tau_r, \tau_l, o, read)$; $e_{sr} \leftarrow \mathsf{enc}\ \tau_r, pk(SP)$; send $\widehat{SP}, \widehat{RA}, e_{sr}$; receive $\widehat{RA}, \widehat{SP}, e_s$; $(i_r \parallel \mathsf{K}_r \parallel h') := \mathsf{dec}\ e_s, sk(SP)$; new z; send $\widehat{SP}, \widehat{C}, z$; receive $\widehat{C}, \widehat{SP}, h$; receive $\widehat{RA}, \widehat{SP}, e_{OT} := \mathsf{OT}\ (i_r)$; $(\sigma_r \parallel \mathsf{K}_c \parallel valid) := \mathsf{dec}\ e_{OT}, \mathsf{K}_r$; $e_{sl} \leftarrow \mathsf{enc}\ \tau_l, pk(SP)$; send $\widehat{SP}, \widehat{LA}, e_{sl}$; receive $\widehat{LA}, \widehat{SP}, e'_s$; $(i_l \parallel d \parallel exp \parallel T \parallel sig) := \mathsf{dec}\ e'_s, sk(SP)$; receive $\widehat{LA}, \widehat{SP}, (\lambda \parallel code_l[T]) := \mathsf{PIR}(i_l)$; new m; $\iota_{lm} := \iota^{-\lambda m}$; send $\widehat{SP}, \widehat{RA}, \iota_{lm}$; receive $\widehat{RA}, \widehat{SP}, \iota_{lmg}$; $v := \mathsf{eval}\ \sigma_r, \lambda, m, \iota_{lmg}, \widehat{p_{o.a.i}}, exp, sig$ $e_o \leftarrow \mathsf{enc}\ [o], \mathsf{K}_c$; send $\widehat{SP}, \widehat{C}, e_o$; $]SP()$	$\mathbf{Init_{Acc}} \equiv (SP, \tau_r, \tau_l, o, read, b, pwd_c)[$ send $\widehat{C}, \widehat{SP}, (\tau_r, \tau_l, o, read)$; receive $\widehat{SP}, \widehat{C}, z$; $y := \mathsf{hash}(b \parallel pwd_c)$; $h := \mathsf{hash}(z \parallel y)$; send $\widehat{C}, \widehat{SP}, h$; receive $\widehat{SP}, \widehat{C}, e_o$; $[o] := \mathsf{dec}\ e_o, \mathsf{K}_c$ $]C([o])$ $\mathbf{Auth_{Role}} \equiv ()[$ receive $\widehat{SP}, \widehat{RA}, e_{sr}$; $e_s := \mathsf{dec}\ e_{sr}, sk(RA)$; send $\widehat{RA}, \widehat{SP}, e_s$; send $\widehat{RA}, \widehat{SP}, \mathsf{OT}\ (\cdot)$; receive $\widehat{SP}, \widehat{RA}, \iota_{lm}$; $\iota_{lmg} := (\iota_{lm})^{\gamma}$; send $\widehat{RA}, \widehat{SP}, \iota_{lmg}$; $]RA()$ $\mathbf{Auth_{Loc}} \equiv ()[$ receive $\widehat{SP}, \widehat{LA}, e_{sl}$; $e'_s := \mathsf{dec}\ e_{sl}, sk(LA)$; send $\widehat{LA}, \widehat{SP}, e'_s$; send $\widehat{LA}, \widehat{SP}, \mathsf{PIR}\ (\cdot)$; $]LA()$
$\theta_C \stackrel{def}{=} \{\tau_r, \tau_l, \mathsf{K}_c, pwd_c, b, o, a\}$	$\phi_{C,R} \stackrel{def}{=} \{z, y, h, e_o, [o]\}$
$\theta_{SP} \stackrel{def}{=} \{\widehat{P_{o.a}}, \iota^{-1} \mod \phi(N)\}$	$\phi_{SP,R} \stackrel{def}{=} \{\tau_r, \tau_l, o, a, i_r, \mathsf{K}_r, h', z, h, \sigma_r, \mathsf{K}_c, valid, i_l, d, exp, T, \lambda$ $sig, m, \iota^{-\lambda m}, \iota^{-\lambda m\gamma}, \iota^{\rho x\lambda m}, \iota^{\rho\lambda m}, s, \iota^{\alpha\delta}, \iota^{\alpha\delta m} code_l[T], v, e_o\}$
$\theta_{RA} \stackrel{def}{=} \{\iota, \rho, \gamma, \phi(N), x, \sigma_r, \mathsf{K}_r, \mathsf{K}_c, valid, \tau_r, \}$	$\phi_{RA,R} \stackrel{def}{=} \{\iota^{-\lambda m}, \iota^{-\lambda m\gamma}\}$
$\theta_{LA} \stackrel{def}{=} \{\lambda, i_l, d, exp, sig, code_l[T], \tau_l\}$	$\phi_{LA,R} \stackrel{def}{=} \emptyset$

Fig. 5. PCL specification of roles for $\mathcal{Q}$, including the sets indicating prior knowledge (ϕ) and plaintext knowledge gained (θ) during run R of $\mathcal{Q}$.

and y, $x \stackrel{?}{=} y$ is conventional notation to refer to the question of the equivalence of x and y. We also write $\{x \stackrel{?}{=} y\}_P$ to indicate that a principal P is able to correctly answer the question. In our formalization and our proofs, we frequently refer to the **knowledge** that a principal has. For brevity, we omit IND-CPA-secure messages from this knowledge, as

$$\Pr[\{m \stackrel{?}{=} \widehat{m}\}_P \mid \mathrm{Has}(P, \{\mathrm{Enc}_K(m), \mathrm{Enc}_K(\widehat{m}), K, m\})] - \Pr[\{m \stackrel{?}{=} \widehat{m}\}_P \mid \mathrm{Has}(P, \emptyset)] \leq \mathrm{negl}(n).$$

That is, the ciphertexts reveal no useful information to P, even if P knows the key used for both and one of the plaintexts. Similarly, we omit the details of messages exchanged in PIR and OT, as the privacy of those schemes is demonstrated in existing work.

