Location-Aware Authentication and Access Control - Concepts and Issues

Elisa Bertino
CS Department and CERIAS
Purdue University
West Lafayette, IN, USA
bertino@cs.purdue.edu

Michael Kirkpatrick
CS Department
Purdue University
West Lafayette, IN, USA
mkirkpat@cs.purdue.edu

***Abstract*—The paper first discusses motivations why taking into account location information in authentication and access control is important. The paper then surveys current approaches to location-aware authentication, including the notion of context-based flexible authentication policies, and to location-aware access control, with focus on the GEO-RBAC model. Throughout the discussion, the paper identifies open research directions.**

Security; privacy; distributed systems; mobile applications

I. Introduction

The combined use of Internet, the Web and mobile technologies (e.g. mobile devices, mobile and wireless communication) makes it possible for users to connect to remote resources and services from a wide range of settings. Such pervasive, highly-connected environments make not just routine tasks, such as checking e-mail and bank accounts, more convenient, but also activities that are more bandwidth expensive, like conference calls and virtual meetings. Also, such environments make it possible for organizations and governmental agencies to quickly react to unforeseen situations and emergencies.

Although the main purpose of such environments and connectivity is to increase the availability of information and services to users, security is also a crucial requirement. Many applications that benefit from mobility and enhanced connectivity need to access sensitive information and services that need to be adequately protected. Examples of such applications include those in healthcare, law enforcement, finances, border control, and homeland security. Securing information and resources in such settings involves not only protecting network communications, but also providing strong authentication and access control, as they are crucial to secure the end-points of computing infrastructures and to make sure that information and services are used according to the organizational policies and legal requirements.

Authentication typically refers to verifying that a certain user wishing to use a login name to connect to a computer system is the actual user to which the login name has been assigned. It also often refers to verifying the credentials presented by a user, or by a client acting on behalf of a user. Access control refers to verifying that a user has the permission to access a specific piece of information or use a service. Access control basically checks each access request to protected resources against a set of authorizations, specified through an access control policy language. If an authorization exists matching the request, the request is allowed; it is denied otherwise.

Because of their relevance in security solutions, authentication and access control have been widely investigated and a large number of sophisticated techniques are available as well standards, such as the XACML [15] access control standard by Oasis. However, most of those approaches have been designed without considering location and, more generally, context, as a relevant factor. Conventional access control mechanisms, for example, are based on the assumption that the requesters' profiles and identity information, such as login names, fully determine what these users can do. However, when dealing with mobility, the fact that a user is in a given location is also crucial for determining what the user can do. For example, a user, who from his/her office has access to highly confidential documents, should most likely be disallowed to access them while traveling on a crowded subway or when in an insecure location. Location thus becomes an important factor to be considered when making access control decisions. Location also affects authentication; for example, stronger authentication may be required when a user connects to the system from a location that is unusual for the user. In such a case, the location determines the actual authentication policy to be applied.

Because location information can be relevant to security solutions, a few approaches to location-aware authentication and access control have been proposed [1, 10, 11]. However these approaches have mainly focused on the conceptual level. There is not even a comprehensive view of what issues are relevant when implementing and deploying such approaches.

The goal of this paper is to discuss key proposed approaches to location-aware authentication and access control, to identify relevant challenges, and outline open research directions. The rest of this paper is organized as follows. Section 2 discusses authentication and introduces approaches for verifying user location and then discusses the notion of authentication policies. Section 3 discusses extensions to conventional access control mechanisms to support location awareness. Sections 4 and 5 discuss open research issues and outline few conclusions, respectively.

II. Authentication

There are two distinct ways to use location information for authentication. The first is to use the location as one component in the authentication process. The use of location, then, becomes another version of multi-factor authentication, where by multi-factor authentication we refer to providing multiple authentication proofs. The main challenge in using location in this manner is that the user may not be trusted to report his location. Consequently, the design must include mechanisms that prevent the user from forging this information. The second use of location information would be to determine the security policy. For example, if a user is connecting to his company's network from home, he may be required to present more credentials than if he were connecting from his office. In this view, the location information is external to the policy, whereas location is incorporated into the policy in the former approach. In what follow we elaborate on those different uses.

