Accepted Manuscript

A Context-Aware System to Secure Enterprise Content: Incorporating Reliability Specifiers

Oyindamola Oluwatimi, Maria Damiani, Elisa Bertino

PII: S0167-4048(18)30301-8
DOI: 10.1016/j.cose.2018.04.001
Reference: COSE 1321

To appear in: *Computers & Security*

Received date: 17 November 2017
Revised date: 31 March 2018
Accepted date: 2 April 2018

Please cite this article as: Oyindamola Oluwatimi, Maria Damiani, Elisa Bertino, A Context-Aware System to Secure Enterprise Content: Incorporating Reliability Specifiers, *Computers & Security* (2018), doi: 10.1016/j.cose.2018.04.001

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

ACCEPTED MANUSCRIPT

A Context-Aware System to Secure Enterprise Content: Incorporating Reliability Specifiers

Oyindamola Oluwatimi
Department of Computer Science
Purdue University
West Lafayette, IN, USA
ooluwati@purdue.edu

Maria Damiani
Department of Computer Science
University of Milan
Milan, Italy
mdamiani@di.unimi.it

Elisa Bertino
Department of Computer Science
Purdue University
West Lafayette, IN, USA
bertino@purdue.edu

ABSTRACT

The sensors of a context-aware system extract contextual information from the environment and relay that information to higher-level processes of the system so to influence the system's control decisions. However, an adversary can maliciously influence such controls indirectly by manipulating the environment in which the sensors are monitoring, thereby granting privileges the adversary would otherwise not normally have. To address such context monitoring issues, we extend CASSEC by incorporating sentience-like constructs, which enable the emulation of "confidence", into our proximity-based access control model to grant the system the ability to make more inferable decisions based on the *degree of reliability* of extracted contextual information. In CASSEC 2.0, we evaluate our confidence constructs by implementing two new authentication mechanisms. Co-proximity authentication employs our time-based challenge-response protocol, which leverages Bluetooth Low Energy beacons as its underlying occupancy detection technology. Biometric authentication relies on the accelerometer and fingerprint sensors to measure behavioral and physiological user features to prevent unauthorized users from using an authorized user's device. We provide a feasibility study demonstrating how confidence constructs can improve the decision engine of context-aware access control systems.

Keywords

Access Control; Context Awareness; BYOD; Security; Mobility; Biometric; Authentication; Reliability; Proximity

1. INTRODUCTION

Context-aware access control systems aim to secure access to sensitive resources by adapting their access authorizations to the current context *without* explicit user intervention. In fact, enterprise organizations have adopted context-aware systems that leverage proximity-based access control (PrBAC) to mitigate threats of information leakage. That is, access control decisions are not solely based on the requesting user's location, but also on the location of other users in the physical space. In our previous paper [30], we introduced a secure, automated PrBAC architecture and prototype system that we referred to as the Context-Aware System to Secure Enterprise Content (CASSEC). CASSEC addressed two proximity-based scenarios often encountered in enterprise environments (c.f. Section 2): Separation of Duty (SoD) and Absence of Other Users (AOU).

To address such access control scenarios, CASSEC took a wireless, infrastructure-based approach to achieve the localization of occupants within a monitored space which enables geo-spatial RBAC [9, 22]. A wireless, infrastructure-based approach makes the system more resilient to malicious attacks; we assumed, for example, the least amount of trust in users since users may attempt to circumvent the access control process by not manually reporting their location or providing false location data. In addition, the architectural model allowed a fluid context-sensitive authorization process, thereby enabling zero interaction authorization (i.e., it did not require user intervention). While our system was agnostic with respect to the technological choices for detecting physical proximity, we had provided a simple implementation of the complete CASSEC architecture. We utilized Bluetooth and WiFi devices, which are widely used in enterprise environments, to address the occupancy detection problem [17], and therefore, no additional hardware was needed to deploy our system. We first showed how to enforce SoD by using Bluetooth MAC addresses of Client devices of nearby occupants as proof-of-location. That is, we extracted the MAC address from these devices to determine *who* was in a given space. We then showed how to enforce AOU by exploiting the degradation of WiFi received signal strength as a result of human-induced interference when people are near access points. That is, we utilized WiFi-capable devices to determine *how many* people were in a given space. With such information obtained passively by a Proximity Module (PM), the Authorization Server (AS) component was able to enforce PrBAC policies whenever an authenticated Client requested from the Enterprise Content Server (ECS) component access to resources depending on the presence, or lack thereof, of users. Our approach was the first to incorporate WiFi signal interference caused by occupants as part of a PrBAC system. Figure 1 displays CASSEC's architectural components.

The previous approach, however, has several drawbacks. First, it does not take into account the phenomena of radio signals permeating through walls. Multiple proximity modules residing in adjacent proximity zones would simultaneously detect the same Bluetooth-enabled Client, when in fact, the Client only existed in one of said proximity zones. As a result, such a benign occurrence is automatically inferred as malicious activity. Given that Bluetooth's omni-directional transmission range is 10m (~33 ft), the number of false attack detections may increase in standard enterprise settings, such as small offices or conference rooms.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

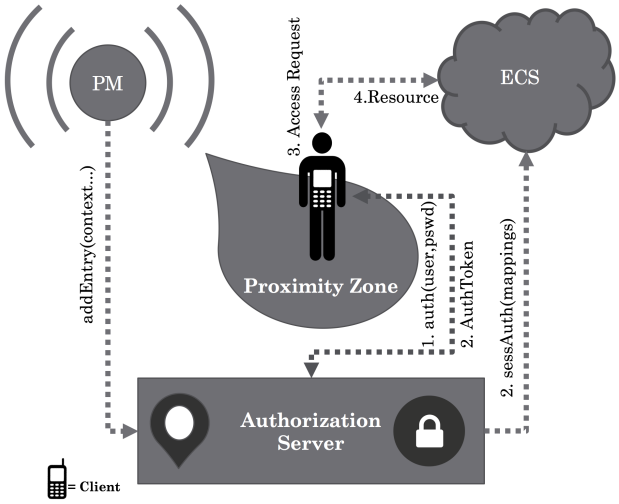


Figure 1: CASSEC's Proximity-based access control architecture. Arrows indicate wireless network communication.

A non-negligible false detection rate is a major drawback that hinders the practicality and ease-of-adoption of the solution. Second, the system was susceptible to observable Bluetooth manipulation (see Section 8), such as an unauthorized individual obtaining an authorized user's phone, whether by theft or voluntary provision. If such an attack occurs, the unauthorized individual can gain access to restricted resources s/he would normally otherwise not have access to.

To address such context monitoring issues, we further investigate techniques that leverage existing contextual information from both the physical and computing realms. Contextual information extracted from the environment can help a context-aware system in inferring the situation of entities within that environment. However, being able to infer the correct or more probable conclusion w.r.t. the situation of an entity highly depends on the reliability of the extracted contextual information. Reliability could be measured, for example, by the level of accuracy, precision, or security in using a technique or technology (e.g., occupancy detection or biometric authentication) to extract or process contextual information. With respect to security, a context-aware system may also need to adapt its access control decisions to the *degree of reliability* of such information. Given the dynamic nature of BYOD (Bring-Your-Own-Device) scenarios and idiosyncratic phenomenon observed in occupancy detection technologies (see Section 3), it is essential that context-aware systems emulate a sentient characteristic when making inferred decisions: confidence. Access control policies should incorporate confidence constructs when specifying contextual restrictions.

In this paper, we thus propose a major extension to CASSEC, which we refer to as CASSEC 2.0, by adding confidence constructs to the location and role constructs in PrBAC policies. In addition, we conduct a feasibility study to show that the approach is viable within an enterprise environment, which can be achieved via preexisting technologies and solutions integrated within the enterprise's mobile IT infrastructure. Through the location construct, a policy can specify that resource access authorization is granted only if the context-aware system can determine to a specified probability that a user is in a room. We employ Bluetooth Low Energy (BLE) capabilities of PMs and Clients to perform continuous co-proximity authentication, and use BLE beacons transmitted during this authentication phase to provide a certain degree of confidence that the Client is in a particular proximity zone, even when multiple PMs in adjacent rooms detect the same Client. Through the role construct, a policy can specify that access to resources is only granted if the system can determine with high confidence whether the current user of a Client device is the true owner of the device. We leverage accelerometer and fingerprint sensors within smartphones to achieve behavioral and physiological biometric authentication. Behavioral biometric authentication is achieved by passively analyzing the gait patterns of the Client's current user via the smartphone's accelerometer. Although human gait is behavioral and resistant to significant change over time, various factors can slightly influence the extracted gait features at runtime [26]. Consequently, if the Client cannot passively identify the current user through runtime gait measurements with high levels of assurance, the Client will take an active approach and request the user to authenticate him/herself via the fingerprint sensor (i.e., physiological biometric authentication) when the user next requests access to resources.

The CASSEC 2.0 system has thus the following contributions:

1. *Confidence Constructs:* We incorporate confidence specifiers into context-based access control policies. More specifically, we incorporate such specifiers into PrBAC's role and location constructs, thereby enabling the CASSEC 2.0 system to factor in the degree of reliability of contextual information during authentication and authorization processes.

2. *Feasibility Study:* We conduct a feasibility study to show that the approach of CASSEC 2.0 is viable in a practical enterprise setting. We leverage solutions within an enterprise's preexisting IT infrastructure to evaluate confidence constructs, and apply such constructs to biometric and co-proximity authentication.

3. *Co-Proximity Authentication:* We provide a timed challenge-response protocol using BLE beacons as our underlying co-proximity authentication technology. The protocol prevents an adversary, who has modified his device's unique user ID, from impersonating another user. However, our study shows that using distance-bounding techniques over BLE beacons is a feasible defense only against a sophisticated attacker able to execute relay attacks under a certain adversarial model.

4. *Biometric Authentication:* We leverage behavioral and physiological biometric authentication to evaluate confidence specifiers. Our study shows that our approach is feasible as we are able to verify that the current user of the Client device is the true owner with high confidence when the phone is placed on the hip and within the pocket, respectively.

The paper is organized as follows. Section 2 introduces proximity-based scenarios and specific examples that motivate this work. We then briefly discuss background information on occupation detection and biometric authentication

ACCEPTED MANUSCRIPT

techniques in Section 3. We provide in Section 4 a PrBAC policy specification for CASSEC 2.0. Section 5 establishes our system's assumption. Section 6 introduces the architecture and underlying components of our approach. Section Section 7 discusses implementation details followed by a report of data collected from our use case study. We analyze the security of our approach in Section 8. Next, we discuss relevant work in Section 9. Section 10 concludes the paper.