Fig. 5 shows the translation of Protocol $\mathcal{Q}$ into protocol composition logic behaviors[8] that indicate the actions taken by an honest participant. Due to space constraints, we will not provide an overview of PCL, and refer the reader to the work of Datta et al. [16]. Using this specification, we can model the knowledge gained by each participant in a formal manner. That is, we write $\mathrm{Has}(P, \theta_P)$ to indicate that P knows the value of all variables in the set θ_P. To describe the knowledge gained by P during run R of $\mathcal{Q}$, we use the proposition:

$$\mathrm{Has}(P, \theta_P)[R]_P\ \mathrm{Has}(P, \phi_{P,R}).$$

Our convention is to use θ for prior knowledge, while ϕ includes information gained during the execution.[9] For simplicity, we assume $\theta_P \subseteq \phi_{P,R}$ (P does not forget the data in θ_P) and generally omit this notation for brevity. Fig. 5 formalizes these sets for honest parties in $\mathcal{Q}$. Note that, also for the sake of brevity, we omit public data and private keys from these sets. That is, public keys and the group $\mathbf{Z}_N^*$ are known to all parties at all times, while private keys are known only to the corresponding principal.

The first concern of our protocol is whether or not an eavesdropping adversary $\mathcal{A}$ can impersonate one of the principals by observing the execution by honest participants. To model intrusion, we start by defining the following impersonation experiment:

The impersonation experiment $\mathrm{Imp}^P_{\mathcal{A},\mathcal{Q}}$:

1. All parties execute run R of $\mathcal{Q}$, with $\mathcal{A}$ taking on the role of principal P.
2. Honest participants abort R if nonces are replayed, malformed messages are received, or intermediate value comparisons fail.
3. All participants $\widehat{P}$ honestly reveal $\phi_{\widehat{P},R}$.

8. The proper PCL terminology for what we call "behaviors" is "roles," which is problematic when discussing a protocol used for RBAC enforcement.

9. Note that ϕ is used to denote both the information gain and Euler's totient function.

4. Upon request, each participant $\widehat{P}$ will decrypt any message e_K if $K \in \theta_{\widehat{P}}$.
5. The experiment outputs 1 if and only if no honest participant can demonstrate $\mathcal{A}$ has attempted to impersonate P. Otherwise, the output is 0.

We say **impersonation of *P* by $\mathcal{A}$ fails** if, for a sufficiently large positive integer n:

$$\Pr\left[\mathrm{Imp}^{P}_{\mathcal{A},\mathcal{Q}} = 1\right] \leq \frac{1}{n} + \mathrm{negl}(n).$$

That is, $\mathrm{Imp}^{P}_{\mathcal{A},\mathcal{Q}} = 1$ iff $\mathcal{A}$ can execute the behavior of P without detection by any other principals.[10] The parameter $1/n$ is used to account for $\mathcal{A}$ performing an extraordinary feat (e.g., forging a cryptographic hash or encrypted message by blindly guessing).

Our other concern in this paper is what information the protocol reveals to honest participants about C. To evaluate this goal, we define an additional experiment as follows:

The privacy-preservation experiment $\mathrm{Priv}^{C}_{\mathcal{A},\mathcal{Q}}$:

1. All parties execute $\mathcal{Q}$, yielding the knowledge sets in Fig. 5. Let u denote the user's identity, r denote the role used, and l denote the user's location.
2. $\mathcal{A}$ guesses user $\widehat{u}$, role $\widehat{r}$, or location $\widehat{l}$.
3. The experiment outputs 1 if and only if $u = \widehat{u}$, $r = \widehat{r}$, or $l = \widehat{l}$. Otherwise, the output is 0.

We say a protocol **preserves the privacy of the client *C* against the adversary** $\mathcal{A}$ if,

$$\Pr\left[\mathrm{Priv}^{C}_{\mathcal{A},\mathcal{Q}} = 1 \mid \mathrm{Has}(\mathcal{A}, \phi_{\mathcal{A},R})\right] - \Pr\left[\mathrm{Priv}^{C}_{\mathcal{A},\mathcal{Q}} = 1 \mid \mathrm{Has}(\mathcal{A}, \emptyset)\right] \leq \mathrm{negl}(n).$$

That is, the adversary's chance of guessing the role, location, or identity of the user is negligibly different than blindly guessing.

4.2 Security against Intruders

In this section, we consider security under the Dolev-Yao adversarial model. That is, $\mathcal{A}$ observes all data transmitted, can send and receives messages, store and retrieve data, and decrypt messages with known keys. We assume $\mathcal{A}$ begins with only public knowledge. As such, $\theta_{\mathcal{A}} = \emptyset$. Recall that our protocol is built on the assumption that underlying channels are encrypted, and we omitted this fact from the protocol definition for brevity and clarity. As such, $\mathcal{A}$ can only see the encrypted versions of data transmitted.

As a preliminary, consider an eavesdropping adversary $\mathcal{A}$ observing run R of $\mathcal{Q}$. Given $\theta_{\mathcal{A}} = \emptyset$, the information gained by $\mathcal{A}$ (excluding OT and PIR) consists of the following pieces of data:

$$\phi_{\mathcal{A},R} = \left\{(\tau_r, \tau_l, o, a), e_{sr}, e_s, z, h, e_{sl}, e'_s, \iota^{-\lambda m}, \iota^{-\lambda m\gamma}, e_o\right\}.$$

Lemma 1. *Consider adversary $\mathcal{A}$ attempting to execute run R' of $\mathcal{Q}$. Impersonation of* RA *by an eavesdropping adversary $\mathcal{A}$ fails.*

Proof. By eavesdropping on R, the following proposition holds:

$$\mathrm{Has}(\mathcal{A}, \emptyset)\ [R]_{\mathcal{A}}\ \mathrm{Has}(\mathcal{A}, \phi_{\mathcal{A},R}).$$

In run R', $\mathcal{A}$ would receive $\widehat{e_{sr}}$. As $\mathcal{A}$ cannot determine if $\widehat{e_{sr}} \stackrel{?}{=} e_{sr}$, if $\mathbf{C}$ denotes the set of ciphertexts,

$$\Pr[\{\tau_r \stackrel{?}{=} \widehat{\tau}_r\}_{\mathcal{A}} \mid \mathrm{Has}(\mathcal{A}, \phi_{\mathcal{A},R} \cup \{\widehat{e_{sr}}\}) \leq \frac{1}{\mid \mathbf{C} \mid} + \mathrm{negl}(n).$$