A. Verification of User's Location

Verifying the user's location can be done in a number of different ways. A first approach would be to use a physical coordinate system, such as GPS. The user's device would establish the coordinates and send them along with the access request. One drawback of this technique, though, is that it requires the system designers trust the user's device to report the data accurately. An attacker could supply forged location information to get illicit access to the requested object.

As an alternative, the system could require logical location information, rather than physical. For example, if the security policy requires the user to be present in a particular room, a small certifier device could be permanently installed. The user could then connect to this certifier to get a token that proves the user's location in the room. Provided that cryptographic techniques are in place to prevent forgeries, this approach could be used to satisfy the location-aware security policy.

A third technique for location-based authentication would be to restrict the access to an immobile device. For example, assume that an organization has offices in multiple cities. Each office has a secure room with restricted access. Users in each city need to share sensitive data between these rooms exclusively. To accomplish this goal, a workstation is installed in the secure rooms, and the system binds the authentication process to this workstation.

Binding authentication to the workstation can be accomplished in different ways. One approach would be to use a Trusted Platform Module (TPM) [5] to store a cryptographic key. This key would only be provided to trusted applications that prevent the key from being leaked. Consequently, the user would be unable to transfer the key to a different machine to bypass the security policy.

Another hardware-based technique is the use of physically unclonable functions (PUF) [6,7,8,9]. It is an inherent limitation of the manufacturing process that no two physical devices can be made to be perfectly identical. Each physical device provides a unique signature that can be quantified and measured. PUF technology turns this flaw into an advantage. For example, to protect a cryptographic key k, PUFs combine the key with the device specific measurement m by computing the bitwise XOR $k + m = x$. The result x is then stored and the key k is discarded. Then, at run-time, x is combined with m to re-create the key k. Depending on the application, x could be stored on the workstation or on the remote server.

As each technique offers its own advantages and disadvantages, more research is needed to assess the strength and usability of each such technique and to efficiently integrate them in authentication protocols.

B. Authentication Policies

Many application environments require authentication with varying degrees of strengths depending on the user context, and in particular the user location, and the resources that subjects need to access. Strong authentication is a key to ensure accountability and security in today systems and applications. However, strong authentication can be expensive to deploy and complex to manage. For example, adopting one-time passwords [13, 14] for all users of an organization, independently from the tasks they have to perform and the resources they have to access, may be very expensive; ideally one would like to require such types of authentication only for users who need to access sensitive resources and use conventional passwords for other users.

We thus need flexible authentication mechanisms driven by *context-based authentication policies*. Such policies are high-level directives, expressed in a language that the authentication mechanism is able to understand, specifying the type of authentication required from users, depending also on location information about users and on other context information. During a recent discussion, the Head of the Information System division of a major healthcare organization in Indiana proposed an interesting application scenario. In this environment, an important requirement concerning authentication for doctors is that "authentication should be as quick as possible and have high usability – the use of proximity badges may be sufficient." If, however, "the doctor is not in the place where he/she usually is, a stronger authentication should be applied." Supporting such requirements implies the need for multiple authentication policies, each characterized by a specific context for application.

A preliminary authentication policy language addressing such requirement has been developed by Squicciarini et al. [12]. Such language is based on the notion of *Authentication Factor*. Authentication factors define the features of a specific authentication, where by *authentication* we intend the execution of one authentication protocol using a single mechanism. Authentication factors are specified in terms of one or a combination of *descriptors*. A descriptor is essentially a predicate expressing a property required for the authentication factor. A simplified example of an authentication factor specified

by two descriptors is {Mechanism = Biometrics, Feature = Fingerprint}. The first descriptor requires the authentication be based on the use of a biometric authentication mechanism, whereas the second descriptor specifies that the feature used in the biometric authentication must be the fingerprint. The language also supports temporal descriptors that are used to specify "freshness conditions" concerning authentication; these conditions specify how recent an authentication factor must be. An authentication policy is associated with a specific protected resource (or group of resources) and consists of one or more factors; as such the language directly supports the specification of multi-factor authentication. The language also supports the specification of a context, including location, within a policy. Therefore, the policy is applied only if the actual context of the user matches the policy context.