2. MOTIVATING SCENARIOS

Pervasive computing has enabled context-aware systems to be leveraged in a variety of settings, including mobile cloud services, hospitals, enterprises, and military organizations [15, 18, 33, 38]. In what follows we present scenarios from [30] motivating the need for context-aware systems in which access to sensitive resources must be controlled based on proximity parameters.

Consider a military organization with monitored government facilities such as restricted military bases or buildings. Military personnel are assigned roles that reflect ranking and privileges. The roles *General* and *Private* are assigned to the highest- and lowest-ranking personnel in the army, respectively. In terms of accessing restricted facilities or resources, the former is granted many privileges, while the latter has very few. Consider also the role *Civilian*, which indicates an individual operating outside of the military organization, and who is granted no privileges. Suppose that three military personnel, two *Generals* and one *Private*, are granted access to documents classified up to the level of *top secret* and *restricted*, respectively, according to a multi-level security model.

Separation of Duty Scenario. *A document classified as top secret is highly sensitive, and requires that* ***at least*** *two personnel with the role General be present in order for it to be accessed. The document is accessed via desktop terminal and is stored within a designated, but restricted office in which only Generals are allowed to enter.*
This scenario reflects the security principle SoD. That is, two or more people are responsible for cooperatively completing a task. In addition, the circumstances requires that said document must be accessed at a specific location.

Absence of Other Users Scenario. *A document classified as restricted, but with the additional caveat "****for your eyes only****", requires that a specific Private can access it via smartphone mobile, however, only if no other individuals are present at the time of access.*
Such an absence-based restriction not only includes military personnel of various rankings, but also individuals that assume the role of Civilian. Civilians are often temporarily recruited to work on military projects, but are highly monitored and usually given only the set of privileges needed to complete the project and nothing more. We note that, unlike the SoD scenario, in this AOU scenario the document can be accessed via the Private's smartphone device in any location including locations that Civilians may have access to. Therefore, less infrastructure is required as it is not necessary to know the identity of every person in the Private's vicinity.

3. BACKGROUND

3.1 Occupancy Detection

There is a variety of technologies that address the localization problem, that is, to determine and retrieve a user's location. Generally, each positioning system has at least two separate hardware components, a transmitter and a receiver to send and receive signals, respectively [40]. The receiver analyzes one of the following three characteristics of the received signal: angle-of-arrival (AoA), received-signal strength (RSS), and time of arrival (ToA). For example, the most widely used technology in context-aware applications is the Global Position System (GPS). It is a positioning tool which uses the propagation time of signals (i.e., ToA) from satellites to compute the position of a receiver anywhere on Earth. Other positioning techniques with different technologies include Infrared (IR), Radio Frequency (RF), Radio Frequency Identification (RFID) [25], magnetic field [27], ultrasound [33], Bluetooth [11], and WiFi [7, 10, 20, 36, 38].

Bluetooth Low Energy (BLE) can also be used to retrieve a user's relative location in an energy efficient manner, and it has been employed by beaconing services [43]. By utilizing widely-used BLE-based beacon protocols (e.g., Apple's iBeacon, Google's Eddystone, and AltBeacon [43]), a beacon region or the proximity of other BLE-enabled beacon devices (e.g., smartphones) can be detected. Detection is achieved by periodically broadcasting beacons that are picked up by BLE-enabled devices. We utilize Google's implementation as it is open source. Google's beacon provides two measurements. The distance measurement is an indicator of the proximity of one device to another which is determined based on the RSS value. The ranging measurement is an intuitive, user-friendly indicator of the distance between two devices which falls into one of the following ranges: *Immediate* (very close), *Near* (at a distance of 1-3m), *Far* (greater than 3m), or *Unknown* (the distance cannot be accurately determined). We also investigate other distance-bounding techniques. In particular, we investigate techniques that measure the time elapsed, i.e., the round trip time (RTT), during the exchange of packets between the transmitter and receiver. We implemented a distance-bounding system using BLE beacons as our underlying technology, which were programmed using Android's *android.bluetooth.le* APIs [4].

3.2 Biometric Authentication

Biometric information characterizes measurable human biological features [2]. Most biometric features are unique per person and can be found in every individual. In the context of security, *biometric authentication* refers to techniques that rely on such features to uniquely identify and validate the identity of an individual. Human biometrics can be classified into two types: physiological and behavioral. Physiological biometric authentication is based on static physical attributes such as fingerprints, iris, retina, or facial features, whereas behavioral biometric authentication relies on identifiable characteristics of a user's behavior that typically do not change over time such as keystroke dynamics, signature, or gait.

At a high level, biometric authentication has two phases: enrollment and authentication. Before authentication can occur, an individual must first be enrolled into the system by extracting and storing his/her biometric data within a template. Later in the authentication phase when the identity

ACCEPTED MANUSCRIPT

of the individual must be verified, the biometric data collected at runtime is compared to the previously constructed template. From this comparison, a similarity matching score is produced, and whether an individual is accepted/rejected (i.e., non-/identified) depends on a threshold set for the system. In this paper, we employ both physiological and behavioral biometric authentication for user verification using two techniques: fingerprint and gait recognition. Modern mobile devices already have integrated solutions to enroll and authenticate users via fingerprint scanning technology [4]. However, such devices lack gait recognition solutions. We therefore only describe user verification via gait recognition below.

3.2.1 User Verification via Gait Recognition

Lee and Grimson defined gait as "an idiosyncratic feature of a person that is determined by, among other things, an individual's weight, limb length, footwear, and posture combined with characteristic motion. Hence, gait can be used as a biometric measure to recognize known persons and classify unknown subjects" [26]. Empirical evidence supports this definition as researchers have conducted experiments which analyzed over 700 users' gait patterns and found gait patterns to be unique [29]. As a result, it is possible to verify whether the user of a mobile device is the true owner of that device.

Gait recognition for the purpose of user verification is not novel [2], nor is it the focus of this paper. The main approaches to measuring and analyzing gait biometric are machine vision, floor sensor, and wearable sensor. Deploying additional hardware incurs additional costs, as is the case in the first two approaches. Fortunately, state-of-the-art cellular devices are embedded with a set of sensors, including accelerometers, which have now become a standard for modern smartphones. Consequently, we only employ a wearable sensor approach. We leverage a recent work proposed by Ren *et al.* [34] for several reasons: (1) it utilizes readily available accelerometers embedded within smartphones to detect possible user spoofing in mobile healthcare systems; (2) it takes into account the fact that computational resources are limited on mobile devices; and (3) it is robust to variations in users' walking speed. See Section 6 for more details.

4. POLICY SPECIFICATION

Several research efforts have focused on the design of access control policy languages [3, 9, 16, 23, 28, 32]. In this section we introduce a simple, yet expressive policy specification (Table 1) that leverages existing policy languages. We adopt the syntactical structure of XACML, which is an XML-based language for access control, and apply it in defining proximity-based RBAC policies for CASSEC. The terms in quotes ' ' represent static tokens. The terms in italics indicate functions.

As it is standard in RBAC policies, a **role** is a job function that represents a set of privileges to perform actions on objects. An **object** is a data construct that is acted upon by a subject that has assumed a role. An **action** is an appropriate operation that can be applied to an object. We assume that users of our system may be mobile, and therefore, we incorporate usage controls regarding continuity of access [32]. An *obligation* specifies that certain constraints must be satisfied *prior* to or *while* accessing an object. A *topology* indicates a relation between the role and the **location** within

Table 1: PrBAC POLICY LANGUAGE

<Policies> ::= 'Begin' <policy-list> 'End'
<policy-list> ::= <policy> <policy-list> | <policy>
<policy> ::= <role-predicate> <object> <action> (<context>)
<role-predicate> ::= <role> (<confidence>) | <ranking>'('<role>')' (<confidence>)
<confidence> ::= <digit>
<digit> ::= ['0'-'9']
<ranking> ::= *equal* | *inferior* | *superior*
<action> ::= *read* | *write* | *delete* ...
<context> ::= <obligation> <location-constraint> | <obligation> <location-constraint> <proximity-constraints>
<obligation> ::= prior | while
<location-constraint> ::= <topology> <location> (<confidence>)
<topology> ::= in | out | adjacent ...
<proximity-constraints> ::= <proximity-constraint> <proximity-constraints> | <proximity-constraint>
<proximity-constraint> ::= <cardinality> <digit> <role-predicate><location-constraint>
<cardinality> ::= at_least | at_most

the spatial domain. Often in enterprise environments, access to restricted resources is contingent on not only the presence (or absence) of other people, but the relation towards the individual requesting access. A **role-predicate** specifies a specific role or relational function that takes the role of the requesting user and outputs a ranking relative to that role (i.e., superior(roleOfRequestingUser)). Last, an entity designated to enforce a policy may need prerequisites to be fulfilled, at least to a certain in extent. A **confidence** indicates the numerical threshold at which a requirement must be fulfilled, otherwise anything below that threshold is considered a policy violation. For example, specifying a role (General) with a confidence constraint (80%) semantically states that the system must be "80%" sure that the current user is the General. Figure 2 provides two examples of access control policies to specify the restrictions in SoD scenario and AOU scenario.

```
<Policies>
    <policy-list>
        <policy>
        <role> General</role>
        <object>TopSecretDocument</object>
        <action>Read</action>
        <context>
            <obligation>while</obligation>
            <location-constraint>
                <topology> in </topology>
                <location> GeneralsRoom</location>
                <confidence>100</confidence>
            </location-constraint>
            <proximity-constraints>
                <proximity-constraint>
                    <cardinality>at_least</cardinality>
                    <digit>2</digit>
                    <role> General </role>
                    <location> GeneralsRoom </location>
                </proximity-constraint>
            </proximity-constraints>
        </context>
        </policy>
    </policy-list>
```

```
<Policies>
    <policy-list>
        <policy>
        <role> Private</role>
        <object>RestrictedDocument</object>
        <action>Read</action>
        <context>
            <obligation>while</obligation>
            <location-constraint>
                <topology> in </topology>
                <location> Room105</location>
            </location-constraint>
            <proximity-constraints>
                <proximity-constraint>
                    <cardinality>at_most</cardinality>
                    <digit>0</digit>
                    <role>empty </role>
                    <location> room105</location>
                </proximity-constraint>
            </proximity-constraints>
        </policy>
        </context>
    </policy-list>
</Policies>
```

Figure 2: The policy on the left refers to the SoD scenario: at least two Generals must be present in order to access the TopSecretDocument. The policy on the right refers to the AOU scenario: the Private can access the ResctrictedDocument only if no one else is around.