That is, $\mathcal{A}$ cannot determine if the role tokens are the same. Consequently, $\mathcal{A}$ has two options. If $\mathcal{A}$ believes $\tau_r = \widehat{\tau}_r$, it can replay e_s; otherwise, it can generate a random guess $\widehat{e_s}$. Consider the former option. If $\tau_r \neq \widehat{\tau}_r$, when C generates $\mathrm{H}(\widehat{b} \parallel pwd_c)$, it will only match the h' retrieved by *SP's* decryption of e_s if $\mathrm{H}(\widehat{b} \parallel pwd_c) = \mathrm{H}(b \parallel pwd_c)$, and, if $\mathbf{H}$ denotes the set of possible hash outputs,

$$\Pr[\neg\{\mathcal{A} \stackrel{?}{=} RA\}_C] \leq \frac{1}{\mid \mathbf{H} \mid} + \mathrm{negl}(n),$$

indicating that C will be able to identify the impersonation with all but trivial probability. Alternatively, if $\tau_r = \widehat{\tau}_r$, at the end of the protocol, $\mathcal{A}$ must compute $(\widehat{\iota^{-\lambda m}})^{\gamma}$ without knowledge of γ. Consequently, $\mathcal{A}$ must guess and

$$\Pr[\neg\{\mathcal{A} \stackrel{?}{=} RA\}_{SP}] = \Pr[\mathrm{Has}(\mathcal{A}, \{\widehat{\iota^{-\lambda m\gamma}}\}) \mid \mathrm{Has}(\mathcal{A}, \{\widehat{\iota^{-\lambda m}}\}) \wedge \neg\mathrm{Has}(\mathcal{A}, \{\gamma\})] \leq \frac{1}{\mid \mathbf{Z}^{*}_{N} \mid} + \mathrm{negl}(n).$$

Thus, if $\mathcal{A}$ replays e_s, impersonation fails. Alternatively, if $\mathcal{A}$ believes $\tau_r \neq \widehat{\tau}_r$, it would generate a random $\widehat{e_s}$ and a random $\widehat{\iota^{-\lambda m\gamma}}$, which is unlikely as shown above. Thus, for a large n,

$$\Pr\left[\mathrm{Imp}^{RA}_{\mathcal{A},\mathcal{Q}} = 1\right] = \Pr[\neg\{\mathcal{A} \stackrel{?}{=} RA\}_C \wedge \neg\{\mathcal{A} \stackrel{?}{=} RA\}_{SP}] \leq \frac{1}{n} + \mathrm{negl}(n).$$

□

Lemma 2. *Consider adversary $\mathcal{A}$ attempting to execute run R' of $\mathcal{Q}$. Impersonation of* LA *by an eavesdropping adversary $\mathcal{A}$ fails.*

Proof. As in the proof of Lemma 1, $\mathcal{A}$ has the choice of replaying e'_s or generating a random $\widehat{e'_s}$. We have already seen that the latter course fails with near certainty. In the former option, similar to before,

$$\Pr[\{\tau_l \stackrel{?}{=} \widehat{\tau}_l\}_{\mathcal{A}} \mid \mathrm{Has}(\mathcal{A}, \phi_{\mathcal{A},R} \cup \{\widehat{e_{sl}}\}) \leq \frac{1}{\mid \mathbf{C} \mid} + \mathrm{negl}(n).$$

If $\tau_l \neq \widehat{\tau}_l$, then the signature required for verification will not match. That is,

$$\Pr\left[\mathrm{Imp}^{LA}_{\mathcal{A},\mathcal{Q}} = 1\right] = \Pr\left[sig \stackrel{?}{=} \widehat{sig}\right] = \Pr\left[\mathrm{H}(d \parallel code_l[T] \parallel \tau_r) = \mathrm{H}(\widehat{d} \parallel \widehat{code_l[T]} \parallel \widehat{\tau}_r)\right] \leq \frac{1}{\mid \mathbf{H} \mid} + \mathrm{negl}(n).$$

Thus, impersonation of *LA* by $\mathcal{A}$ fails. □

Lemma 3. *Consider adversary $\mathcal{A}$ attempting to execute run R' of $\mathcal{Q}$. Impersonation of C by an eavesdropping adversary $\mathcal{A}$ fails.*

10. Clearly, we are not considering external factors such as IP addresses in the determination of $\mathrm{Imp}^{P}_{\mathcal{A},\mathcal{Q}}$, as our focus is on the protocol.

Proof. As before, by eavesdropping on R, the following proposition holds:

$$\mathrm{Has}(\mathcal{A},\emptyset)[R]_{\mathcal{A}}\mathrm{Has}(\mathcal{A},\phi_{\mathcal{A},R}) \supset \mathrm{Has}(\mathcal{A},\{h\}).$$

Note that $h = \mathrm{H}(z \parallel y)$, where $z \in \phi_{\mathcal{A},R}$ and $y = \mathrm{H}(b \parallel pwd_c) \notin \phi_{\mathcal{A},R}$. Furthermore, $pwd_c \notin \phi_{\mathcal{A},R}$. As such, given the assumption that the hash function is collision resistant, $\mathcal{A}$ must resort to guessing $\widehat{h}$, and

$$\Pr[\mathrm{Imp}_{\mathcal{A}}(C)=1] = \Pr[\mathrm{Has}(\mathcal{A},\{\widehat{h}\})] \leq \frac{1}{|\mathbf{H}|} + \mathrm{negl}(n).$$

Thus, impersonation of C by $\mathcal{A}$ fails. □

Lemma 4. *Consider adversary $\mathcal{A}$ attempting to execute run R' of $\mathcal{Q}$. Impersonation of* SP *by an eavesdropping adversary $\mathcal{A}$ fails.*

Proof. If $\mathcal{A}$ believes that the object $\widehat{o} = o$ and the action $\widehat{a} = a$, then the simplest strategy is to bypass Protocol $\mathcal{Q}$ and simply return e_o to C. Note, though, that $\mathcal{A}$ must send *something* to *RA* and *LA* under the terms of $\mathrm{Imp}^{SP}_{\mathcal{A},\mathcal{Q}}$ (otherwise, the impersonation would immediately be detected). It turns out that $\mathcal{A}$ can deceive both *RA* and *LA* by sending random data, as neither principal ever sees a structured message in $\mathcal{Q}$. However, impersonation still fails, as, letting $n = |\mathbf{O}| \cdot |\mathbf{A}\,|$,

$$\Pr[\neg\{\mathcal{A} \stackrel{?}{=} SP\}_C] = \Pr[(o=\widehat{o}) \wedge (a=\widehat{a})] \leq \frac{1}{n} + \mathrm{negl}(n).$$

□

Lemma 5. *Attempting to impersonate any principal yields no information useful for future attacks.*