Supporting such policies requires addressing many issues. First one has to determine the possible user locations, in order to specify policies for all such locations; different approaches are possible, like having default policies that are always applied, unless policies exist that are specific to certain locations. Second since users move, and thus location may change, different policies may need to be applied. The minimization of user authentication steps is however important for usability reasons. Collecting accurate user locations, as discussed in the previous subsection, is also important requirement for the effective enforcement of location-aware authentication policies.

III. Access control

Access control represents a key component of any comprehensive solution to computer security and data privacy. Despite the large number of approaches, location-aware access control is still an open research question with many challenging issues.

In this section after a short overview of basic notions in access control, we introduce role-based access control (RBAC) [2,16], one of the most widely adopted access control approaches, and then GEO-RBAC, which extends RBAC with location-awareness and has been the first access control approach to take location into account. We then conclude the section with a discussion of the broader notion of event-based access control.

A. Basics of Access Control and Selected Approaches

Access control is a key component in that it determines which subject can perform which action under which circumstances on the protected resources and services. Whenever a subject tries to access a resource, the access control service checks the rights of the subject against a set of *authorizations*, usually stated by some security administrator. Authorizations thus encode the *access control policies* of a given organization. Any system that offers any level of security will ensure that the access control service intercepts all requests by subjects in order to ensure that all these requests are properly authorized before the subject gains access to the requested resource.

Figure 1. General Architecture of Access Control

Figure 1 shows a general architecture for an access control service. The *reference monitor* intercepts each access request in order to determine whether it can be authorized. The reference monitor typically requires different types of information to make such a decision. It needs to know which subject is requesting access to which resource. Generally the subject is determined by authenticating each user and associating each process, or applications, that the user runs with the user's identity. The resource is determined from the request itself and by consulting information maintained by the resource manager, for example the operating system or the database management system (DBMS), about the protected resources. The reference monitor also needs to know the type of action the subject wishes to execute on the protected resources. The more common actions include read, write and execute.

Such a simple notion of access control has been widely extended; a notable extension has been the introduction of role-based access control (RBAC) under which authorizations for access to the protected resources are not assigned to users directly but to different entities, called *roles*, representing functional units within an organization or application domain. Another notable extension is represented by the notion of attribute-based access control, under which properties, referred to as attributes and often collected into profiles, characterizing the users and/or the protected resources are taken into account in access control decisions. By using such an approach one can express policies specifying that a user can access a resource only if a certain attribute in the user profile verifies a given condition. Such approaches support high-level specification of policies and simplify access control management and interoperability across multiple domains. Another important extension is represented by location-based access control; such an approach extends conventional attribute-access control by including location of the requesting user among the information used to make access control decision.

B. Role-Based Access Control (RBAC)

A crucial problem in the deployment of access control services is the administrative costs for the maintenance of access control lists or other similar access control data structures. RBAC attempts to reduce such costs. It is based on by the notion of *role*, which acts an intermediary between users and permissions. Typically roles are associated with job descriptions and, because there are many fewer roles than users, considerable savings in administration can be achieved. The basic concepts of RBAC, which has been

standardized by ANSI [16], are shown in Figure 2. RBAC includes two main components: the *core* component, which does not include role hierarchies, and the *hierarchical* component, which does.