5. THREATS AND ASSUMPTIONS

We make the following assumptions about the proposed system and the adversary. Each user, including the adversary, has full access to his/her device. Each device has been preauthorized by the IT admin for BYOD use. Preauthorization consists of verifying that (1) the device supports hardware-backed cryptographic key generation and storage and (2) the device's sensors, including Bluetooth, accelerometer, and fingerprint sensors, are functioning correctly. Consequently, we assume IT admins can be trusted. Each device must generate asymmetric cryptographic keys via Androids Hardware-Backed Keystore [4], in which the public key for that device is uploaded to a server for later use while the unexportable private key is stored securely in hardware. We trust the Android access control system, which includes the Android middleware and Linux Kernel, to correctly enforce all security policies. Physical security or video monitoring is employed to prevent the adversary from compromising proximity modules and entering the environment with foreign objects such as a non-secured phone. We only consider a passive adversary, and not active adversary. That is, the adversary has control of the communication channel, but is not able to inject new packets or compromise transmitted packets. The adversary is only able to relay packets transmitted between parties. In other words, the attacker posses standard Dolev-Yao capabilities [14]. We assume each proximity module has access to the public keys of Client devices, which can be retrieved on demand or during the installation of the proximity module.

6. SYSTEM DESIGN

In this section, we describe our CASSEC 2.0 platform that securely supports the SoD scenario and the AOU scenario described in Section 2. We adhere to design goals from the previous work, which include providing a secure, automated, and generalized architecture with responsibilities of each system component clearly defined. In CASSEC 2.0's architecture, we assume the least amount of trusted parties as possible. Our context-aware system proactively monitors and collects information about the environment in lieu of manual intervention by entities within that environment. Specifically, we do not rely on users, possibly malicious, to manually report their location. Therefore, we choose an infrastructure-based approach that uses wireless hardware to localize occupants within a monitored space. In the rest of the section, we define our interpretation of the term *proximity* and then provide an overview of the architectural components of CASSEC and how they relate to our access control framework.

6.1 Proximity Zone

We rely on geographical proximity, which indicates that two entities are located within a certain distance in the physical space [18]. That is, in our work, *proximity* of a user is defined by a region of space monitored by a proximity module. The user must be within the region of space in order to gain access. We refer to this region of monitored space as a *proximity zone*. The level of precision in determining the location of a user and the proximity of other users is application dependent [10, 17].

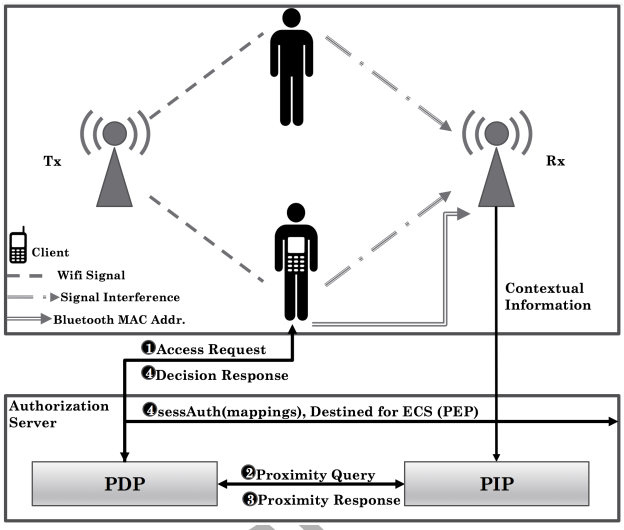


Figure 3: CASSEC's access control framework communicating with our prototype proximity module.

6.2 Components

6.2.1 Access Control

Here we describe the architectural components tasked with enforcing our PrBAC policies.

Enterprise Content Server (ECS): The ECS, which acts as the Policy Enforcement Point (PEP), delivers enterprise resources to users who request access. By designing this component as a server, a heterogeneous network of end-users' devices can be serviced. Therefore, access to resources can be requested from desktop terminals or mobile devices.

Authorization Server (AS): The AS hosts the access control decision-making engine of the authorization framework. After a user has been authenticated by the AS via login credentials, it returns an authentication token to the Client device. The token, which is submitted to the ECS by the Client, is used to associate an authenticated user with authorized roles. The AS itself is composed of two sub-components: Policy Decision Point (PDP) and Policy Information Point (PIP). We discuss in more detail the construction of the authentication token and AS's sub-components later in Section 6.3.

6.2.2 Contextual Information

In order to extract contextual information from the environment, we take both an active and passive approach. We use the terms *active* and *passive* to indicate whether or not users are required to physically interact with the entity collecting contextual information. The components involved in contextual information acquisition are as follows:

Client: A Client is a device used to request access to a resource by a user. If the request is granted, a user can view the data on the device (e.g., desktop terminal or mobile smartphone). Unlike their desktop counterparts, smartphone devices allow mobility with respect to embedded sensors and network connectivity. Consequently, in our prototype system, we take an active approach to user verification via biometric authentication by utilizing a smartphone as the Client device. That is, the Client is also designated to verify that the current user of the device is the true owner

ACCEPTED MANUSCRIPT

of the device. We note that solutions have been developed that take a passive approach to the collection of biometric features, which may be more secure. If we were to take a passive approach to biometric authentication, the example policies in Figure 2 would also include confidence thresholds under the *role* specifier since the AS is designated to evaluate if policies are adhered to. We discuss this further in Section 8.

Proximity Module (PM): The role of the PM is to collect and analyze contextual information in order to detect the proximity of users. This detection process occurs periodically, and proximity-related information is sent to the AS. Although a PM is the set of physical devices that determine proximity, we consider them as independent of the PIP as the PIP is the entity that directly communicates with the PDP. Users do not physically interact with the PM in our prototype system, and therefore it is considered passive.

Our architectural components are shown in Figure 1. We do not discuss cryptographic schemes to protect network communication between the entities in our system model. We assume that an underlying secure network infrastructure is in place, as usual in enterprise environments. Although the figure only shows one PM and consequently only one proximity zone, in practice an enterprise building will have multiple PMs, possibly one for each room.

6.3 Access Control Framework

The **PDP** is the specific entity that is delegated to make access decisions. It maintains a database of PrBAC policies. Given these policies, the PDP first verifies if someone is a user of the system. The PDP then retrieves the latest information regarding the user's location and the presence of other users from the PIP. Such information allows the PDP to determine the set of authorized geo-spatial roles if proximity constraints are satisfied. Next, the PDP constructs and returns to the Client an authentication token. The token, at minimum, contains a generated temporary ID. It may also contain an expiration date. As such, the token is utilized as a session identifier. Last, the PDP maintains a database mapping of session IDs to the set of active authorized geo-spatial roles for each user. This mapping is *also* sent to the PEP each time a role is authorized.

The **PEP**'s role, implemented as part of the ECS, is to enforce proximity restrictions for enterprise content. During a request, a Client submits an authentication token to the ECS. The PEP extracts the temporary session ID from the token. The PDP continually updates the PEP of mappings of session IDs to a set of active authorized geo-spatial roles. First, the mapping makes it possible to enforce access restrictions according to the roles associated with that ID. Second, it also enables it to service multiple Client devices simultaneously. Third, this design anonymizes users as the PEP does not have any information that identifies users such as locations and credentials.

The **PIP**'s role is to store and maintain contextual information about an enterprise's proximity zones. Each PM, after co-proximity authentication of Clients, is required to transmit four pieces of information to the PIP: a proximity zone identifier, the number of people detected, a list of captured UIDs[1] and corresponding RSS values of BLE beacons, and a timestamp. The PIP then records the collected data into its context database. Instead of the PIP polling the PM for information, we minimize communication by requiring that the PM updates the AS only when characteristics of the proximity zone changes. In addition, this clear designation of duties also minimizes overhead in both the PM and AS. Considering the dynamic nature of the environment, the PIP must update the PDP as frequently as the occurrences of updates to the context database. Such updates allow the PDP to continuously check for any instance of proximity-based violations by users. At the time of violation, the PDP invalidates the relevant session ID mappings by associating existing session IDs with newly recomputed *authorized* geo-spatial roles, if any, according to PrBAC policies. The PDP then remotely informs the PEP of invalid mappings while providing new authorized ones. The PDP can also alert the enterprise's administrators to take appropriate action. Such a design makes the system completely automated by only requiring users to be authenticated once by the AS.

[1]We assume that each user of the system has an identifier unique to that user.

6.4 Co-Proximity Authentication

Radio signals permeate through walls, and therefore it is possible that two PMs located in two adjacent rooms may detect the same Client device, even though in reality the Client is located in one of the rooms. However, such signals exhibit attenuation as they pass through walls. We leverage this phenomena to determine the likelihood that a Client is in a given room. In particular, we analyze the RSS values from BLE beacons to initiate the co-proximity authentication process, which determines that a *legitimate* Client is within a specific proximity zone.

Overview. The protocol to authenticate the user's co-proximity to a PM consists of two phases: the initialization phase and the location authentication phase. First, the initialization phase establishes a temporary session key (SK) securely shared and only accessible between a PM and a Client. Next, the SK is later used in the location authentication phase, in which a timed challenge-response protocol is executed. The crux of authenticating the user's co-proximity is analyzing the content of the beacon as well as the measured round trip time. We explain both phases in detail below.

Initialization Phase. The initialization phase is activated once the user enters Δ_2, that is, the concentric region as indicated by BLE's *Near* ranging measurement (i.e., between 1-3m from the PM as displayed in Figure 4). Placing a PM at the center of an average sized conference room (e.g., ~6m x 6m) allows the PM to detect and monitor the movements of any Client device that enters the room. In addition, positioning in such a way may minimize the overlapping of concentric regions of two adjacent PMs' proximity zones. Once the Client enters Δ_2, the PM generates a temporary SK and encrypts it with the Client's public key. As stated in Section 5, the public key can be retrieved from the authorization server on demand or during the installation of the PM. The SK is a one-time pad which consists of a string of bits generated using a cryptographically secure pseudo-random number generator. The encrypted SK (Step 1 in Figure 5) is then sent to the Client via the AS, which is then decrypted at the Client using the Client's hardware-bound private key. The Client finalizes the initialization phase by responding with an acknowledgement of message receipt, which is relayed back to the PM. We note that there are a number of methods to securely exchange temporary session keys. For

ACCEPTED MANUSCRIPT

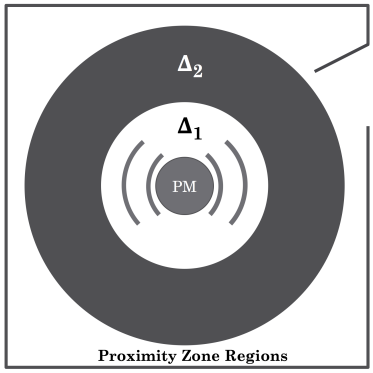


Figure 4: A Proximity Module's proximity regions in a conference room. The regions are virtually constructed using the ranging measurements of Bluetooth Low Energy: Δ_1 and Δ_2 is any position less than 1m and between 1-3m, respectively.