Proof. In the case of $\mathrm{Imp}^{RA}_{\mathcal{A},\mathcal{Q}}$ or $\mathrm{Imp}^{LA}_{\mathcal{A},\mathcal{Q}}$, note that $\phi_{\mathcal{A},R'} = \{\widehat{e_{sr}}, \widehat{\iota^{-\lambda m}}\}$ or $\{\widehat{e_{sl}}\}$, respectively. Without the corresponding secret key, $\mathcal{A}$ cannot decrypt the messages $\widehat{e_{sr}}$ and $\{\widehat{e_{sl}}\}$. Additionally, the IND-CPA security ensures that $\mathcal{A}$ cannot determine if this data decrypt identically as future such messages. Next, with no knowledge of ι, ι^{-1}, $\widehat{\lambda}$, or $\widehat{m}$, $\widehat{\iota^{-\lambda m}}$ is meaningless. Furthermore, $\widehat{m}$ is a nonce, so $\widehat{\iota^{-\lambda m}}$ provides no useful information. Now, consider $\mathrm{Imp}^{C}_{\mathcal{A},\mathcal{Q}}$. Observe that, in the middle of run R',

$$\mathrm{Has}(\mathcal{A},\phi_{\mathcal{A},R})\ [\mathrm{receive}\ \widehat{SP},\widehat{C},\widehat{z}]\ \mathrm{Has}(\mathcal{A},\phi_{\mathcal{A},R}\cup\{\widehat{z}\}).$$

Based on this information, $\mathcal{A}$ must be able to compute $\widehat{h}$, which we showed above was unlikely. Consequently, *SP* will detect the impersonation attempt when validating the hash and abort the protocol. Thus, with all but trivial probability,

$$\mathrm{Has}(\mathcal{A},\phi_{\mathcal{A},R})\ [R']\ \mathrm{Has}(\mathcal{A},\phi_{\mathcal{A},R}\cup\{\widehat{z}\}),$$

which contains no information useful for future attacks. If the hash guess is successful, the only additional information received by $\mathcal{A}$ is $\widehat{e_o}$. Again, this is based on an IND-CPA-secure encryption, and is not useful. Now consider $\mathrm{Imp}^{SP}_{\mathcal{A}\mathcal{Q}}$. We can formalize the maximum information learned by $\mathcal{A}$ as

$$\{(\tau_r,\tau_l,o,a),\widehat{e_{sr}},\widehat{e_s},\widehat{z},\widehat{h},\widehat{e_{OT}},\widehat{e_{sl}},\widehat{e'_s},\widehat{\lambda},\widehat{code_l[T]},\widehat{r}\}).$$

Again, most of the data (including τ_r and τ_l) are IND-CPA-secure or nonces, and provide no useful information for future attacks. The only pieces of data that could potentially be used for future attacks are $\widehat{\lambda}$ and $\widehat{code_l[T]}$. Since $i_l \notin \phi_{\mathcal{A},R'}$, $\widehat{\lambda}$ is simply a randomly selected location identifier $\lambda \in \mathbf{Z}^*_N$, which is public data. On the other hand, $\widehat{code_l[T]}$ is sensitive, as this could be used to forge the signature used in Protocol 2 by the *LD*. However, these are rolling codes and only valid for a short time frame, limiting the potential use by $\mathcal{A}$. Furthermore, forging the corresponding $\widehat{\tau_l}$ requires $i_l \in \phi_{\mathcal{A},R'}$, which is not the case. Consequently, this knowledge is useless for a future impersonation attack. □

Theorem 2. *Protocol $\mathcal{Q}$ is secure against impersonation attacks under the Dolev-Yao adversarial model.*

Proof. Follows directly from Lemmas 1-5. □

4.3 Preservation of Client Privacy

Before proceeding with the following lemmas, we emphasize here that our definition of privacy preservation focuses on the information exchanged in a single run of the protocol. As such, we do not consider attacks based on inference over time. That is, background knowledge of the unencoded policies may allow an adversary to aggregate the information gained from a significant number of access requests to break the privacy guarantees of our framework. Addressing this problem is beyond the scope of our current work, and we do not have a clear method for quantifying this threat. Furthermore, at this time, we are uncertain whether there exists a computationally feasible approach that would mitigate this threat. We leave such questions for future research.

Lemma 6. *$\mathcal{Q}$ preserves the privacy of* C *against* RA.

Proof. Recall $\phi_{RA,R} = \{\iota^{-\lambda m}\}$ with λ and m as large values unknown to *RA*. Additionally, the mapping from λ to locations is unknown to *RA*. Thus, for a randomly selected $\widehat{l}$ and $\widehat{\lambda}$

$$\begin{aligned}&\Pr[\{l \stackrel{?}{=} \widehat{l}\}_{RA} \mid \mathrm{Has}(RA,\phi_{RA,R})]\\ &\quad - \Pr[\{l \stackrel{?}{=} \widehat{l}\}_{RA} \mid \mathrm{Has}(RA,\emptyset)] \leq\\ &\Pr[\{\lambda \stackrel{?}{=} \widehat{\lambda}\}_{RA} \mid \mathrm{Has}(RA,\{\iota^{-\lambda m}\}) \wedge \neg\mathrm{Has}(RA,\{\lambda,m\})]\\ &\quad - \Pr[\{l \stackrel{?}{=} \widehat{l}\}_{RA} \mid \mathrm{Has}(RA,\emptyset)] \leq \mathrm{negl}(n).\end{aligned}$$

Similarly, for role r and a random role $\widehat{r}$, as encryption is IND-CPA-secure

$$\begin{aligned}&\Pr[\{r \stackrel{?}{=} \widehat{r}\}_{RA} \mid \mathrm{Has}(RA,\phi_{RA,R})]\\ &\quad - \Pr[\{r \stackrel{?}{=} \widehat{r}\}_{RA} \mid \mathrm{Has}(RA,\emptyset)] =\\ &\Pr[\{\tau_r \stackrel{?}{=} \widehat{\tau_r}\}_{RA} \mid \mathrm{Has}(RA,\{e_{sr},\widehat{e_{sr}}\})]\\ &\quad - \Pr[\{\tau_r \stackrel{?}{=} \widehat{\tau_r}\}_{RA} \mid \mathrm{Has}(RA,\emptyset)] \leq \mathrm{negl}(n).\end{aligned}$$

As *RA* cannot determine the role session, it cannot determine the user either. Thus, $\mathcal{Q}$ preserves the privacy of C against *RA*. □

Lemma 7. *$\mathcal{Q}$ preserves the privacy of* C *against* LA.

Proof. Recall $\phi_{LA,R} = \emptyset$. Then, trivially

$$\Pr\big[\mathrm{Priv}^{C}_{LA,\mathcal{Q}} = 1 \mid \mathrm{Has}(LA, \phi_{LA,R})\big] - \Pr\big[\mathrm{Priv}^{C}_{LA,\mathcal{Q}} = 1 \mid \mathrm{Has}(LA, \emptyset)\big] = 0 \leq \mathrm{negl}(n).$$

□

Lemma 8. *$\mathcal{Q}$ preserves the privacy of* C *against colluding principals* RA *and* LA.

