applied system innovation

Article

Design and Implementation of a Contextual-Based Continuous Authentication Framework for Smart Homes

Yosef Ashibani *, Dylan Kauling and Qusay H. Mahmoud

Department of Electrical, Computer and Software Engineering, University of Ontario Institute of Technology, Oshawa, ON L1G 0C5, Canada; dylan.kauling@uoit.net (D.K.); qusay.mahmoud@uoit.ca (Q.H.M.)

* Correspondence: yosef.ashibani@uoit.net

Received: 16 November 2018; Accepted: 9 January 2019; Published: 17 January 2019

Abstract: There has been a rapid increase in the number of Internet of Things (IoT) devices in the last few years, providing a wide range of services such as camera feeds, light controls, and door locks for remote access. Access to IoT devices, whether within the same environment or remotely via the Internet, requires proper security mechanisms in order to avoid disclosing any secure information or access privileges. Authentication, on which other security classes are built, is the most important part of IoT security. Without ensuring that the authorized party is who it claims to be, other security factors would be useless. Additionally, with the increased mobility of IoT devices, traditional authentication mechanisms, such as a username and password, are less effective. Numerous security challenges in the IoT domain have resulted in the proposal of many different approaches to authentication. Many of these methods require either carrying an authentication token, such as a smartcard, or restricting access to a particular physical location. Considering that most IoT devices contain a wide array of sensors, a large amount of contextual information can be provided. Thus, real-time security mechanisms can protect user access by, for example, utilizing contextual information to validate requests. A variety of contextual information can be retrieved to strengthen the authentication process, both at the time of access request and throughout the entire access session, without requiring user interaction, which avoids the risk of being discovered by attackers of these features. In this paper, we introduce a continuous authentication framework that integrates contextual information for user authentication in smart homes. The implementation and evaluation show that the framework can protect smart devices against unauthorized access from both anonymous and known users, either, locally or remotely, in a flexible manner and without requiring additional user intervention.

Keywords: contextual information; quality of contextual information; continuous authentication; IoT; smart homes

1. Introduction

Due to the growth of available services over the last few years, the amount of information exchanged between Internet of Things (IoT) devices has been on the rise, highlighting the important issue of security. IoT security has been the primary concern of most users, as well as one of the most important reasons for determining the adoption of IoT applications, such as the smart home. Security is an extensive field that is commonly divided into four classes: authenticity, validating the claimed identity; confidentiality, securing the communication channels among communicating parties; integrity, protecting exchanged information against alteration; and availability, protecting the availability of the service [1]. Authenticity is ranked first because it is the most important part of security, being that on which the other security classes are built and, because, without ensuring that the authorized party is who it claims to be, other security factors would be useless. Thus, to achieve a degree of higher security, it is necessary to ensure the correct identity, which can be accomplished by a robust authentication technique [2,3].

Authentication approaches commonly fall into three categories: knowledge-based, object-based, and physiological. However, each of these approaches has its drawbacks. The knowledge-based approach needs users to remember authentication credentials, such as username and password, which may be susceptible to attacks in two ways: users keep their credentials when having many—especially complicated credentials that are difficult to remember—in a written form and, in scenarios such as brute force, dictionary, malware, video recording, and phishing attacks, this may be open to compromise. Although, as demonstrated in [4], the password is still the preferred option for authentication by many users, many incidents of password attacks have occurred during the last few years [5]. For example, the usernames and passwords of 68 million accounts in Dropbox have been breached online. MySpace also confirmed that the passwords and email addresses of hundreds of millions of their users were stolen in 2012 hacks [6]. Object-based techniques that require users to have tokens are susceptible to loss or theft. Moreover, it is impractical for users to regularly carry tokens, as well as unrealistic for those users who have many identification tokens [7]. Physiological approaches are the most robust techniques available for user authentication [8].

The previously mentioned authentication techniques are typically applied at the start of a session, and grant access to users until a given time or until they manually log out. During the session, all privileged services, applications, and information are accessible to the authenticated user. Consequently, sensitive information is susceptible to misuse by an unauthorized user when gaining access to devices or services immediately after authentication is completed.

Therefore, there is a need for continuous authentication techniques that periodically examine the access situation and the legitimacy of the communicated party [9]. By involving contextual information in the authentication process, users will be able to access devices and services with enhanced security [10]. Such information can be obtained from different resources of context, such as the environment, a device, a network, or online resources, which, in turn, can be incorporated into the authentication decision, either as separate authentication factors or as an authentication adaptation, augmenting the decision to grant a user access to a given resource. Even though considering a high number of attributes for authentication would increase security and reliability, the associated costs regarding processing time would also increase. Thus, any proposed solution should include contextual attributes that do not need user interaction, and any verification functions should work in the background without resulting in any response delay. There is much contextual information that can be included as further support for traditional authentication mechanisms. Such authentication techniques would be flexible for users and provide security in a simple manner without requiring much user interaction. As a result, there is the need for a security mechanism that can acclimate based on surrounding changes, and that can adopt security policies accordingly.