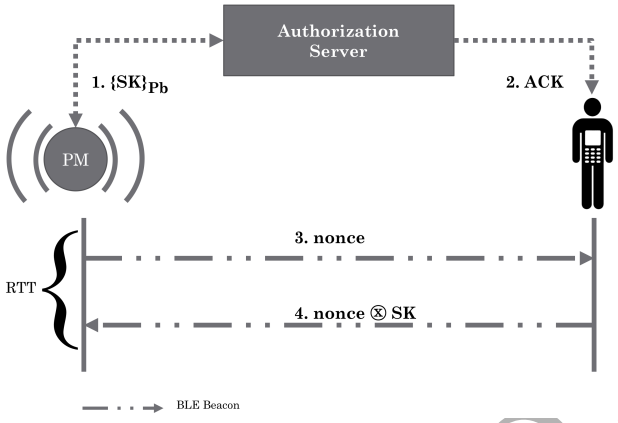


Figure 5: CASSEC's co-proximity authentication protocol.

example, the PM could purely rely on BLE beacons to transmit the encrypted SK, thereby minimizing communication with the server. However, we did not choose this mode of transmission because of limited data capacity in BLE beacon's advertising data structures [43].

Authentication Phase. The PM will continue to monitor and track the Client's movements. The authentication phase is activated once the user enters Δ_1, that is, the concentric region as indicated by BLE's *Immediate* ranging measurement (i.e., less than 1m from the PM). At this point, the PM initiates a timed challenge-response protocol with the target Client. The PM generates a fresh nonce (string of random bits), embeds the nonce into a BLE beacon, and transmits the beacon. Upon successful transmission, the PM records the time of transmission and precomputes the expected response. Upon reception, the Client calculates an XOR value, using the nonce and the SK as the two inputs. XOR operations are simple and require minimal CPU cycles to compute as opposed to other widely-used cryptographic schemes with non-negligible encryption/decryption times [24]. Leveraging XOR operations thus allows the Client to minimize the time to calculate a response to the challenge, and subsequently package and transmit the response within a BLE beacon. Upon reception of the Client's response beacon, the PM calculates the RTT value and verifies that the precomputed value matches the received value. If the values match and the RTT is less than or equal to a specified threshold (RTT_{TH}), the PM informs the AS that the specific Client's location has been authenticated with 100% confidence, otherwise the PM and Client must repeat both the initialization and authentication phases. We discuss how we determined RTT_{TH} in Section 8. Both phases must be repeated since information about the temporary session key that is generated in the initialization phase is leaked in the authentication phase. An attacker can simply perform an XOR of the nonce, which was transmitted in cleartext, and the Client's response beacon to calculate the session key.

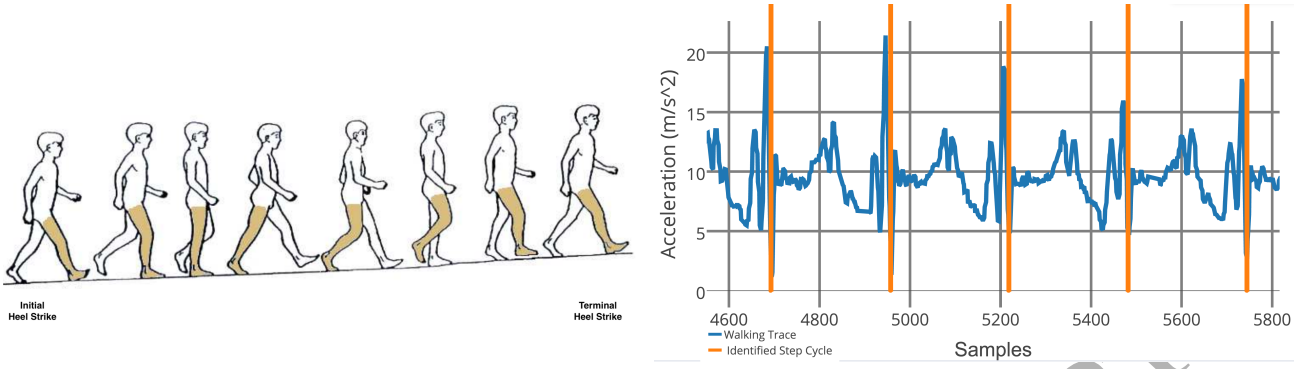
To address circumstances resulting in proximity zones partially overlapping, we take a binary approach. In the case that multiple PMs detect and authenticate a Client via BLE beacons simultaneously given that BLE beacons can travel several meters, for simplicity, we classify a Client to be in one of the corresponding rooms with 100% confidence only if information sent by a PM meet two conditions: (1) the RSS value (measured from the beacon) is the strongest of all RSS values detected by other PMs; (2) the number of people detected and captured UIDs match. Otherwise, there is 0% confidence in the Client's location. The left policy in Figure 2 provides an example of PrBAC policy that specifies that the entity enforcing the policy must determine that the General is in fact located in the *GeneralsRoom* with 100% confidence to grant access to the *TopSecretDocument*.

6.5 Biometric Authentication

User verification via biometric authentication is isolated to the only active component in our prototype system, that is, the Client. We specifically develop an Android application that leverages the smartphone's capabilities to scan fingerprints and measure acceleration in order to achieve physiological and behavioral biometric authentication, respectively. User verification is abstractly a two phase process (see Section 3): the enrollment and authentication phases. With respect to security, it is vital that enterprise administrators proctor the enrollment phase in-person to confirm that biometric measurements taken by a Client device match the true owner of the device. Fingerprint scanning and the collection of walking traces are achieved and easily integrated into our application using Android's Fingerprint Authentication and Sensor Manager APIs[2]. To ensure the privacy of users, the fingerprint and gait templates constructed during the enrollment phase never leave the device.

We implemented the behavioral component of the user verification framework in a similar fashion as proposed by Ren *et al.* [34]. The framework consists of three components, which can be abstracted to the enrollment and authentication phases previously mentioned: Step Cycle Identification, Step Cycle Interpolation, and Similarity Comparison. The components are built on the fact that human gait should be cyclic in nature, and hence should exhibit high correlation. Here, a step cycle is the period defined by the two consecutive heel strikes on the same leg (see Figure 6(a)). The Step Cycle Identification component identifies step cycles in a walking trace, and then uses the extracted features to

[2]We do not elaborate on implementation details as Android provides detailed instructions and samples to utilize Android Fingerprint Authentication and acceleration measuring [4].



(a) (b)

Figure 6: 6(a) is an illustration of a complete gait cycle from the initial heel strike to the terminal heel strike (from [39]). 6(b) displays preliminary measurements of accelerometer signals of a walking trace in the vertical direction we collected using a Nexus 6P smartphone. Orange lines indicate step cycles identified by heel strike impacts.

construct and store a biometric template. Although smartphone accelerometers provide signals in three dimensions, the framework extracts only the signals from the vertical direction to identify impacts caused by heel strikes. Figure 6(b) displays a walking trace with identified cyclical heel strike impacts. Users usually walk at varying speeds, which would negatively impact the verification process if the template and the runtime measurements are of traces with different speeds. Addressing this potential problem, the Step Cycle Interpolation phase enables robust user verification by normalizing identified step cycles of different lengths into fixed lengths. Figure 7 displays the interpolated accelerometer signals, recorded using a Nexus 6P, of slow (slower than 0.7 m/s), normal (about 0.7 - 1.1 m/s), and fast (about 1.1 - 1.4 m/s) walking traces to a fixed length of 400 samples. The figure demonstrates that step cycles are highly correlated regardless of walking speed. Last, user authentication is performed in the Similarity Comparison phase, which utilizes a weighted Pearson correlation coefficient (PCC) based method.

We apply defense-in-depth within the authentication phase. We first use Pearson correlation coefficients when computing the similarity between the gait template and the walking trace runtime measurements. Users are only verified if similarity scores are above a predefined threshold (see Section 7.4). If similarity scores fall below the threshold, the user is then required to perform authentication via fingerprint scanner when the user attempts to access the phone. We are unable to set a threshold for fingerprint authentication as we rely on the Client device's integrated fingerprint solution. If the user neither can be verified via behavioral nor physiological biometric authentication, the Client ensures that sensitive enterprise content is inaccessible by locking the device[3]. In addition, the Client can alert enterprise administrators for possible user spoofing.

[3]We build an application on the Client using Android's Device Administration API, which includes the device lock ability [4].

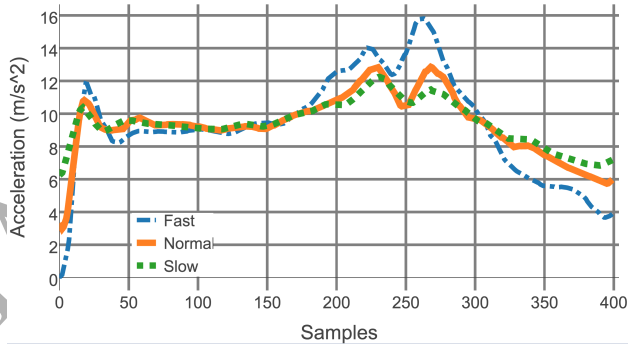


Figure 7: Step cycle interpolation applied to walking traces collected using our Nexus 6P smartphone at three different speeds: slow, normal, and fast.

7. PROTOTYPE IMPLEMENTATION

7.1 The ECS

The ECS was implemented in PHP and hosted on a remote commercial server. The resources that it could serve to Clients were simple text files. We implemented user interfaces (UI) in order for Clients to request access to specific files. The ECS provides a function that can be remotely invoked via URL: *sessAuth(mappings)*. The function is invoked by the AS to update the ECS regarding the active geo-spatial roles for Clients in the event that location updates reflect proximity violations.

7.2 The AS

The AS was also implemented in PHP and hosted on the same server as the ECS. We implemented the UI in order for Clients to pass in authentication credentials via a login page. The AS provides two functions that can be remotely invoked via URLs: *auth(user,psswd)* and *addEntry(pzoneID,numOfPpl, UIDs+RSSs, time)*. The first function is invoked by a Client via the UI and the second is invoked by the PM to update proximity information within

ACCEPTED MANUSCRIPT

the context database.

7.3 The PM

As in any basic positioning system, a PM incorporates a transmitter and a receiver. We define a transmitter as a wireless-enabled device that is a source of contextual information regarding the occupants within a proximity zone. A receiver is a wireless-enabled device that acts as a sink for such contextual information.