Proof. Let $\mathcal{A}$ denote the adversary resulting from the collusion of *RA* and *LA*. Then,

$$\phi_{\mathcal{A},R} = \phi_{RA,R} \cup \phi_{LA,R} = \phi_{RA,R}.$$

Observe that $\mathcal{A}$ now has (from prior and learned knowledge), ι, γ, and $\phi(N)$, but PIR ensures that $\mathcal{A}$ does not have λ. Furthermore, even with ι and $\phi(N)$, the intractability of the discrete logarithm prevents $\mathcal{A}$ from learning λm from $\iota^{-\lambda m}$. In addition, OT prevents $\mathcal{A}$ from learning the role token (and, as a result, the role and user) used. Consequently, pooling the knowledge of *RA* and *LA* is of no use, and

$$\Pr\big[\mathrm{Priv}^{C}_{\mathcal{A},\mathcal{Q}} = 1 \mid \mathrm{Has}(\mathcal{A}, \phi_{RA,R} \cup \phi_{LA,R})\big] - \Pr\big[\mathrm{Priv}^{C}_{\mathcal{A},\mathcal{Q}} = 1 \mid \mathrm{Has}(\mathcal{A}, \emptyset)\big] \leq \mathrm{negl}(n).$$

□

Lemma 9. *Given the policy encoding scheme,* SP *can statically link two policies with the same role-location values with probability only negligibly better than guessing.*

Proof. Observe that, as a result of the RSA assumption, *SP* cannot compute $((\rho\lambda)^{-1}(\alpha\delta))^{-1} \bmod \phi(N)$, as the factorization of N is unknown to *SP*. This defense prevents linkability of policies across objects or actions. Consider the following three policies:

$$\widehat{p_{o.i}} = <(\rho\lambda)^{-1}(\alpha\delta), \iota^{\alpha\delta}>$$
$$\widehat{p_{o.j}} = <(\rho\lambda)^{-1}(\alpha\widehat{\delta}), \iota^{\alpha\widehat{\delta}}>$$
$$\widehat{p_{o.k}} = <(\widehat{\rho}\widehat{\lambda})^{-1}(\alpha\widehat{\delta}), \iota^{\alpha\widehat{\delta}}>.$$

Without knowledge of $\phi(N)$, *SP* cannot compute the multiplicative inverses and link $\widehat{p_{o.i}}$ with $\widehat{p_{o.j}}$. Furthermore, without knowledge of any of the factors or ι, *SP* cannot distinguish between $\widehat{p_{o.j}}$ and $\widehat{p_{o.k}}$ given knowledge of $\widehat{p_{o.i}}$. Thus,

$$\Pr[\{p_{o.i}[r] \stackrel{?}{=} p_{o.j}[r]\}_{SP} \vee \{p_{o.i}[l] \stackrel{?}{=} p_{o.j}[l]\}_{SP} \mid \mathrm{Has}(SP, \{\widehat{p_{o.i}}, \widehat{p_{o.j}}\})] -$$
$$\Pr[\{p_{o.i}[r] \stackrel{?}{=} p_{o.j}[r]\}_{SP} \vee \{p_{o.i}[l] \stackrel{?}{=} p_{o.j}[l]\}_{SP} \mid \mathrm{Has}(SP, \emptyset)] \leq \mathrm{negl}(n).$$

□

Lemma 10. *$\mathcal{Q}$ preserves the privacy of* C *against* SP *for a single access request.*

Proof. As $\lambda \in \phi_{SP,R}$, *SP* learns a small amount of information about C. However, without the mapping from λ to locations (known only to *LA*), for a randomly selected $\widehat{l}$,

$$\Pr[\{l \stackrel{?}{=} \widehat{l}\}_{SP} \mid \mathrm{Has}(SP, \phi_{SP,R})] - \Pr[\{l \stackrel{?}{=} \widehat{l}\}_{SP} \mid \mathrm{Has}(SP, \emptyset)] \leq \mathrm{negl}(n).$$

Similarly, *SP* does not have the mapping from ρ to roles. Furthermore, $\rho \notin \phi_{SP,R}$. Hence,

$$\Pr[\{r \stackrel{?}{=} \widehat{r}\}_{SP} \mid \mathrm{Has}(SP, \phi_{SP,R})] - \Pr[\{r \stackrel{?}{=} \widehat{r}\}_{SP} \mid \mathrm{Has}(SP, \emptyset)] \leq \mathrm{negl}(n).$$

Finally, consider an encoded policy:

$$\widehat{p_{o.i}} = <(\rho\lambda)^{-1}(\alpha\delta), \iota^{\alpha\delta}>.$$

As $\rho, \alpha, \delta, \iota, \phi(N) \notin \phi_{SP,R}$, *SP* cannot determine if this was the policy satisfied, even if it knows (by organizing the storage of policies) that $\iota^{\alpha\delta}$ indicates the object-action pair. Consequently, *SP* cannot trace $\phi_{SP,R}$ back to the original, unencoded $p_{o.i}$ that was satisfied. Thus,

$$\Pr[\mathrm{Priv}^{C}_{SP,\mathcal{Q}} = 1 \mid \mathrm{Has}(SP, \phi_{SP,R})] - \Pr[\mathrm{Priv}^{C}_{SP,\mathcal{Q}} = 1 \mid \mathrm{Has}(SP, \emptyset)] \leq \mathrm{negl}(n).$$

□

Lemma 11. *Given runs R and R' of $\mathcal{Q}$ made with location tokens τ_l and $\widehat{\tau}_l$,* SP *can link the requests if the location is identical.*

Proof. *SP* always sees λ in the clear, so the claim trivially holds when λ is identical. □

Corollary 11.1. *Location linkage threats can be mitigated by administrative action.*

Proof. At regular intervals, *IA* can regenerate the λ values as λ'. *IA* can provide *LA* with an updated mapping from λ' to location. Simultaneously, *IA* can regenerate the encoded policies $\widehat{p_{o.i}}$, sending the updated versions to each *SP*. While this process requires significant work on the part of *IA*, the policy regeneration can be performed offline before the update takes effect. Thus, *IA* can mitigate the threat of location linkage over time. □

Lemma 12. *Given runs R and R' such that $\rho = \widehat{\rho}$ and $\lambda = \widehat{\lambda}$, where ρ, λ are used in R and $\widehat{\rho}, \widehat{\lambda}$ are used in R', there is a strategy by which* SP *can link these requests, but it is probabilistically detectable by* RA.