Motivation

Many smart home devices, including baby monitors and cameras, garage door locks, and smart lighting, can be controlled through end-user devices, such as smartphones and tablets. As an example, smart home applications connect many devices and objects to each other, and the interconnection of these devices can be achieved wirelessly and through the Internet to users' end-devices, making daily life easier. On the other hand, these devices are susceptible to theft or loss. Thus, the use of these devices by unauthorized users will lead to unauthorized access to home services, consequently allowing access to the user's information. Many examples of security incidents have been seen throughout the literature, such as unauthorized access to baby monitors and thermostats, allowing access to home networks as a result of using weak or default passwords. Hence, any proposed solution for securing smart homes should consider continuous user authentication for realizing any change in the user's context. Behavior profiling can be used for the continuous authentication of users, based on how they behave or what they do. For example, end-user devices, such as smartphones, smartwatches, and other wearable technologies, have several sensors that can help to characterize the user and usage patterns [11–13].

Behavior profiling, which has many advantages [14], including the ability to be continuously computed without the knowledge of the user, can be obtained without additional hardware. As an example, usage of applications (apps) [15,16] and location tracking can be employed for continuous user authentication. This research is motivated by multi-factor authentication, which requires two or more authentication factors, such as a password, token or proximity of the requesting user. However, many of these factors involve physical or static tokens, which are only considered as a point-of-entry, and are susceptible to theft or loss, in addition to not considering the provision of continuous authentication for the entire access session. Common problems that are highlighted in the current literature include:

- The absence of clearly defined types of contextual information that could be leveraged for security
- The lack of considering continuous authentication by many proposed approaches beyond point-of-entry
- The lack of considering contextual information as a part of authentication or the tendency to consider only one or two attributes
- The present lack of concern for user privacy with advanced authentication methods

Thus, adopting different types of contextual information related to users, services, and devices together will uniquely define the users' access requests and monitor their access sessions. Thus, it is convincing that including contextual information in the authentication process will increase the level of security beyond point-of-entry by providing non-intrusive, convenient, and continuous security throughout the access session. This is possible by enhancing standard knowledge-based static authentication with contextual information.

This paper presents a framework that utilizes contextual information for continuous authentication. We first present a taxonomy of contextual information that could be leveraged for security purposes, and then examine the available contextual information with regards to its permission requirements and retrieval time. According to the categorization of the IoT applications as presented in [17], the implementation and evaluation of the proposed framework is based on the first category, which is the smart home scenario. Following this approach, we introduce a continuous authentication mechanism for smart homes that integrate retrieved contextual information in a real-time manner. To this end, the contributions of this paper are:

(1) A taxonomy of contextual information that could be leveraged for security purposes, and an examination of the available contextual information with regards to its permission requirements and retrieval time.

(2) An authentication framework that utilizes contextual information as a factor for user authentication, and continuous user authentication during the access session, allowing access to devices and services without necessarily requiring constant interactions, and simply supplying non-expert users with options for configuring the security of their network regarding authentication.

(3) An implementation of the prototype and evaluation regarding: overheads imposed on the system; authentication assigning weights and thresholds; handling of multiple simultaneous requests; and the ability of the prototype to protect against attacks. Evaluation results show that our framework provides security in a flexible manner without requiring additional intervention by users.

The rest of this paper is organized as follows: Section 2 discusses the related work. The details of our proposed framework are presented in Section 3. The implementation of the proof of concept prototype is discussed in Section 4. Evaluation results are presented in Section 5, and security analysis are discussed in Section 6. Finally, Section 7 concludes the paper and offers ideas for future work.

2. Related Work

Traditional authentication techniques, such as usernames and passwords, are insufficient for a mobile environment where users require access from different locations. This requires the adoption of alternative authentication approaches, or enhancements to traditional authentication methods. To this end, many alternative authentication approaches, that can be adapted for smart homes, have been proposed.