We utilize BLE-enabled smartphones and WiFi access points (APs) as transmitters. In regards to smartphones, we embed three values into BLE beacons to support co-proximity authentication. Generally, these devices periodically broadcast their 48-bit Bluetooth MAC addresses with a less than 10 meter range indoors when Bluetooth is enabled. However, since Android 6.0, the MAC address found in a BLE beacon is replaced with a random value at various intervals to protect user privacy [4]. User privacy is not a concern within the enterprise scenarios that CASSEC targets. Disabling this feature would require modifying the Android OS, which reduces the deployability of our solution. Therefore, we cannot rely on this hardware address to identify users. Instead, in CASSEC 2.0, we embed a 48-bit UID into BLE's local name data structure using Android's *BluetoothAdapter.getDefaultAdapter().setName(UID)*. The BLE beacon data protocol is limited with respect to the amount of custom data we are able to embed within a beacon. As a result, the nonce, as well as the one-time pad SK generated by the PM, is restricted to 12-bytes. With the remaining space, we embed a 16-byte service UUID which enables Clients and Proximity Modules to communicate under a beacon service. We require that users of the system permanently enable their smartphones' Bluetooth. Such a requirement can be easily enforced by Enterprise Mobility Management services [31]. WiFi APs transmit data over signals that can be measured. However, such signals are significantly influenced by the environment. We rely on the interference of signals as a result of human activity to determine the number of occupants in a proximity zone.

The PM was implemented as two physical devices: a Pixel C tablet running Android Oreo (API 26 v8.0) and a laptop using Python running Linux. For brevity, we refer to these devices as simply the PM. The PM was charged with periodically scanning signals produced by BLE and WiFi devices. Beacon scan settings were set to *SCAN_MODE_LOW_LATENCY* from the *ScanSettings* API, while WiFi signals were scanned every 10 seconds. The PM extracts the UIDs of beacons from nearby occupants' smartphones. The UIDs are used as proof-of-location once co-proximity authentication has been established, which determines *who* is in a given space. The PM also measures the received signal strength from a designated WiFi AP. The receiver processes the measured WiFi RSS value and determines *how many* occupants are in a given space. Last, the receiver publishes the UIDs, beacon RSS values, and the number of occupants to the authorization server only when previously collected contextual information changes.

We note that the various components of the CASSEC's system architecture can be integrated into the same physical component when implemented. For example, a smartphone mobile device can act both as *Client* and transmitter because the same device used to request access to a resource is the same device that periodically broadcasts its Bluetooth data structures. Similarly, a desktop terminal can act both as *Client* and receiver because it can also be used to scan and process WiFi and Bluetooth contextual information.

7.4 Use Case

In this section, we evaluate features of the CASSEC 2.0 prototype system in order to provide clear insights into addressing the issues raised in Section 2. We measure the performance of the system's biometric and co-proximity authentication components to prove the feasibility of securing enterprise content under a proximity-based access control model.

7.4.1 Deployment

We deployed our hardware and tested our prototype system in a two bedroom apartment whose layout is shown in Figure 8. We now briefly describe the hardware utilized in our platform.

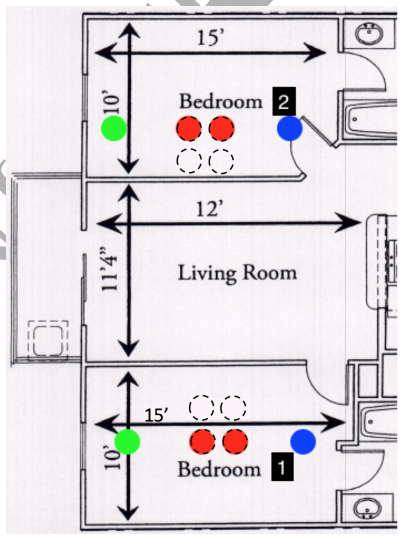


Figure 8: The blueprint of a two-bedroom apartment in which the prototype system had been deployed. The blue markers and green markers indicate the positions of WiFi access points and laptops, respectively. The dotted lines indicate the two possible positions for each human, and transitions simply require moving two steps without changing body orientation. The red dots represent the current positions of the humans standing still while facing the laptop.

The Wireless-N (802.11n) WiFi AP transmitter was a Motorola SURFBoard SBG6580, indicated in blue, that supports two frequency bands which are 2.5GHz and 5.0GHz. We chose the higher-frequency band to take advantage of additional channels that are less prone to interference than 2.4GHz. The receiver was a Dell Latitude E6430, indicated in green, equipped with a BCM4313 802.11bgn wireless network adapter and a Dell Wireless 380 Bluetooth 4.0. The transmitter and the receiver were placed 3 meters apart and were elevated 1 meter above the floor. The Bluetooth-enabled transmitters used in our study were Samsung S3 GT-i9300 and Nexus 6P. The Nexus 6P, which has a fingerprint scanner and an accelerometer that supports a 200Hz

ACCEPTED MANUSCRIPT

sampling rate, was used for biometric collection and analysis.

7.4.2 Use Case Evaluations

Evaluation 1: Selecting Frequency Channel. Given a wireless link between a transmitter and a receiver, an individual crossing the line of sight between the two communicating wireless sensors affects the RSS measured by the receiver. However, the change in RSS depends on the frequency channel [10]. Our goal is to determine which channel is the best for detecting human activity based on our particular WiFi-enabled devices. We test 2 non-overlapping 40MHz channels: Channel A (5180MHz) and Channel B (5220MHz). The experimental setup is as follows. Throughout the complete test, we continuously measure the RSS value sampling twice per second. Every 30 seconds we change the number of individuals obstructing the LOS by 1 starting from zero to two, and then in a decreasing fashion. The occupants were situated equidistant from each receiver. A Python script was written to automatically begin the test. The tests were conducted in Bedroom 1.

The results in Figure 9 demonstrate that there is no significant difference in measurement variation in human-induced interference in RSS signals between Channel A and Channel B. At first, Channel A appears to be more consistent as the level of signal interference in samples 60 - 120 aligns with values in samples 180 - 240 when the number of individuals increases from zero to one and two to one, respectively. This is not observed in Channel B during that period. However, the values for Channel A appear to indicate the presence of a number of individuals different from the number of individuals actually present from samples 330 onward. This fluctuation is not observed in Channel B. Although Figure 9 shows the results of only one complete test, we performed this test 3 times and observed similar changes in values. Given these observations, we select Channel B as a means for testing in the rest of the study.

We also make some general observations about human-induced RSS changes. We observed distinct variances in signal strength almost every 30 seconds (multiple of 60 units in Figure 9). First, by initiating the test with no individuals obstructing the LOS, we were able to establish a baseline for the signal strength between the transmitter and receiver. The RSS value remained always constant within that time period up until to two seconds after the 30 second mark. That is, using our existing hardware, we were able to determine that once we increase the number of individuals by one, the individuals must remain in the LOS for *at least* one second for the receiver to observe some interference from human activity. Such phenomena was also observed at the beginning or end of each period. Second, regardless of the selected channel, when the LOS is obstructed by an individual the RSS on average decreases. In addition, distinct dBm drop ranges exist depending on the number of individuals. Therefore we can infer the presence or absence of humans based on RSS' ranges. For example, in Channel A, we consistently observed a drop range of 6-8 dBm between 30-60, 90-120, 150-180, and 210-240 seconds. We note that our observations are likely to change using different WiFi-enabled hardware.

Evaluation 2: WiFi Detection Accuracy. The goal here is to test the WiFi localization component of our PM. Specifically, we implemented a simple algorithm to detect the number of people within the LOS based on our observations of human-induced RSS changes from Evaluation 1. The setup to this test is similar to the setup for Evaluation 1, except that we perform the test in *both* Bedrooms 1 and 2. We conduct the test on Channel B.

Location	Detection
Bedroom1	89%
Bedroom2	43%

Table 2: We leverage the human-induced signal interference in WiFi received signal strength (RSS) to detect occupancy within a monitored room. 89% and 48% accuracy was achieved for Bedroom 1 and 2, respectively. We believe that detection accuracy in the latter case were low because our technique was based on data acquired in Evaluation 1 which analyzed signal interference from Bedroom 1. Specifically, we believe that there were *other* unseen environmental factors that influenced the RSS values of human activity in Bedroom 2.

Table 2 displays the results. The system was able to detect with strong accuracy (89%) the number of occupants obstructing the line of sight in Bedroom 1. At certain points, sporadic fluctuations occurred that caused the system to return an incorrect number. On the other hand, the system was only able to detect occupancy with 44% accuracy in Bedroom 2. After further analysis (by performing Evaluation 1 in Bedroom 2), we observed the human-induced interference was slightly different in RSS levels. Although the physical layouts of Bedroom 1 and 2 are identical, there may be other (unseen) environmental factors that also influenced the RSS levels to slightly differ between the two rooms. For example, such factors may include overlapping wireless networks (possibly using the same channel) from neighboring apartments, appliances and electronics emitting radio frequency interference, and simply walls and floors blocking wireless signals in different ways depending on the location of access points [6]. We leave further analysis of WiFi signal interference caused by various environmental factors for future work.

Evaluation 3: Gait Recognition Detection Latency. The goal here is to determine the required length of a walking trace to identify the true owner of a Client device using gait recognition. CASSEC was developed with certain enterprises in mind that desire high assurances that sensitive enterprise content on end-users' devices is well protected. We therefore set the pre-defined threshold to 0.8. A single user participated in this study using a Nexus 6P smartphone device to record and analyze accelerometer values. In order to execute the test, the user performed the enrollment and authentication phases. We first collected from the user six 60-second walking traces at normal speed: the first and subsequent five traces to be used for gait template construction and runtime measurements in the enrollment and authentication phases, respectively. Then, in the enrollment phase, $n, n = 5, 6, ...60$, gait templates were constructed for the user which were derived from extracting n seconds from the first trace. Next, in the authentication phase, we extracted n seconds from each of the subsequent traces to analyze and compare biometric templates with run-time measurements that are of corresponding lengths. In its entirety, we repeated this test twice; differentiating the two by placement

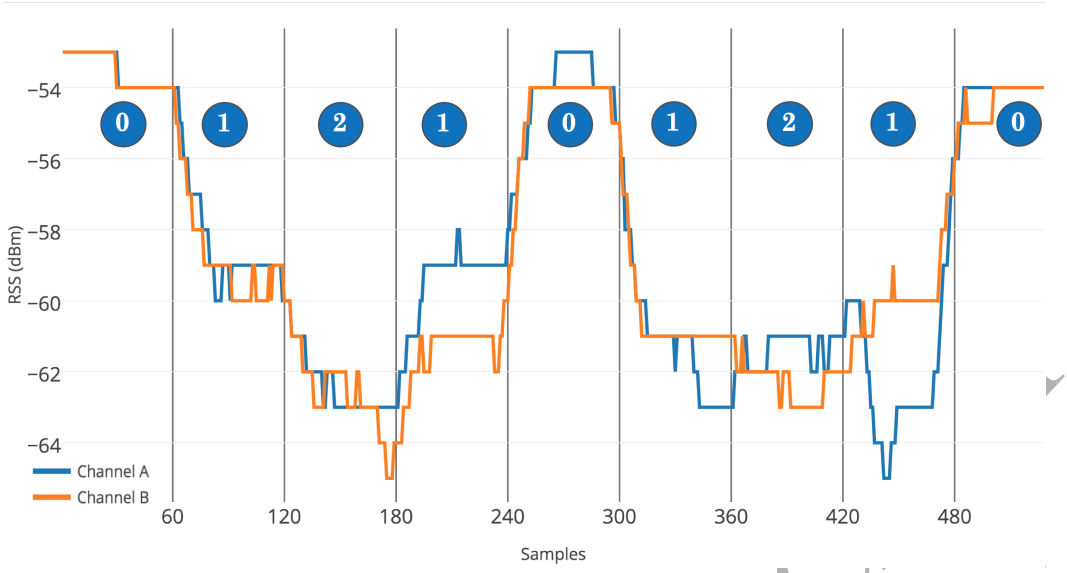


Figure 9: RSS measurements of wireless links on different frequency bands when human bodies obstruct the line of sight (LOS). The blue circles indicate the number of humans in the LOS within each 60-sample period (i.e., every 30 seconds).

of the smartphone device: on the hip and within the pocket.