Proof. Assume $\lambda = \widehat{\lambda}$. If *SP* repeats the value m, then $\phi_{SP,R} \cup \phi_{SP,R'}$ contains $\iota^{\rho\lambda m}$ and $\iota^{\widehat{\rho}\widehat{\lambda}m}$. Consequently, it is trivial for *SP* to determine if $\rho = \widehat{\rho}$. Thus, by repeating m when λ is the same, *SP* can state with absolute certainty that both requests used the same role and location. However, $\iota^{-\lambda m} \in \phi_{RA,R} \cap \phi_{RA,R'}$. Consequently, *RA* can state with absolute certainty that $-\lambda m = -\widehat{\lambda}\widehat{m}$, though this does not necessarily indicate the m values are identically. However, if c denotes the cardinality of the congruence class $[\widehat{m}]_{\bmod \phi(N)}$,

$$\Pr[-\lambda m = -\widehat{\lambda}\widehat{m} \mid \text{ only } \widehat{m} \text{ is random}] = \frac{c}{\mid \mathbf{Z}^{*}_{N} \mid}.$$

As this probability is small, such an equivalence is a likely indication of a repeated m, providing *RA* with a strong probabilistic guess that *SP* is not acting honestly. □

Corollary 12.1. *Collusion by multiple* SPs *is detectable by* RA.

Proof. Follows from Lemma 12, under the premise that R and R' (and additional such runs) are runs by the distinct, colluding *SPs*. Other than λ (which is not useful without the ability to map it to a location or a policy), the runs reveal no useful data unless m is identical. As such, *RA* can detect the repeated m value. □

Lemma 13. *Assuming* RA *prevents reuse of* m *for the same locations, with the exception of location,* $\mathcal{Q}$ *preserves the privacy of* C *against* SP, *even over time.*

Proof. First, note that repeated runs of $\mathcal{Q}$ from the same location l will repeatedly reveal λ to *SP*. Consider the sequence of runs $R_1, \ldots, R_n$ that satisfy policies $p_1, \ldots, p_n$, where λ is identical in all runs. If there is only a single location that satisfies $p_1, \ldots, p_n$, then *SP* can determine this location. However, from Lemmas 10-12 and Corollary 12.1, *SP* can only link the roles used in requests by repeating m for the same location value λ. Also, *RA* can detect this attempt with near certain probability (and a negligible false positive rate). As such, if *RA* prevents reuse (either by blacklisting corrupt *SPs* or by rejecting and requesting a new m), then *SP* can only determine that the location used in multiple requests were the same. Consequently, if we modify the privacy-preserving experiment to eliminate the discussion of location

$$\Pr\Big[\mathrm{Priv}^{C}_{SP,\mathcal{Q}} = 1 \mid \mathrm{Has}\Big(SP, \bigcup \phi_{SP,R_i}\Big)\Big] - \Pr\big[\mathrm{Priv}^{C}_{SP,\mathcal{Q}} = 1 \mid \mathrm{Has}(SP, \emptyset)\big] \leq \mathrm{negl}(n).$$

□

Lemma 14. $\mathcal{Q}$ *fails to preserve the privacy of* C *against an adversary* $\mathcal{A}$ *that consists of* SP *who colludes with either* RA *or* LA.

Proof. If *SP* colludes with *RA*, *SP* could simply send τ_r to *RA*, who decrypts the token to determine i_r, the index of the role session. *RA* can then retrieve σ_r, and use this information to find the corresponding ι^{ρ}, which *RA* can use to uniquely identify the role. Thus, $\mathcal{A}$ is guaranteed to correctly state $r = \widehat{r}$, and

$$\Pr\big[\mathrm{Priv}^{C}_{\mathcal{A},\mathcal{Q}} = 1\big] = 1.$$

If *SP* colludes with *LA*, *SP* can send τ_l to *LA* with similar results. Therefore, $\mathcal{Q}$ fails to preserve the privacy of C against an adversary $\mathcal{A}$. □

Theorem 3. *Assuming* SP *does not collude with* RA *or* LA, $\mathcal{Q}$ *preserves the privacy of* C *against* SP, RA, *and* LA.

Proof. The claim follows from above. □

5 Summary of Protocol Properties

The characteristics of the proposed framework can be summarized informally as follows:

- *RA* fails to learn which user, role, or location is involved in the request. For multiple distinct requests, *RA* is prevented from determining if the user, role, or location is the same in both.
- Similarly, *LA* fails to learn which user, role, or location is involved in the request. For multiple distinct requests, *LA* is prevented from determining if the user, role, or location is the same in both.
- Encrypted storage of role session records ensures that *SP* can only retrieve information for which it has an authentic τ_r.
- The policy encoding scheme ensures that *SP* cannot statically link encoded policies that have common characteristics (e.g., same role, location).
- Protocol $\mathcal{Q}$ protects the user's location information in the short term. Over time, *SP* may be able to determine the location by linking policies. However, without additional information, *SP* cannot link this to a particular user. Furthermore, *IA* can mitigate this threat by updating the λ values at regular intervals.
- Unless *SP* colludes with *RA* or *LA*, $\mathcal{Q}$ protects the privacy of the user's identity and role use, *even over time*.
- $\mathcal{Q}$ protects the user's privacy if *RA* and *LA* collude.
- Collusion by multiple *SPs* is detectable by *RA*, which can then abort the current run of $\mathcal{Q}$, thus stopping the attack.

6 Related Work

A number of works have considered the enforcement challenges for spatial and contextual access control policies [17]. In Cricket [18], user devices analyze their own distance from known beacons within an indoor space; this approach assumes user devices make honest location claims in their access requests. Other proposed solutions include the use of near-field communication (NFC) [19], [20] or RF-based sensors [21], [22], while others have focused on ensuring a user's contextual claim is consistent with those of other users in a mobile environment, such as a train [23]. Another approach includes the interposition of an access control engine *between* the user and the location service [24]. However, all of this work assumes the back-end servers are honest and does not attempt to protect the user's privacy against corrupted or malicious service providers.

Our work is related to the question of privacy-preserving queries in location-based services (LBS) [5] and k-nearest neighbor queries [25]. However, as our work focuses on access control, the security concerns are more stringent. For instance, in most cases of LBS, the use of location is to provide local information to the user; if the user provides a location that is not accurate, he would receive inaccurate results. In spatial access control, however, the service provider must ensure that the user's location claim is correct. In addition, our model also employs RBAC, which requires authentication of the role, as well.