Single sign-on (SSO) is a user authentication technique that uses a single login credential to access multiple applications [18–20]. This method does not require users to provide login credentials for many applications. However, it could increase the risk of hacking all other applications that can be accessed under the same authentication if, for example, the login credentials are attained by an attacker. Moreover, SSO is still point-of-entry, and it is impractical to require users to continuously provide SSO credentials for authentication. Additionally, authentication mechanisms such as SSO assume that the legitimate user is the one who accesses the service throughout the entire access session [20]. Token-based authentication approaches have been developed for overcoming the drawbacks of secret knowledge-based approaches such as passwords. Token-based authentication methods can be classified into two categories: hardware tokens and software tokens. Hardware token mechanisms need the user to carry tokens. However, physical tokens, which are only considered as point-of-entry, are susceptible to theft or loss, and are also difficult to utilize for continuous authentication throughout an entire access session. Software tokens use the end-user's device by, for example, sending the password as a message to the user's registered device, such as mobile phone, to be used for logging in to the service. A software token, such as Google Authenticator, in the form of software installed on the end-user's device, issues a new password (time-based one time password) that changes with every access time [20,21]. Token-based authentication can provide some advantages over knowledge-based authentication techniques as they remove the burden from users having to remember and choose robust passwords. However, it is inconvenient for users to have to continuously provide generated passwords for continuous authentication throughout the entire access session.

In order to overcome the drawbacks of using static authentication credentials, another technique that involves biometric features, such as fingerprints, which are being developed and used especially for specific locations where users are expected to access the required services [22–24]. Although these attributes are considered to be strong against attacks, they are impractical for mobile users who need access to services from different places. Biometric features, such as fingerprint or facial recognition, can be considered as robust for authentication, and would have a very small response time limitation in the smart home environment. However, not all smart devices are provided with physiological biometrics such as fingerprint or iris recognition capabilities. In addition, it would be inconvenient to require users to re-enter their biometrics every time for continuous authentication. Moreover, one of the issues with biometric authentication features is that they are not privacy-preserving [25], which limits the adoption of biometrics for many applications. For example, one of the issues regarding biometrics is that it is impossible to revoke registered biometrics information, in cases where is compromised by an attacker [26]. In addition, the system of biometrics is not widely accepted by users because it is perceived as intrusive and a violation of personal privacy [27]. While technologies such as iris scanners are readily available on smartphones, the recent case of the Samsung Galaxy S8 iris scanner demonstrated how easy this technology can be deceived by using an infrared image and a contact lens [28].

In general, authentication approaches that are based on usernames and passwords or tokens prove only the presence of the identity, and not the user. Thus, if such credentials or tokens are stolen, loaned, misused, or forged, there will be no proof of the user's legitimacy. The same issue exists with biometric-based authentication techniques. If an illegitimate user is using the end-user's device before the expiry of the access session, all applications and services that are accessed via this device could be accessed by unauthorized persons since there is no technique for verifying if the current user is authorized one.

Much research has been conducted into enhancing traditional authentication techniques with contextual information. Table 1 provides a comparison of some relevant related works regarding utilized information, advantages, and limitations.

The authors in [9] provide an authentication approach that involves GPS location, time and the tasks performed on the operating system. However, their approach considers location based on a GPS signal that will not be available when accessing services indoors. Based on the assumption that mobile users tend to use their apps in different locations at different times, location-based authentication is described in [29]. Assuming that users perform similar tasks at certain times during weekdays, a user profile approach that gathers behavioral information, such as text messages, calls and geographic location, is proposed in [30]. For observed actions, such as habitual or good events, this approach assigns a score. The study in [31] proposes an anomaly-based detection system based on monitoring users' actions, such as sending SMS messages or making calls. Additionally, other studies, such as [29–31], consider text messages and calling behavior for abnormal usage detection. As the number of mobile apps increases, SMS and calling functions are being ignored and replaced with apps that achieve the same purpose. In addition, unauthorized users who access mobile devices will, most probably, tend to operate inconspicuously, hence these functions provide insufficient evidence of the intended user.

Table 1. A comparison of some relevant related works.