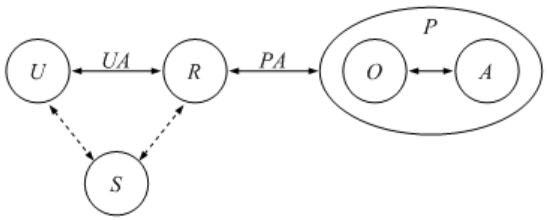

Figure 2. Diagram of Core RBAC

The RBAC model is based on a set of users *U*, a set of permissions *P*, and a set of roles *R*. A permission is usually a resource-action pair, where an action is synonymous with an access right. Users are associated with roles using a user-role assignment relation *UA*. This relation is a set of pairs of the form *(u,r)*, meaning that user *u* is assigned to role *R*. Permissions are similarly associated with roles using a permission-role assignment relation *PA*. Users interact with an RBAC system by activating a *session*, from a set *S* of sessions. Typically the user authenticates to the system and chooses to act in one or more of the roles to which he is assigned. RBAC further reduces the administration costs by introducing the notion of *role hierarchy*, which is represented as a directed acyclic graph in which the roles are nodes. The basic idea is that a role high up in the hierarchy will inherit the permissions of lower roles, without having to be explicitly assigned to those permissions. Clearly, this significantly reduces the number of permissions that need to be assigned to more senior roles, thereby reducing the administrative overheads in an RBAC system. Some important extensions to RBAC include static and dynamic separation of duties, constraining the roles that can be assigned and used by users, respectively.

C. GEO-RBAC

GEO-RBAC extends the RBAC model by incorporating spatial awareness. As in traditional RBAC, users are assigned to roles in GEO-RBAC, and object permissions are assigned according to roles. For example, access to a set of health records could be restricted to the role of *Doctor*. Any user that can provide credentials to activate the *Doctor* role would then have access to the data. In GEO-RBAC roles are associated with particular spatial extents; such extents represent spatial regions, in the reference space, that are of interest to the application domain. Locations in GEO-RBAC are logical concepts, such as “in room 314” or “at home,” rather than physical coordinates, like GPS. A mapping function is used to translate the physical position of the user, expressed by the physical coordinates, onto one or more logical position. Note that there could be several logical positions corresponding to a given physical position. Each role, thus, has an associated a mapping function.

Following the model of the Open GIS Consortium, the space of possible locations is defined according to geometric properties, such as lines, points, and polygons. *Features*, such as rooms and buildings, are defined by the borders, or *spatial extents*, between features. *Spatial roles* are defined to be the combination of a role and the extents enclosing the region on which the role is defined. *Permissions* are defined to be the pairing of objects and operations on those objects. The access control policy then maps permissions to spatial roles.

Returning to the example of the health care system, access to a patient's record could be restricted to the spatial role *Doctor-in-Ward*. If the same user logs in from his home office, he would be assigned the spatial role *Doctor-at-Home*. In either case, the credentials and authentication process are identical. The only difference is the factor of the location.

Because many roles may be associated with the same type of features, GEO-RBAC actually distinguishes between the notions of role schema and role instance. A role schema specifies a role name, for example *Doctor-in-Ward*, and a feature type, for example *Hospital*. A role instance is obtained from a role schema by instantiating the feature type to a specific feature; an example of role instance of the role schema <*Doctor-in-Ward*, *Hospital*> is <*Doctor-in-Ward*, *Saint-Elizabeth*>, where *Saint-Elizabeth* is an instance of the feature type *Hospital*, that is, an actual hospital. Notice that this approach allows one to support fine-grained user assignments to roles, in that users are assigned to role instances and not to role schema. In our example, if a user is only assigned to the <*Doctor-in-Ward*, *Saint-Elizabeth*> role instance, this user cannot use the role *Doctor-in-Ward* at other hospitals. Role schemas also allow one to factorize the definition of the mapping functions into the schema. The definition of GEO-RBAC is summarized by the diagram in Figure 3; such diagram only reports core GEO-RBAC. We refer the reader to [1] for additional details.