Rabin introduced the notion of OT in [10], and the concept has been well studied in the literature. However, OT protocols (e.g., [26]) often have $\Theta(n)$ communication complexity. PIR schemes aim at greater communication efficiency while relaxing the security requirements; specifically, the server no longer has the guarantee that the client can retrieve only a single record. Chor et al. [27], [11] showed that there is no nontrivial PIR solution with a single server in an information-theoretic setting; however, they proposed a scheme with $K \geq 4$ noncolluding servers with $O(n^{1/\log K} K \log K)$ communication cost. This bound

was improved to $O(n^{\frac{\log\log K}{K\log K}})$ in [28]. More recently, Yekhanin [29] showed that if infinitely many Mersenne primes exist, then there is a three-server PIR protocol with $O(n^{1/\log\log n})$ communication complexity. Kushilevitz and Ostrovski [30] proposed a single-server computation PIR (cPIR) protocol with $O(n^{\epsilon})$ communication cost for arbitrarily small $\epsilon > 0$. Cachin et al. [31] improved this bound with a polylogarithmic communication protocol that relies on the ϕ-hiding assumption (ϕHA).

While PIR protocols retrieve a single bit at a time, Private Block Retrieval (PBR) schemes achieve more efficient performance by retrieving blocks at a time. Lipmaa [32] introduced a $O(\log^2 n)$ communication protocol, whereas Chan [33] used the Paillier [34] cryptosystem to achieve $O(\log n)$ complexity. Finally, Gentry and Ramzan [35] introduced a constant rate, communication-efficient PBR protocol that can be used under several distinct intractability assumptions, including ϕHA.

7 Conclusions

In this work, we proposed a privacy-preserving framework for evaluating spatially aware RBAC policies. We defined an architecture with reasonable computation assumptions, and specified protocols for evaluating the requests. We formally modeled the main request protocol using PCL. Using this formal model, we proved that our framework is secure against external attacks, and we also proved that our protocols preserve the privacy of users for individual requests. In addition, the protocols can even protect users' privacy over time. Finally, we highlighted the similarities and differences between our scheme and attribute-based encryption, with the main difference being that our scheme requires *transient* credentials (in contrast to persistent static credentials for attribute-based encryption) that are possessed by users for a single request at a time before reuse by others.

We propose three directions for future work. First, we find the notion of transient attribute credentials to be very powerful; applying this concept to other domains could produce desirable results. Second, our work demonstrates the feasibility of reconciling robust privacy guarantees with strong security requirements; this raises the question of what other settings could benefit from such results. Finally, our security and privacy guarantees rely on two assumptions: the presence of a trusted third party (*IA*) and there is no collusion between *SP* and the authority servers (*RA*, *LA*, and *IA*). A solution that reduces these assumptions would be very attractive and would aid in the adoption of this work.

8 Formal Notation

Summary of selected notation	
$\mathbb{A}, \mathbb{L}, \mathbb{O}, \mathbb{P}, \mathbb{R}, \mathbb{S}$	set of actions, locations, protected objects, principals, roles, and subjects
$\leftarrow$	probabilistic value assignment
$:=$	deterministic value assignment
$\supset$	logical implication
$\rho, \lambda, \delta, \alpha$	integer encoding of a role, location, object (document), and action
ι	a system-wide constant known only to *IA*
γ	a constant specific to *RA*
$p_{o.i}, \widehat{p_{o.i}}$	a policy $< r, l, a >$ granting access for role r to perform action a on object o in location l, and its encoded version: $< (\rho\lambda)^{-1}(\alpha\delta) \mod \phi(N), \iota^{\alpha\delta} \mod N >$
$\widehat{x}$	a value interpreted as the variable x; $\hat{x}$ may or may not have the same value as a known value of x
τ_r, τ_l	role and session token
σ	a role session identifier
$\phi(N)$	Euler's totient function
$\phi_{P,R}$	set of information known to P after run R of protocol $\mathcal{Q}$
$\theta(P)$	set of information known to P prior to a protocol run
$\{x \stackrel{?}{=} y\}_P$	P can correctly determine if x and y are equal
$\mathsf{Has}(P, S)$	P possesses the data in set S
$A\ [R]_P\ B$	P executes run R of protocol $\mathcal{Q}$; A is true prior to the run, and B is true after

Acknowledgments

This work has been partially supported by Sypris Electronics and by the MURI award FA9550-08-1-0265 from the Air Force Office of Scientific Research.

References

[1] M.L. Daimani, E. Bertino, B. Catania, and P. Perlasca, "GEO-RBAC: A Spatially Aware RBAC," *ACM Trans. Information and System Security*, vol. 10, pp. 1-34, 2007.

[2] F. Hansen and V. Oleschuk, "SRBAC: A Spatial Role-Based Access Control Model for Mobile Systems," *Proc. Eighth Nordic Workshop Secure IT Systems (NORDSEC)*, pp. 129-141, Oct. 2003.

[3] S. Aich, S. Sural, and A.K. Majumdar, "STARBAC: Spatiotemporal Role Based Access Control," *Proc. OTM Conf. the Move to Meaningful Internet Systems*, 2007.

[4] I. Ray, M. Kumar, and L. Yu, "LRBAC: A Location-Aware Role-Based Access Control Model," *Proc. Int'l Conf. Information Systems Security (ICISS)*, pp. 147-161, 2006.

[5] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private Queries in Location Based Services: Anonymizers Are Not Necessary," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, pp. 121-132, 2008.

[6] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," *Proc. IEEE Symp. Security and Privacy*, pp. 321-334, 2007.

[7] A. Sahai and B. Waters, "Fuzzy Identity Based Encryption," *Proc. Workshop the Theory and Application of Cryptographic Techniques (EUROCRYPT '05)*, vol. 3494, pp. 457-473, 2005.

[8] C.S. Jensen, H. Lu, and B. Yang, "Graph Model Based Indoor Tracking," *Proc. 10th Int'l Conf. Mobile Data Management (MDM)*, pp. 122-131, 2009.

[9] C.S. Jensen, H. Lu, and B. Yang, "Indoor—A New Data Management Frontier," *IEEE Data Eng. Bull.*, vol. 33, no. 2, pp. 12-17, June 2010.

[10] M.O. Rabin, "How to Exchange Secrets with Oblivious Transfer," Technical Report TR-81, Aiken Computation Lab, Harvard Univ., 1981.

[11] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private Information Retrieval," *J. ACM*, vol. 45, pp. 965-981, Nov. 1998.

[12] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. Chapman & Hall/CRC, 2008.

[13] J. Camenisch, M. Dubovitskaya, and G. Neven, "Oblivious Transfer with Access Control," *Proc. 16th ACM Conf. Computer and Comm. Security (CCS)*, pp. 131-140, 2009.