Figure 10 displays the results of the test. We first note that all similarity scores produced were at least greater or equal to the predefined threshold. On average, the system was able to detect that the current user of the Client device is the true owner with approximately 91% and 89% confidence (i.e., similarity score) when the phone is placed on the hip and within the pocket, respectively. The system only required five seconds of each walking trace to make such an assertion. We also observe that longer traces eventually produce higher levels of confidence in identifying the true owner because more gait features were extracted, and therefore more identifying features can be determined during the authentication phase.

Extracting more gait features over a longer period of time produces higher levels of confidence while in the pocket as compared to on the hip. Upon further analysis of individual traces from both the hip and pocket, it appears that hip traces have increased oscillations that are not quite periodic. Particularly, we observed that there are more variations in-between the heel strikes as compared to pocket traces. First, the Step Cycle Identification component may falsely identify when the user's leg comes in contact with the ground if oscillations closely resembles heel strikes. Second, the gait recognition program assumes a cyclic nature, and thus if no repetition occurs within these sporadic oscillations, correct heel strikes, which occur outside of the oscillations, may not be properly analyzed as well. It is evident that the hip is continuously gyrating, and therefore, has a periodic motion. However, we believe that the increased (and erroneous) variations are the result of the method in which we attached the device to the hip. While the device is securely fastened and flushed with the hip clip in order to minimize erroneous movement of the device, it is difficult to replicate such a secure grip with the hip clip itself as it is attached to the wearer's clothing. However, while placed in the pocket, the device is resistant to minor shuffling because it is pressed against the user's clothing and leg. Nevertheless, the results of this test demonstrate that feasibility to detect the true owner of the Client device with high confidence when placed within the pocket or attached to the hip, even considering the inherent erroneous data that is acquired while the device is attached to the hip.

As stated in Section 1 and Section 3, the development of gait recognition techniques for user verification is outside the scope of this work. We emphasize that this work is a feasibility study that demonstrates the application of biometric techniques such as gait authentication to securing enterprise content under a proximity-based access control model solely using one mobile device. We refer readers to the work by Ren *et al.* [34] for an extensive user evaluation of the gait authentication technique we have leveraged.

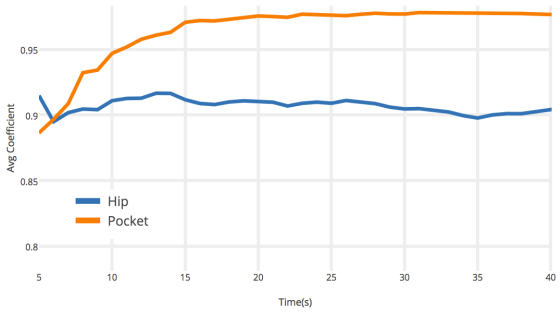


Figure 10: Average similarity score by varying the duration of user profile trace and runtime measurement trace.

Evaluation 4: Robustness Against Different Walking Speeds. The goal here is to test the robustness of the system against various walking speeds. We applied the same

ACCEPTED MANUSCRIPT

methodology as Evaluation 3 with an exception. We also compare the biometric template constructed from the normal walking trace to five runtime measurements collected from the user for both slow and fast walking speeds.

Figure 11 displays the results of the test. We achieved a 100% detection rate with a threshold of 0.8 when similarity score calculations were derived from the normal and fast walking traces. However, in a few instances, our system was unable to authenticate the current user as the true owner of the device. 07% of the similarity scores calculated, which we consider negligible, fell within the range of [0.7,0.8). Nevertheless, we can observe in any walking trace, including the slow walking trace, that there is a positive correlation between the trace length and the similarity scores produced. That is, the system can reliably determine the user with increasing confidence over a longer period of time.

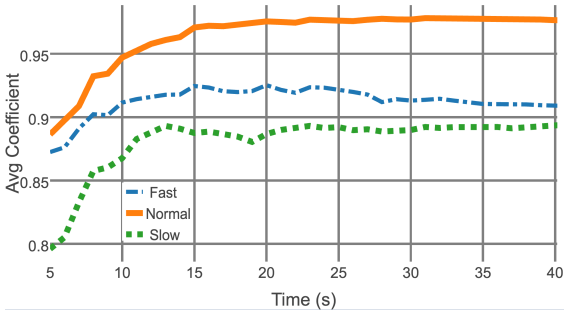


Figure 11: Average similarity score calculated by comparing the normal walking biometric template with both the slow and fast runtime measurement walking traces.

Evaluation 5: Capturing BLE Beacon RTT Values. One type of distance-bounding technique uses the elapsed time between two devices for distance estimation. Our goal for this test is to apply such a technique to BLE and determine if indeed that the round trip time of beacons is a function of distance. We exchanged beacons between two BLE-capable devices (Pixel C tablet and Nexus 6P smartphone) and recorded 100 RTT values at various distances. The devices were laid down across a wooden desk with the front screen facing upwards. Figure 12 shows the distribution of RTT values measured between the two devices. We note that displayed values reflect distance estimation as implemented in our co-proximity authentication. We first observed that most of the RTT values, at each distance, are centered around the median (the black line within the inner quartile range). For example, at distances of 1ft, 4ft, and 6ft, the RTT values are centered around approximately 81ms (± 1ms), while at a distance of 2ft, RTT values are centered around 77ms. We also observed that the IQR, the box that spans the first and third quartiles, are centered in between 72ms and 86ms. Consequently, no significant statistical variations of RTT values exist when the PM and the Client device executed the timed challenge-response protocol at distances between 1-6ft. Moreover, we produced similar results when we applied the same experimental process, but instead separated the devices with a 1ft wall. We discuss the security implications in Section 8.

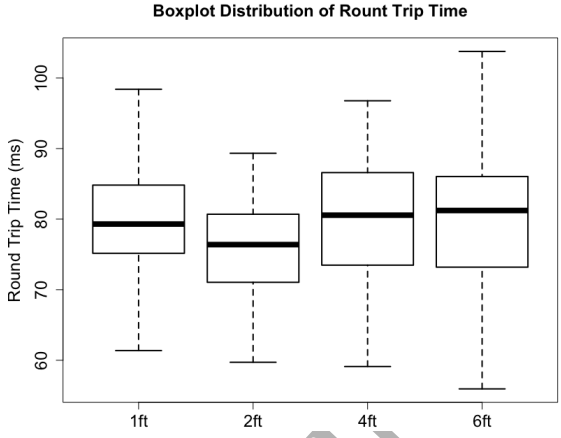


Figure 12: A boxplot diagram showing the distribution of round trip time of 100 Bluetooth Low Energy beacons each at various distances, exchanged between a Proximity Module and a Client.

8. SECURITY ANALYSIS

In this section, we present a security analysis of our CASSEC platform to analyze attacks aiming at circumventing its PrBAC restrictions. Below, we provide various attack vectors that could be used and, subsequently, a means to mitigate the threat or minimize the attack vector surface.

8.1 Bluetooth Manipulation

In our previous work, when a PM publishes a MAC address to the PIP, it attests that a specific individual is at a specific proximity zone. A malicious user may attempt to root his/her device and modify the MAC address in order to impersonate another user of the system. In this paper, however, such malicious modification of MAC addresses would do no harm for two reasons: (1) we rely on UIDs that are dynamically embedded into data structures within BLE beacons; and (2) Android Oreo (API 26 v8.0) automatically randomizes the MAC addresses of beacons. Moreover, an attacker that roots his device to dynamically alter beacon UIDs (through a modified and unauthorized custom OS) to impersonate a legitimate user would fail the challenge-response protocol for several reasons including the attacker's inability to access the legitimate user's private key, which is bound to the user's Client device hardware (i.e., not exportable). In addition, Samsung has demonstrated via Samsung KNOX 2.0, a custom Android OS intended for enterprise environments [31], hardware and software security features that leave the device inoperable once it detects a root attack, which is a sufficient mechanism to defend against malicious modification of the OS.

One attack that malicious users may attempt is masking their smartphones' Bluetooth peripheral services by either disabling the Bluetooth or simply leaving the device in another room. Although we require that Bluetooth be permanently enabled on users' devices, we do not incorporate an enforcement mechanism within the phone to meet such requirement. However, our system is able to detect if the violation of such requirement occurs. The WiFi localization technique is able to determine the number of occupants in the room. If the number of occupants and the number of

ACCEPTED MANUSCRIPT

UIDs, which are published to the PIP, for a given room do not match, the PDP will infer such malicious behaviour and subsequently revoke access to resources. In addition, appropriate actions can be taken by the system administrator. We also note that Android provides Device Administrator APIs for BYOD scenarios, which allow enterprises to take control of sensitive resources and modify system configurations on their employees' devices. Through such APIs, an enterprise can then permanently enable Bluetooth services.

Another attack vector involves an unauthorized individual obtaining an authorized user's phone, whether by theft or voluntary provision. If such an attack occurs, then the unauthorized individual can gain access to restricted resources. In reality, this sort of attack exploits social engineering and/or insider threats that are usually already covered as part of an enterprise's global security efforts. Nevertheless, in our extended system, we mitigate this previously unaddressed attack vector by incorporating mechanisms that are able to determine biometric signatures for every user in the system. In particular, we employ behavioral (i.e., dynamic gait analysis) and physiological (i.e., fingerprint analysis) biometric authentication, which ensures that unauthorized users will not be able to bypass security by using someone else's device.