No.	Utilized Information	Advantages	Limitations
[8]	Location and e-receipt	Provides authentication model for mobile environments	Only includes location and e-receipt as contextual information based on the user's presence
[9]	Location, time and operating system processes	Provides context-aware authentication and implementation	Only uses GPS location, which will not be available in most cases, especially indoors
[21]	SMS	Provides transparent and continuous authentication in addition to implementation	SMS function is ignored and replaced with apps that achieve the same purpose.
[29]	Calls, SMS and GPS based location.	Enables implicit and continuous authentication	Only uses GPS location, which will not be available in most cases, especially indoors. In addition, SMS and calling functions are ignored and replaced with apps that achieve the same purpose.
[30]	SMS, calls and geographic location	Provides implicit continuous authentication	SMS and calling functions are ignored and replaced with apps that achieve the same purpose. Only used GPS coordinates collected, which will not be available in most cases, especially indoors.
[31]	Telephone calls and SMS	Provides illegitimate user detection	SMS and calling functions are ignored and replaced with apps that achieve the same purpose.
[32]	Wearable clothing colors	Provides continuous user authentication	Unsuitable, for example, in an environment with a uniform clothing style; also restricts users to a specific type of clothing whenever they want to access the desired service.
[34]	Small hardware tokens	Provides continuous authentication based on the user's presence	Limits access to the use of location-based contexts
[35]	Hardware token (RFID tags) location and profile	Perform authentication and access control approach in a very flexible and scalable model	Limited to only using locations and profiles; also does not provide any implementation or evaluation of the proposed framework.
[36]	Location	Provides an analysis of the requirements of the design and implementation	Only uses Wi-Fi based location; no evaluation is provided.

An approach for recognizing users based on wearable clothing colors as supportive identification attributes is proposed in [32]. While this approach may support the authentication process, it is unsuitable, for example, in an environment with a uniform clothing style and also restricts users to a specific type of clothing whenever they want to access the desired service. A continuous authentication method for wearable wireless devices is suggested in [33]. Taking into account that such devices are susceptible to theft or loss, this approach is impractical for mobile users. In [34], a transient authentication mechanism is proposed for user authentication through small hardware tokens, limiting access to the use of location-based contexts.

A contextual attribute authentication model for mobile environments is defined in [8]. However, this model only includes location and e-receipt as contextual information. Even though the authors in [35] provide a context security approach, the proposed scheme is limited to only using locations and profiles. Furthermore, it does not provide an implementation and evaluation of the proposed framework. A Wi-Fi-based IoT smart home system is depicted in [36], and includes the requirements of the design and implementation. As an example, the proposed system provides user access and configures and controls the system with an easy user interface running on mobile devices. However, it does not involve contextual information for the authentication process.

Our work differs from the aforementioned solutions in several ways. It not only depends on static credentials or tokens but also on available contextual information such as location, calendar and profile. Moreover, this study examines the available contextual information in regards to the permission requirements and retrieval time to be used for authentication and supports security questions in a different way, by using profile information rather than predefined (static) security questions, which are often forgotten. Finally, it provides continuous authentication by checking contextual information in real time during the access session without user intervention unless the situation requires it.

3. Framework Design

Our framework allows for contextual information to be obtained [37,38], and integrated for the continuous authentication of mobile clients (users) to access smart home devices (services) beyond the initial login using the traditional credentials of username and password. This section presents a contextual information taxonomy with a brief discussion of an important factor, namely, the Quality of the Contextual Information. Furthermore, it describes the high-level architecture of the proposed framework, along with a use case scenario for user authentication in the smart home.

3.1. Contextual Information

Although there are many contextual information classifications, there is still no unified definition for contextual information. Many researchers consider the following definition as appropriate: "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications [39]". Context includes any information that is related to a user's situation, such as location, device status, and any information related to the environment, such as temperature, loudness, and brightness. Another classification of security-relevant contextual attributes includes physical environment context, (e.g., time or temperature); service type context (e.g., premium or basic service); user's context (e.g., location); platform context (e.g., trusted state of the platform); and particular transaction (e.g., an electronic token or electronic receipt) [8]. In addition, contextual information includes personal contexts (e.g., preferences); activity contexts (e.g., meeting schedule or shopping list); physical contexts (e.g., time and location); device contexts (e.g., power and display size); systematic contexts (e.g., network bandwidth); application contexts (e.g., agent and service); and environmental contexts (e.g., light level) [25,40].

3.1.1. Contextual Information Taxonomy

There are many contextual information classifications. For example, [41] classifies contextual information based on operational categorization to sensed, static, profiled, and derived contexts. In [42], contextual information is classified based on three common questions regarding where you are, who you are with, and what resources are nearby. However, from a security perspective, especially for the authentication process, contextual information can be classified as follows:

- Direct Contextual Information: This can be achieved directly without performing operations on, or modifying, the contextual values (e.g., activities or historical movement information,) as well as social relationships (e.g., friends and family). Direct contextual information, which is mentioned in [43] as the primary context, includes any retrieved contextual information without relying on already existing contextual information. Direct contextual information includes:
    - User context: any information about the user such as profile, calendar, social networks, and access patterns
    - Device context: any information related to the used end-device which can be retrieved from sensors, such as location, current and voltage values, Wi-Fi access points, operating systems, and running/installed applications
    - Network data: IP address, Media Access Control (MAC) address, link speed, ping times, and trace routes
    - Environmental context: any information related to the physical environment, such as temperature, weather, lighting, loudness, or humidity
- Indirect Contextual Information: This can be indirectly achieved by performing operations on or modifying the contextual values: for example, calculating power consumption using voltage and current values or speed from multiple GPS locations. This classification is mentioned in [43] with the term 'secondary contextual information', which refers to any contextual information resulting from performing any operation on existing contextual information.
- Other classifications: Contextual information can also be classified based on the status of the system and its characteristics, as well as the resource of the involved contextual information attributes [44]. Such classifications could include:
    - Static contextual information: contextual information that changes very slowly or does not change at all, and includes the address and name of the user
    - Dynamic contextual information: contextual information that changes over time, such as time and location of the user
    - Internal contextual information: contextual information that is retrieved from the used devices by the user, to access services or other devices, including battery level, and current and voltage readings
    - External contextual information: contextual information that can be retrieved from external resources such as location of the user, as retrieved by the GSM operator

3.1.2. Contextual Information Gathering

Table 2 shows what contextual information would be available and how it would be collected. The vast majority of these contextual attributes could be used when the user is accessing the system remotely, with some environmental information able to be collected with trusted sensors installed at a remote location including, for example, by using a Bluetooth module to check if the user's device is nearby. Such contexts that would require these trusted sensors at remote locations are listed as 'possible'. Some of the contextual information can be collected and maintained solely by the Gateway and normal responses from user devices, while others would require extra data from the user's device itself. Lastly, while most of this data can be collected in the background, some other contextual information, such as the user's profile information, security questions, calendar, and password, would require specific interaction with the system.

3.1.3. Quality of Context Information

Another important factor regarding retrieved contextual information that is related to security is Quality of the Contextual Information (QoC). The quality is based on the accuracy and trust of the received values. While accuracy is often desirable in applications such as GPS location tracking, [45] mentions that the QoC is only related to the information itself, and does not involve the resource that provides the contextual information or the performed processes. However, we argue that it is also important to confirm the legitimacy of the received information. Consequently, the received data need to be verified as correct and legitimate, as coming from the user or device, and not having been altered during collection or transmission. Thus, it is important to consider QoC regarding the accuracy and trust of the received values. Many different methods [45–47] can be adopted for measuring the quality of contextual information. The popular approaches include:

- Statistical analysis, which is based on mathematical models to exclude unreliable values: for example, retrieving the temperature and pressure values from different resources and discarding any anomalous values. Statistical analysis is an effective approach that can be applied especially to environmental contextual information, and also when having multiple resources of the received contextual information.
- Confidence value, which is based on assigning different confidence values for the involved authentication tools and the used devices. As an example, physical tokens have higher confidence values than traditional credentials such as username and password. In our work, higher confidence values will be assigned to the retrieved contextual information based on the trust of the related resource. For instance, by considering MAC address spoofing, a MAC address will be assigned a lower confidence value than voltage value. In addition, the confidence level will be assigned based on information resulting from historical analysis as long-term contextual information.

Table 2. Contextual information to be used in the proposed framework.

Context Type	Feature	Data Collected by	Available Remotely	Requires App or External API	Requires User Intervention
User Context	Location (GPS)	Device	Yes	No	No
	Access patterns (logs)	Gateway	Yes	No	No
	Profile	Device	Yes	No	Yes
	Calendar	Device	Yes	Yes	Yes
Device Context	Location (Bluetooth)	Gateway	Possible	No	No
	Operating System	Gateway	Yes	No	No
	Browser	Gateway	Yes	No	No
	Voltage value	Device	Yes	Yes	No
	Wi-Fi access points	Device	Yes	Yes	No
	Used applications	Device	Yes	Yes	No
	Battery level	Device	Yes	Yes	No
	MAC address	Gateway	Yes	No	No
	Motion detection	Device	Yes	Yes	No
	Rotation detection	Device	Yes	Yes	No
	Compass (environment detection)	Device	Yes	Yes	No
Network Context	IP address	Gateway	Yes	No	No
	Connection type	Device	Yes	Yes	No
	Ping	Gateway	Yes	No	No
	Speed	Device	Yes	Yes	No
	Trace route	Gateway	Yes	No	No
Environmental Context	Lighting	Device	Possible	Yes	No
	Temperature	Both	Possible	Yes	No
	Pressure	Both	Possible	Yes	No
	Humidity	Both	Possible	Yes	No
	Loudness	Device	Possible	Yes	No

We assume that the achieved contextual information is collected from secured resources and examined in real-time. Taking into consideration that authentication can be strengthened by multi-factor authentication attributes (a combination of static credentials and contextual information), this work includes many authentication attributes for decision making, with an assigned confidence level for each. The confidence level with an access threshold is assigned to users in an easy manner, giving homeowners the choice of including the available factors according to their situation.