[14] T.P. Pedersen, "Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing," *Proc. 11th Ann. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO)*, pp. 129-140, 1992.
[15] D. Dolev and A. Yao, "On the Security of Public-Key Protocols," *IEEE Trans. Information Theory*, vol. IT-29, no. 2, pp. 198-208, Mar. 1983.
[16] A. Datta, A. Derek, J.C. Mitchell, and A. Roy, "Protocol Composition Logic (PCL)," *Electronic Notes Theoretical Computer Science*, vol. 172, pp. 311-358, Apr. 2007.
[17] C. Ardagna, M. Cremonini, S. De Capitani di Vimercati, and P. Samarati, "Access Control in Location-Based Services," *Privacy in Location-Based Applications*, C. Bettini, S. Jajodia, P. Samarati, and X. Wang, eds., pp. 106-126, Springer, 2009.
[18] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," *Proc. MobiCom*, pp. 32-43, 2000.
[19] M.S. Kirkpatrick and E. Bertino, "Enforcing Spatial Constraints for Mobile Rbac Systems," *Proc. 15th ACM Symp. Access Control Models and Technologies (SACMAT)*, pp. 99-108, 2010.
[20] L. Bauer, L.F. Cranor, M.K. Reiter, and K. Vaniea, "Lessons Learned from the Deployment of a Smartphone-Based Access-Control System," *Proc. Third Symp. Usable Privacy and Security (SOUPS)*, 2007.
[21] N. Sastry, U. Shankar, and D. Wagner, "Secure Verification of Location Claims," *Proc. Second ACM Workshop Wireless Security (WiSe)*, pp. 1-10, 2003.
[22] P. Bahl and V.N. Padmanabhan, "RADAR: An In-building RF-based User Location and Tracking System," *Proc. IEEE INFOCOM*, pp. 775-784, 2000.
[23] M.J. Covington, W. Long, S. Srinivasan, A.K. Dev, M. Ahamad, and G.D. Abowd, "Securing Context-Aware Applications Using Environment Roles," *Proc. Sixth ACM Symp. Access Control Models and Technologies (SACMAT)*, pp. 10-20, 2001.
[24] C.A. Ardagna, M. Cremonini, E. Damiani, S.D.C. di Vimercati, and P. Samarati, "Supporting Location-Based Conditions in Access Control Policies," *Proc. First ACM Symp. Information, Computer and Comm. Security (ASIACCS)*, 2006.
[25] B. Yang, H. Lu, and C.S. Jensen, "Probabilistic Threshold k Nearest Neighbor Queries over Moving Objects in Symbolic Indoor Space," *Proc. 13th Int'l Conf. Extending Database Technology (EDBT)*, pp. 335-346, 2010.
[26] M. Naor and B. Pinkas, "Oblivious Transfer and Polynomial Evaluation," *Proc. 31st Ann. ACM Symp. Theory of Computing (STOC)*, pp. 245-254, 1999.
[27] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private Information Retrieval," *Proc. 36th Ann. Symp. Foundations of Computer Science (FOCS)*, pp. 41-50, 1995.
[28] A. Beimel, Y. Ishai, E. Kushilevitz, and Jean-Fran, "Breaking the Barrier for Information-Theoretic Private Information Retrieval," *Proc. 43rd Ann. Symp. Foundations of Computer Science (FOCS)*, pp. 261-270, 2002.
[29] S. Yekhanin, "Locally Decodable Codes and Private Information Retrieval Schemes," PhD dissertation, MIT, 2007.
[30] E. Kushilevitz and R. Ostrovsky, "Replication Is Not Needed: Single Database, Computationally-Private Information Retrieval," *Proc. 38th Ann. Symp. Foundations of Computer Science (FOCS)*, pp. 364-373, 1997.
[31] C. Cachin, S. Micali, and M. Stadler, "Computationally Private Information Retrieval with Polylogarithmic Communication," *Proc. 17th Int'l Conf. Theory and Application of Cryptographic Techniques (EUROCRYPT '99)*, pp. 402-414, 1999.
[32] H. Lipmaa, "An Oblivious Transfer Protocol with Log-squared Total Communication," *Proc. Information Security Conf. (ISC)*, pp. 314-328, 2005.
[33] Y.-C. Chan, "Single Database Private Information Retrieval with Logarithmic Communication," *Proc. Australasian Conf. Information Security and Privacy (ACISP)*, pp. 50-61, 2004.
[34] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," *Proc. 17th Int'l Conf. Theory and Application of Cryptographic Techniques (EUROCRYPT '99)*, pp. 223-238, 1999.
[35] C. Gentry and Z. Ramzan, "Single-Database Private Information Retrieval with Constant Communication Rate," *Proc. Int'l Colloquium Automata, Languages and Programming (ICALP)*, pp. 803-815, 2005.

Michael S. Kirkpatrick received the graduate degrees from Purdue University and Michigan State University. He is an assistant professor of computer science at James Madison University. His main research interests focus on security engineering, access control, applied cryptography, privacy, and secure operating systems. Prior to earning graduate degrees from Purdue University and Michigan State University, he spent several years working in the field of semiconductor optical proximity correction for IBM Microelectronics (later Server & Technology Group) in Essex Junction, VT. He has served as a reviewer for a number of top journals and conferences, including *IEEE TDSC*, *IEEE S&P*, IEEE ICDM, ACM SACMAT, ACSAC, *IEEE TVLSI*, *ACM TISSEC*, and IFCA Financial Cryptography & Data Security.

Gabriel Ghinita is an assistant professor with the Department of Computer Science, University of Massachusetts, Boston. His research interests lie in the area of data security and privacy, with focus on privacy-preserving transformation of microdata, private queries in location-based services, and privacy-preserving sharing of sensitive data sets. Prior to joining University of Massachusetts, he was a research associate with the Cyber Center at Purdue University, and a member of the Center for Education and Research in Information Assurance and Security (CERIAS). He served as a reviewer for top journals and conferences such as *IEEE TPDS*, *IEEE TKDE*, *IEEE TMC*, *VLDBJ*, *VLDB*, WWW, ICDE, and ACM SIGSPATIAL GIS.

Elisa Bertino is a professor of computer science at Purdue University, and serves as a research director of the Center for Education and Research in Information Assurance and Security (CERIAS) and an interim director of Cyber Center (Discovery Park). Previously, she was a faculty member and department head at the Department of Computer Science and Communication of the University of Milan. Her main research interests include security, privacy, digital identity management systems, database systems, distributed systems, and multimedia systems. She received the 2002 IEEE Computer Society Technical Achievement Award for outstanding contributions to database systems and database security and advanced data management systems and the 2005 IEEE Computer Society Tsutomu Kanai Award for pioneering and innovative research contributions to secure distributed systems. She is a fellow of the IEEE and a fellow of the ACM.