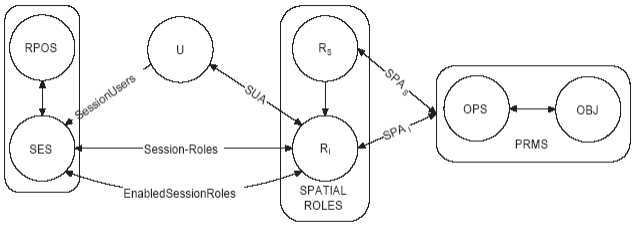

Figure 3. Diagram of Core GEO-RBAC

One novel feature of GEO-RBAC is the distinction between *role enabling* and *role activation*. A role is automatically enabled when the user enters the region associated with a spatial role. When the user leaves the region, the role is then disabled. Role activation, on the other hand, occurs when the user chooses to use that role during the current session. The role is deactivated either when the user chooses to do so or when the user leaves the extents of the spatial role. That is, a role can only be active if it is currently enabled.

D. Towards Event-based Access Control

Although much work has been done to develop abstract models for location-aware access control and also on context-aware access control, there is an interesting problem that has not yet been addressed. Specifically, what happens when things change has only been discussed in very basic terms. For example, in GEO-RBAC, a role is automatically deactivated and disabled when the user leaves the extents of the role. However, this is only one example of how the settings can change. Other examples include a user entering the spatial range of a conflicting role, or another authorized user attempting to activate the same role in the same location.

The notions of role enabling and activation, introduced as part of GEO-RBAC, can be seen as specific cases of the changes of state in the lifecycle of a role. Transitions among a role's states are triggered by specific events. In the case of GEO-RBAC, events are simple changes to the location information, with users moving in to or out of the extents associated with roles. It would be interesting to generalize such notion and introduce additional events and related constraints. Examples of events would include context changes, like emergencies arising or other entities moving inside the extent of a role, in which a user is already located and using the role. An example of a constraint would be that a role can be used by a given user if no other user is present in the role extent (referred to as *absence constraint*). Enforcement of such constraints would require an event detector able to detect the occurrence of events that could violate the constraints, thus resulting in the role disconnection.

IV. Open Research directions

The field of location-aware authentication and access control offers a wide range of directions for future research. First, a number of models, such as GEO-RBAC, have been proposed for reasoning about incorporating spatial awareness into role-based access control. However, these models have focused on high-level abstractions. More work is needed to bridge the gap between these models and real implementations. For example, developing general architectures that define the principals and protocols involved remains an open problem.

Another area of research is how to provide reliable location information. As we mentioned previously, GPS is one approach, but it may not provide adequate security guarantees. Testing for proximity to a location-certifying device can be accomplished using technologies such as CellID, Bluetooth or Near-Field Communication (NFC) [3]. However, these technologies have security concerns of their own, such as eavesdropping attacks and collusion [4]. Physically unclonable functions (PUF) offer great promise in binding authentication to a physical device. However, it is still just an experimental technology and has only been implemented for FPGAs and SRAMs, to the best of our knowledge. Thus, there is not yet an appropriate general-purpose technology that provides location information while satisfying the security requirements for access control. It is an open question of whether existing technology can be adapted to provide these guarantees.

The incorporation of risk and uncertainty into access control policies introduces an additional number of challenges for researchers. For example, if multiple devices are used to determine the user's location, it is not clear how to resolve conflicts if they arise between the devices. If location is used to determine the access control policy, there may be times when it would be desirable to grant an access even if the user cannot produce all of the required credentials. As an example scenario, given the time-critical nature of an emergency response, postponing the access until the user has a chance to gather the credentials may have negative consequences. However, these are the exceptional cases. Identifying when to grant or deny access given such uncertainty is a complex subject that requires more research.

Finally, the use of location information in access control opens up the possibility of invasions of a user's privacy. Depending on the application scenario, the user may wish to reveal his location only for the authentication, but would not want any user, even an administrator, to have access to this information. The design of any location-aware access control implementation should take care to protect the privacy of the user. One possible approach could include cryptographic techniques, such as zero-knowledge proofs, that would allow the user to prove their location but without actually having to reveal it, although accomplishing this goal is far from trivial. The design of such privacy protections offers an opportunity for future research.