One of the objectives of our paper is to address context monitoring issues including adversarial context manipulation via passive attacks (e.g., malicious relay of BLE beacons). However, we emphasize that if we relax assumptions stated in Section 5 and elevate the adversary's capabilities to active attacks, we envision two active attack vectors the adversary could employ that would *not* circumvent the security of the system: packet injection and Denial-of-Service (DoS). An astute active attacker would determine that an advantage cannot be gained by injecting packets at either Step 3 or Step 4 of the co-proximity authentication protocol (Figure 5). Intercepting the BLE beacon that encapsulates *nonce* at Step 3 and transmitting a new malicious beacon that encapsulates a *nonce'*, which would be now received by the Client, is unnecessary. The original *nonce* is transmitted in cleartext, thereby allowing the adversary to simply record the observed value, which may be potentially used in Step 4. However, the attacker again would not need to inject packets in Step 4 since the attacker has acquired the information needed (i.e., *nonce*) to extract and calculate the temporary session key (SK) from the BLE beacon sent by the Client. Knowledge of *nonce* and SK also does not violate the security of the system (see Section 6.4). In summary, injecting attacker-generated BLE beacons would serve no purpose towards the goal of fooling the CASSEC 2.0 system into establishing co-proximity between the PM and the Client.

Given that intercepting and subsequently injecting malicious BLE beacons between the PM and the Client would serve no purpose towards circumventing co-proximity authentication, an active attacker may instead rely on DoS attacks. A malicious user may attempt a DoS attack by acquiring a high-powered Bluetooth-enabled device [8, 13, 42]. Specifically, the user first adjusts the special device to mimic his original (or another user's) smartphone's UID, and then boosts the signal strength. As a consequence, receivers in different rooms within a certain radius may incorrectly publish the proof-of-location. Therefore, the PDP will believe that multiple violations are occurring. First, the system is inherently resistant to such attack. Because of signal attenuation, proximity modules, which have lower transmission capabilities than of the adversary's high-powered device, may not be able to transmit the challenge beacon to the malicious device, which may be potentially far from the proximity module. However, if reception of the challenge does occur, several methods could be employed to counteract this attack. For example, our study shows that the majority of beacon RTT values fell between 72ms and 86ms. The PM could invalidate the challenge, thereby invalidating the corresponding response, after 86ms has elapsed. We emphasize that we are describing this DoS attack under the assumption that the attacker is able to somehow relay the temporary session key (once it has been decrypted) from his Client to the special device, otherwise the objective of the DoS attack would be to simply waste computing resources by repeatedly initiating the co-proximity authentication process.

The results in Section 7.4 demonstrated that no significant statistical variations of RTT values exist when the PM and the Client device executed co-proximity authentication at distances between 1-6ft. Consequently, time-based distance estimation techniques that rely on BLE beacons as its underlying technology are not reliable methods for differentiating between adjacent proximity zones *within* an enterprise environment. However, such techniques may be resistant to an adversary's attempts to execute relay attacks when the Client is far away, that is, *outside* the enterprise environment. Let us assume the adversary's attack takes the form of a ghost-and-leech attack vector [21] in which the adversary employs two relay devices ($\mathcal{A}_{\mathcal{PM}}$, $\mathcal{A}_{\mathcal{C}}$) that are each within 6ft of the PM and the Client, respectively, and the two malicious devices communicate over a high-speed connection. Let us also assume that $\mathcal{A}_{\mathcal{PM}}$ and $\mathcal{A}_{\mathcal{C}}$ have similar hardware and software to that of the Client and PM, respectively. The total RTT (RTT_T) is the sum of the RTT values between the PM and $\mathcal{A}_{\mathcal{PM}}$ (RTT_{PM}), $\mathcal{A}_{\mathcal{PM}}$ and $\mathcal{A}_{\mathcal{C}}$ (RTT_{PMC}), which consists of RTT values between the network communication nodes that support the high speed connection, and $\mathcal{A}_{\mathcal{C}}$ and the Client (RTT_C). The communication relationship between said entities is visually depicted as:

$$PM \text{ — } \mathcal{A}_{\mathcal{PM}} \cdots \mathcal{A}_{\mathcal{C}} \text{ — } Client$$

$$RTT_T = RTT_{PM} + RTT_{PMC} + RTT_C$$

It is difficult to approximate RTT_{PM} and RTT_{PMC} because their values are significantly influenced by and dependent on many factors (e.g., communication nodes' connection medium, network traffic load, propagation delay, etc.). However, since the beacon transmission between $\mathcal{A}_{\mathcal{C}}$ and Client simulates the transmission between the PM and the Client as consequence of employing similar hardware and software, we are able to approximate RTT_C to 81ms based on our study. In addition, the triangle inequality theorem ensures us that $RTT_{PM} + RTT_{PMC} > RTT_C$ since the path from the PM to the Client is not a direct route. Thus, the RTT threshold (RTT_{TH}) should be set to 81ms for each legitimate proximity module to prevent relay attacks when the Client is outside the enterprise environment.

ACCEPTED MANUSCRIPT

8.2 WiFi Manipulation

We leverage the WiFi signal interference caused by human activity to determine the number of occupants in a given room. A malicious user could attempt a DoS by disrupting WiFi signals. That is, an attacker could acquire a special device that would, for example, completely nullify WiFi signals [8, 13, 42]. Another means to circumvent the system would be to obstruct the LOS with something other than a human body such as a chair. Therefore, in either case, when the receiver processes the signal interference, it may publish an incorrect number of users within that room. However, the authorization server will detect violations because inconsistencies will exist within the PIP.

Regardless of whether Bluetooth or WiFi manipulation is employed, the scenarios that we address make it more difficult to circumvent CASSEC. That is, in both the SoD Scenario and the AOU scenario, multiple users with mutual interests must collude and agree in order to attempt bypassing the system.

8.3 True Continuous Authentication

Our passive biometric authentication scheme only provides continuous authentication while the Client smartphone device is within the user's pocket. It is possible that an authorized user, whom the Client had previously authenticated, simply removes the device from the pocket, and subsequently gives the device to an unauthorized user. Consequently, the device is unlocked and its content is accessible by the unauthorized user. Therefore, other biometric authentication must be used. While there are both active and passive biometric authentication solutions, passive solutions should be used to maximize usability as they would not require users to actively authenticate themselves. To protect against such an attack, other passive biometric techniques to continually authenticate while the user is holding the phone should be used. Some biometric features that could be analyzed and passively authenticated include timing of keystrokes, touchscreen behavior, face, retina, or iris [2, 35, 37]. In fact, passive facial recognition technology has been recently (Nov, 2017) integrated into the Apple's new flagship mobile device: iPhone X [5].

9. RELATED WORK

The role-based access control (RBAC) model is mainly used in enterprise settings to facilitate administration of access control polices [16]. In such settings, users are assigned different roles whereby each role is granted predefined access privileges to enterprise resources. Various access control models and systems have been proposed that use RBAC as a foundational paradigm, and some augment the model so that privileges associated with a role can only be exercised if contextual parameters are adhered to. The most common extension is the inclusion of spatial constraints. GEO-RBAC is a spatially-aware RBAC model that defines the concept of spatial roles which allow an authorized user to assume a role (i.e., role enabling) and exercise its associated privileges (i.e., role activation) only if the user is at or within a designated location specified by physical coordinates [9]. LoT-RBAC and STARBAC are other augmented RBAC models that incorporates spatio-temporal constraints for role enabling and role activation [1, 12]. Such models, however, are not implemented and therefore no enforcement mechanism has been developed to support these models.

Proximity-based Access Control (PBAC) [19] is an access control model developed specifically for Smart-Emergency Environments that takes into account the user's proximity to a resource (e.g., a computer). Prox-RBAC, which extends GEO-RBAC, is a formal authorization model based on a notion of proximity [23]. That is, access control decisions are not solely based on the requesting user's location, but also on the location of other users in the physical space. Prox-RBAC incorporates elements of the $UCON_{ABC}$ usage control model [32]. Prox-RBAC has been further extended to incorporate a large variety of proximity constraints in addition to the spatial ones, namely attribute-based, social, cyber, and temporal proximity constraints [18].

Prox-RBAC was implemented using near-field communication (NFC) allowing a NFC-enabled phone to transmit signals to a NFC reader to lock and unlock a door. Although it provides high-integrity proof of location, it requires user intervention; the user must initiate close-quarters contact between the phone and the NFC reader. Integration of NFC capabilities has become standard in state-of-the-art smartphones, and therefore, an enterprise would be required to expend financial resources to deploy NFC readers for each door in the enterprise setting to fully deploy the Prox-RBAC system. On the other hand, PBAC was implemented using ultra-wide band RFID which calculated AoA and ToA to support automated access access control. Although the system did not require user intervention, active tags (worn by users) and mounted receivers had to be deployed to determine the tags position. Similarly, an enterprise would incur costs to deploy the PBAC system. Moreover, many systems, including Prox-RBAC and PBAC, inherently assume that every individual within a monitored space is trusted. Systems that are solely based on location tracking devices worn or held by users can be easily circumvented through collusion. Consider the two motivating scenarios discussed in Section 2 on which our work is based. In the SoD scenario, which assumes top secret documents to be stored within a protected office, one of the Generals that has a high security clearance will unlock the door with his/her tracking device (e.g., NFC), but a Private can easily follow immediately behind prior to the door locking. By not initiating contact between the transmitter and the receiver, the system would be tricked into believing that no unauthorized personnel is occupying the protected office. The AOU scenario requires that an eyes-only, restricted document to be accessible by a Private only when no other individuals are in the vicinity. However, a Civilian, assuming he/she was given a tracking device, can simply remove the device (e.g., active tag) so as to not be tracked. In addition, costs for deployment and management of these systems, and others used in similar architectures, remain significant and limit the widespread adoption of these systems. Unlike CASSEC 2.0, neither Prox-RBAC or PBAC addressed a major security problem of a user obtaining an authorized user's phone, whether by theft or voluntary provision. Consequently, individuals may be able to circumvent the access control system via collusion, allowing one individual to impersonate another individual by exchanging tracking devices. Various solutions have been proposed to prevent such an attack, but require special hardware to deploy [2]. The approach by Wang *et al.* [41], for example, explores biometric signatures using WiFi-based techniques, but it requires one receiver and one transmitter for *every* user in order to distinguish multiple subjects at

ACCEPTED MANUSCRIPT

a time, which is not practical for enterprise environments. Instead, our goal is to utilize commodity hardware widely available in enterprise environments: smartphone devices. In CASSEC 2.0, we leverage the accelerometer and fingerprint sensors within smartphones to achieve physiological and behavioral biometric authentication, and thereby enabling our system the ability to determine with high confidence whether the current user is the true owner of the device.