In the designed framework, different contextual attributes and properties can be utilized to enhance the determination of a legitimate user. This can be achieved by capturing long-term contextual information (e.g., power consumption, access patterns); short-term contextual information (e.g., location, calendar, profile); sensor data (e.g., pressure, temperature); and dependent data (e.g., power consumption, speed). Additionally, it is important that the collected contextual information is transmitted to and stored securely on the server (Home Gateway), and has backups in case of data loss, so models and profiles do not need to be retrained. Historical information and resources of this information will be considered as factors for measuring the quality of contextual information. Long-term contextual information, sensor data, and dependent data will be utilized for future work using machine learning techniques. The contextual information that is applied in this work includes user profile (e.g., name, ID, age, and other information), location (IP and Bluetooth), user's calendar, and historical information (access patterns and logs). Based on the categorization of IoT applications as presented in [17], the implementation and evaluation of the proposed framework is based on the first category (smart home scenario), as Figure 1 shows. Regarding data collection, various scenarios are possible in an effort to achieve contextual information that is related to the device, environment, user, and the network.

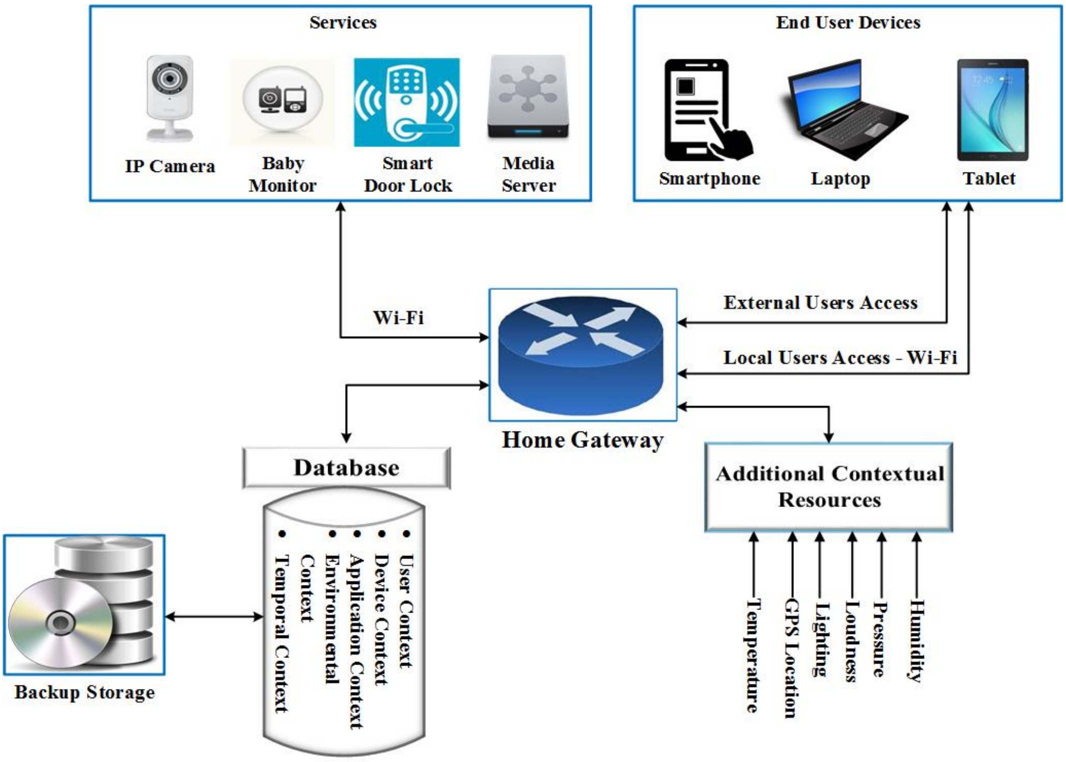


Figure 1. A high-level architecture of the proposed framework.

3.2. Framework Components

This section defines some of the terminology used in the rest of this paper as well, as the major components upon which the framework is built.

(1) A Home Device: A home device might be any device, including an IP camera, thermostat, or smart lock, which can be accessed wirelessly by users through the Gateway. Access request will be

granted if the user is either in the boundary area of the physical home space or via the Internet. This access will depend on a combination of traditional authentication with context-aware attributes such as location, schedule, and time.