V. concluding remarks

In this paper we have discussed recent approaches taking into account location information for authentication and authorization. We have also outlined relevant research challenges. Despite such initial work, the use of location remains however an open problem with many modeling and system issues. Work is needed to devise relevant requirements from real world applications and to understand all issues that need to be addressed to develop actual location-aware authentication and authorization systems. Work is also needed to develop systems able to take into account other context information about users, including current user tasks and past user access history. Ultimately, we will achieve stronger security if security solutions are well integrated into the user ambient and context.

Acknowledgment

The material reported in this paper is based in part upon work supported by the National Science Foundation under grant 0430274 and by the AFOSR grant A9550-08-1-0260.

References

[1] M. L. Damiani, E. Bertino, B. Catania, and P. Perlasca, “GEO-RBAC: A Spatially Aware RBAC,” ACM Transactions on Information and System Security, vol. 10, Feb. 2007, doi:10.1145/1210263.1210265.

[2] R. Sandhu, “Role-Based Access Control Models,” IEEE Computer, vol. 29, issue 2, Feb. 1996, pp. 38-47.

[3] NFC Forum Tag Type Technical Specification, http://www.nfc-forum.org/.

[4] G. Madlmayr, J. Langer, C. Kantner, and J. Scharinger, "NFC Devices: Security and Privacy," Proc. 3rd International Conference on Availability, Reliability and Security (ARES), 2008, pp. 642-647.

[5] Trusted Computer Group, "Trusted Platform Module Main Specification," http://www.trustedcomputinggroup.org/.

[6] M. J. Atallah, E. D. Bryant, J. T. Korb, and J. R. Rice, "Binding Software to Specific Native Hardware in a VM Environment: The PUF Challenge and Opportunity," Proc. Workshop on Virtual Machine Security (VMSec), 2008.

[7] G. E. Suh and S. Devadas, "Physcal Unclonable Functions for Device Authentication and Secret Key Generation," Proceedings of the 44th IEEE Design Automation Conference (DAC), 2007, pp. 9-14.

[8] J. Guajardo, S. S. Kumar, G.-J. Schrijen and P. Tuyls, "Physical Unclonable Functions and Public-Key Crypto for FPGA IP Protection," Proc. International Conference on Field Programmable Logic and Applications, 2007, pp. 189-195.

[9] J. Guajardo, S. S. Kumar, G.-J. Schrijen and P. Tuyls, "FPGA Intrinsic PUFs and Their Use for IP Protection," Proceedings of the 9th Cryptographic Hardware and Embedded Systems Workshop (CHES), 2007, pp. 63-80.

[10] D. Kulkarni and A. Tripathi, "Context-Aware Role-based Access Control in Pervasive Computing Systems," Proceedings of the 14th Symposium on Access Control Models and Technologies (SACMAT), 2008.

[11] R. J. Hulsebosch, A. H. Salden, M. S. Bargh, P. W. G. Ebben and J. Reitsma, "Context Sensitive Access Control," Proceedings of the 11th Symposium on Access Control Models and Technologies (SACMAT), 2005, pp. 111-119.

[12] A. Squicciarini, A. Bhargav-Spantzel, E. Bertino, and A. B. Czeksis, "Auth-SL – A System for the Specification and Enforcement of Quality-Based Authentication Policies," Proceedings of the 9th International Conference on Information and Communications Security (ICICS), Dec. 2007.

[13] RSA SecureId, http://www.rsasecurity.com/node.asp?id=1156/.

[14] ActivIdentity, http://www.actividentity.com/en/products/4_3_3_tokens.php/.

[15] OASIS eXtensible Access Control Markup Language (XACML), ttp://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml/

[16] D. Ferraiolo, R. Sandhu, S. Gavrila, D. Kuhn, and R. Chandramouli, "Proposed NIST Standard for Role-Based Access Control," ACM Transactions on Information and System Security (TISSEC), vol. 4, issue 3, Aug. 2001, pp. 224-274.