XACML is a standardized access control policy language and an abstract enforcement model. In our work, we leverage the syntactical structure of XACML policies (specified in XML) as well as the main components in its enforcement mechanism. Specifically, our policy specification is written in XML (Section 4), and our underlying PrBAC reference architecture is the same as the one of XACML. However, the communication model as well as the duties of each architectural component differ in CASSEC 2.0 as our objective is to provide an abstract context-aware system architecture to support an automated access control system.

10. CONCLUSIONS

In this paper, we propose a proximity-based context-aware access control mechanism that also incorporates constraints concerning the confidence about user and location information. Such constraints allow the system to make decisions based on the *degree of reliability* of extracted contextual information. We have integrated such mechanisms into CASSEC 2.0 and have conducted a feasibility study to show our approach is viable in practice. We have evaluated our confidence constructs and collected some data by implementing behavioral and physiological biometric authentication and extending the occupancy detection mechanism with a robust co-proximity authentication protocol that is resistant against relay attacks. Currently, the biometric authentication component leverages both passive and active techniques. Active techniques require user intervention (e.g., requesting the user to place his finger on a fingerprint reader), which may negatively impact the user experience and user workplace productivity. Even more, such active techniques only authenticate the legitimate user at the time of access, thereby possibly enabling an unauthorized user, who is in proximity to the legitimate user, to gain access to restricted resources subsequent to initial authentication (see Section 8). To maximize the usability and security of the solution, we plan to extend CASSEC 2.0 to support multi-factor passive biometric authentication in order to achieve true continuous authentication, which would allow on-going user verification while data is being accessed.

Acknowledgement

Funding: The work was supported by the NSF [grant number CNS-1111512] and the Center for Science of Information (CSoI), an NSF Science and Technology Center [grant number CCF-0939370].

11. REFERENCES

[1] S. Aich, S. Sural, and A. K. Majumdar. Starbac: Spatiotemporal role based access control. In *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS*, pages 1567–1582. Springer, 2007.

[2] A. Alzubaidi and J. Kalita. Authentication of smartphone users using behavioral biometrics. *IEEE Communications Surveys & Tutorials*, 18(3):1998–2026, 2016.

[3] A. Anderson. Xacml profile for role based access control (rbac). *OASIS Access Control TC committee draft*, 1:13, 2004.

[4] Android. Android developer's guide. http://developer.android.com.

[5] Apple. iphone x. https://www.apple.com/iphone-x/.

[6] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves. Radio link quality estimation in wireless sensor networks: a survey. *ACM Transactions on Sensor Networks (TOSN)*, 8(4):34, 2012.

[7] B. Balaji, J. Xu, A. Nwokafor, R. Gupta, and Y. Agarwal. Sentinel: occupancy based hvac actuation using existing wifi infrastructure within commercial buildings. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, page 17. ACM, 2013.

[8] S. Banerjee and V. Brik. Wireless device fingerprinting. In *Encyclopedia of Cryptography and Security*, pages 1388–1390. Springer, 2011.

[9] E. Bertino, B. Catania, M. L. Damiani, and P. Perlasca. Geo-rbac: a spatially aware rbac. In *Proceedings of the tenth ACM symposium on Access control models and technologies*, pages 29–37. ACM, 2005.

[10] M. Bocca, O. Kaltiokallio, and N. Patwari. Radio tomographic imaging for ambient assisted living. In *Evaluating AAL Systems Through Competitive Benchmarking*, pages 108–130. Springer, 2012.

[11] R. Bruno and F. Delmastro. Design and analysis of a bluetooth-based indoor localization system. In *IFIP International Conference on Personal Wireless Communications*, pages 711–725. Springer, 2003.

[12] S. M. Chandran and J. B. Joshi. Lot-rbac: a location and time-based rbac model. In *Web Information Systems Engineering–WISE 2005*, pages 361–375. Springer, 2005.

[13] L. C. C. Desmond, C. C. Yuan, T. C. Pheng, and R. S. Lee. Identifying unique devices through wireless fingerprinting. In *Proceedings of the first ACM conference on Wireless network security*, pages 46–55. ACM, 2008.

[14] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on information theory*, 29(2):198–208, 1983.

[15] J. El-Sobhy, S. Zickau, and A. Kupper. Proximity-based services in mobile cloud scenarios using extended communication models. In *Cloud Networking (CloudNet), 2015 IEEE 4th International Conference on*, pages 125–131. IEEE, 2015.

[16] D. Ferraiolo, D. R. Kuhn, and R. Chandramouli. *Role-based access control*. Artech House, 2003.

[17] S. K. Ghai, L. V. Thanayankizil, D. P. Seetharam, and D. Chakraborty. Occupancy detection in commercial buildings using opportunistic context sources. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, pages 463–466. IEEE, 2012.

ACCEPTED MANUSCRIPT

[18] A. Gupta, M. S. Kirkpatrick, and E. Bertino. A formal proximity model for rbac systems. *Computers & Security*, 41:52–67, 2014.

[19] S. K. Gupta, T. Mukheriee, K. Venkatasubramanian, and T. Taylor. Proximity based access control in smart-emergency departments. In *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, pages 5–pp. IEEE, 2006.

[20] Y. Jiang, X. Pan, K. Li, Q. Lv, R. P. Dick, M. Hannigan, and L. Shang. Ariel: Automatic wi-fi based room fingerprinting for indoor localization. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 441–450. ACM, 2012.

[21] Z. Kfir and A. Wool. Picking virtual pockets using relay attacks on contactless smartcard. In *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*, pages 47–58. IEEE, 2005.

[22] M. S. Kirkpatrick and E. Bertino. Enforcing spatial constraints for mobile rbac systems. In *Proceedings of the 15th ACM symposium on Access control models and technologies*, pages 99–108. ACM, 2010.

[23] M. S. Kirkpatrick, M. L. Damiani, and E. Bertino. Prox-rbac: a proximity-based spatially aware rbac. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 339–348. ACM, 2011.

[24] Y. Kumar, R. Munjal, and H. Sharma. Comparison of symmetric and asymmetric cryptography with existing vulnerabilities and countermeasures. *International Journal of Computer Science and Management Studies*, 11(03), 2011.

[25] A. Larchikov, S. Panasenko, A. V. Pimenov, and P. Timofeev. Combining rfid-based physical access control systems with digital signature systems to increase their security. In *Software, Telecommunications and Computer Networks (SoftCOM), 2014 22nd International Conference on*, pages 100–103. IEEE, 2014.

[26] L. Lee and W. E. L. Grimson. Gait analysis for recognition and classification. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 148–155. IEEE, 2002.

[27] M. Moreno, J. L. Hernandez, and A. F. Skarmeta. A new location-aware authorization mechanism for indoor environments. In *Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on*, pages 791–796. IEEE, 2014.

[28] T. Moses et al. Extensible access control markup language (xacml) version 2.0. *Oasis Standard*, 200502, 2005.

[29] T. T. Ngo, Y. Makihara, H. Nagahara, Y. Mukaigawa, and Y. Yagi. The largest inertial sensor-based gait database and performance evaluation of gait-based personal authentication. *Pattern Recognition*, 47(1):228–237, 2014.

[30] O. Oluwatimi, D. Midi, and E. Bertino. A context-aware system to secure enterprise content. In *Proceedings of the 21st ACM on Symposium on Access Control Models and Technologies*, pages 63–72. ACM, 2016.

[31] O. Oluwatimi, D. Midi, and E. Bertino. Overview of mobile containerization approaches and open research directions. *IEEE Security & Privacy*, 15, 2016.

[32] J. Park and R. Sandhu. The ucon abc usage control model. *ACM Transactions on Information and System Security (TISSEC)*, 7(1):128–174, 2004.

[33] K. B. Rasmussen, C. Castelluccia, T. S. Heydt-Benjamin, and S. Capkun. Proximity-based access control for implantable medical devices. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 410–419. ACM, 2009.

[34] Y. Ren, Y. Chen, M. C. Chuah, and J. Yang. User verification leveraging gait recognition for smartphone enabled mobile healthcare systems. *IEEE Transactions on Mobile Computing*, 14(9):1961–1974, 2015.

[35] O. Riva, C. Qin, K. Strauss, and D. Lymberopoulos. Progressive authentication: Deciding when to authenticate on mobile phones. In *USENIX Security Symposium*, pages 301–316, 2012.

[36] T. Saelim, P. Chumchu, and T. Mayteevarunyoo. Design and performance evaluation of novel location-based access control algorithm using ieee 802.11 r. *Journal of Convergence Information Technology*, 10(4):33, 2015.

[37] H. Saevanee and P. Bhattarakosol. Authenticating user using keystroke dynamics and finger pressure. In *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE*, pages 1–2. IEEE, 2009.

[38] B. Shebaro, O. Oluwatimi, and E. Bertino. Context-based access control systems for mobile devices. *Dependable and Secure Computing, IEEE Transactions on*, 12(2):150–163, 2015.

[39] T. Søndrol. Using the human gait for authentication. Master's thesis, GjÃÿvik University College, 2005.

[40] M. Vossiek, L. Wiebking, P. Gulden, J. Wieghardt, C. Hoffmann, and P. Heide. Wireless local positioning. *Microwave Magazine, IEEE*, 4(4):77–86, 2003.

[41] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu. Understanding and modeling of wifi signal based human activity recognition. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 65–76. ACM, 2015.

[42] Q. Xu, R. Zheng, W. Saad, and Z. Han. Device fingerprinting in wireless networks: Challenges and opportunities. *IEEE Communications Surveys & Tutorials*, 18(1):94–104, 2016.

[43] F. Zafari, I. Papapanagiotou, and K. Christidis. Microlocation for internet-of-things-equipped smart buildings. *IEEE Internet of Things Journal*, 3(1):96–112, 2016.

Oyindamola Oluwatimi is a PhD candidate in the college of Computer Science at Purdue University. Oyindamola's research area is mobile security and privacy. He focuses on developing and enhancing access control techniques for mobile devices within enterprise mobile information technology infrastructures. In addition, Oyindamola investigates the use of contextual information extracted from the environment, within the physical and computing realms (e.g., user

ACCEPTED MANUSCRIPT

physical activity, time, location, etc.), to influence access control decisions in context-aware systems.

Maria Damiani is an Associate Professor of Computer Science at the University of Milan. She teaches at the School of Science and at the School of Humanities. She is an Action Editor for GeoInformatica, member of the editorial board for Transactions on Data Privacy, and Associate Editor for IEEE Transactions on Dependable and Secure Computing.

Elisa Bertino is professor of Computer Science at Purdue University. She is a Fellow of IEEE, ACM, and AAAS. She is currently serving as editor in chief of IEEE Transactions on Dependable and Secure Computing.

ACCEPTED MANUSCRIPT