(2) End-User's Devices: An end-user's device is any smart device (e.g., smartphone or tablet) that can be used to access protected devices or services that are resident on that device, through an application programming interface (API) installed during the registration stage. This API can collect sensor data and send it to the home server for user behavior analysis. Every access request or command will be accompanied with sensor data measurements for user pattern analysis and evaluation.

(3) Home Gateway (Local Server): The Gateway acts as an intermediary between the user and the connected home devices. It is responsible for the authentication process, and protects access to these devices. The Gateway, which collects the required context information and determines whether the access request satisfies the predefined requirement, is also responsible for verifying identity information that is received or transferred for authentication purposes.

  - Bluetooth Sensor: The embedded Bluetooth sensor in the Raspberry Pi is used for the Gateway and utilized to collect information about the user's location in real-time.
  - The Gateway Database: This contains all the following tables.
    - Access Control Policies: This contains the assigned policies and roles that dictate a user's privileges and the security levels of the various smart devices.
    - Login Access Profile: The system saves a copy of all authentication attempts for each user with related information, such as time, session duration, the number of attempts, access decisions, and any other provided contextual information that is related to every authentication or access attempt. These data will help in interpreting the system's usage, learning usage behavior, and checking for malicious usage.
    - General User Information: This contains the information that describes each user's personal information, such as name, age, and any other identifiers used in authentication.
    - Calendar: This caches the events retrieved from users' calendars that are used in determining access schedules, and periodically compares them against the online copies to check for new events or tampering.

3.3. Framework Features

The following characteristics and features are considered as fundamental properties of the system by default:

- The user does not always have to provide credentials to be continuously authenticated beyond the point-of-entry, unless a specific situation requires it.
- Users are not required to set up any security configuration, but are required to provide some related information and preferences that will be enhanced with contextual information collected by the system itself. The homeowner configures the user and group policies, which can be easily applied.
- Any undesired offensive event, such as using another user's credentials, will not permit access to services with high-security levels.
- The devices need to be protected in two scenarios:
  - From local users with access to the wireless network who are not permitted to access the home devices
  - From remote access by attackers who try to access home devices.

- Users are provided access privileges that expire without the need for manual revocation in the case of not manually terminating the access session. For example, re-checking the utilized contextual information, such as calendars and locations, should be performed at particular time intervals. When there is some variation from predefined policies, the access session will be automatically revoked.
- The re-check interval is set to one minute as an initial value, and this time is then updated based on the average of the previous access sessions of the user. For example, if the average of the previous access sessions is 10 minutes, then the frequency of re-checking the context information would be one-third of this time. When any of the authentication requirements do not meet predefined roles (the predefined confidence levels, as shown in Table 6, are the minimum threshold for accessing the required device), the access session will be automatically revoked. Access to some devices is restricted to home (local area) boundaries, using short-range wireless communication methods to prevent access from outside the home environment via the used wireless network. Another attribute used for ensuring the correct connected device is by comparing some of the device sensor readings, such as temperature, with locally established sensor readings.
- Once the authentication process is complete, the device that can be accessed will appear in the user's access page devices list. Some devices can be accessed anonymously or without additional authentication, while others will be greyed-out or hidden entirely if blocked.
- The access control policies will be assigned based on the resulting authentication average weight. All authentication logs are registered in the database in order to be used for future logging decisions and analysis.

3.4. Use Case Scenario

In this section, we present a brief use case reflecting the typical workflow of a new user.

- Registration stage: The user will provide some details, including preferences and calendar schedule, and choose a basic credential username and password. Considering that predefined security questions are just like other passwords, our framework requires the user to provide some preferences that are not shared with relatives, and are not available on social media websites. These preferences, in the form of questions, will be used in a situation where the minimum confidence level has not been achieved: for example, when the system is not able to retrieve all the necessary contextual information.
- Verification stage: After completion of the registration stage, the homeowner reviews the user registration and activates the account.
- Login stage: For the first login process, the user inputs credentials to obtain access to the required service. In this access, the system achieves contextual information that is related to the user, which will be later used with the predefined information at the registration time for calculating the confidence level for future access requests. For obtaining the highest level of authentication for subsequent access requests, the system associates the device that the users are using with their profile and available contextual information in the database. The Gateway will verify the context data related to the user, such as location, calendar, preferences, and log history. Based on the result, the user will or will not be granted access. The user's level of access is determined by calculating the combined confidence level based on the available contextual information. The weights of these contexts can be easily set by the homeowner based on their preferred view of priority regarding the availability of the contextual information. For example, if a user does not have a calendar, the homeowner would replace the weights according to the available contextual attributes.
- In the case of access from outside the home environment to a specific device, access will be restricted based on the predefined policies and roles by the homeowner.
- Usage stage: The user now has access to smart devices through the web GUI, with the Gateway continually confirming access using the other contextual information and history logs.

4. Implementation

The proposed architecture was implemented at the Devices, Networks and Architecture (DNA) Lab, as Figure 2 shows [37,38]. This section describes the contextual information retrieved and the authentication process based on the collected contextual information.

Contextual Information Retrieval

Some of the contextual information can be collected and maintained solely by the Gateway which, for our implementation, was a software running on a Raspberry Pi, and normal responses from the user devices, while others would require extra data from the user's device itself. This is achieved through the use of an application installed on the end-user's device as a background service, such as an Android app on the Google Nexus tablet or Samsung Galaxy Note 4 phone used in the DNA lab. Lastly, while most of this data can be collected in the background, some contextual information, such as the user's extra profile information or calendar, as well as static credentials, such as a password, would require the user to specifically interact with the system. With all of this information, we can assess which contextual information is easiest to collect and verify, and would have the least impact on users and their operation of the system and smart home devices.

These data are used to build a user profile that is then used to make access decisions for the user and demand additional verification, if required for specific actions. For the sake of privacy and because of susceptibility to loss or theft, no information related to users is stored on the devices. This ensures that if the device is compromised, the adversary cannot learn the user profile and simulate the user's behavior to gain system access. All data is verified and replicated to a backup location to avoid both data loss and the need to retrain user profiles, thus preventing a malicious user from rebuilding a user's profile to match their own contexts.

The prototype implementation utilizes a Linksys E1200 [48] router flashed with DD-WRT [49]. The application then runs on a Raspberry Pi, which can control the router using the SSH functionality provided by the DD-WRT firmware. The router uses two wireless networks: one for users and one for IoT devices, to allow for access control between the two networks. The firewall blocks all access to the IoT network by default, allowing only access to the Pi hosting the Secure Gateway Application, which controls further access. A simple port forward on the router to the Pi would allow for external access to the application and remote control of devices, if permitted.

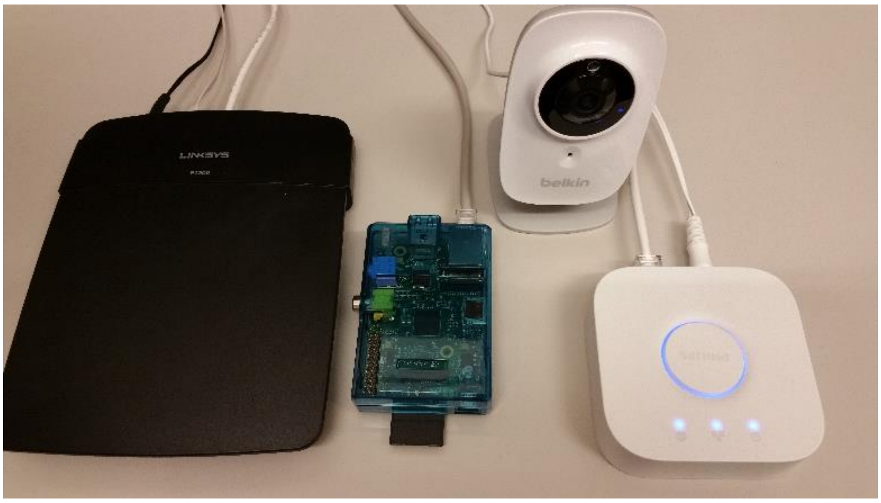


Figure 2. Used devices in the implemented prototype.

The application is a Python program which runs a Flask [50] web server and Paramiko [51] SSH session along with a MySQL database. The Flask server handles user authentication and provides the controls necessary to interact with the IoT devices on the network, along with the homeowner's tools to add or configure new users or devices. The Paramiko library establishes an SSH connection to the router which allows the application to both search for connected devices and control their interactions. Finally, the MySQL database stores users and devices along with their individual configurations, rule sets, and logs. Table 3 shows the detailed specifications of the used devices for the implementation.

To access any of the IoT devices on the network, such as the smart bulbs or switches, the user would first browse to the page where the application is hosted on the Pi. This would either be a statically assigned IP address or hostname that could be easily bookmarked by frequent users or shared with guests. The user would be greeted with a homepage that would allow the control of less sensitive devices and hide or deny control of more sensitive devices. When attempting to control a device that can be seen but not accessed, the user is prompted to either login or provide additional authentication parameters to increase the security level. The user's security level, a combination of the device address, location, login, and time, determines whether a device can be accessed.

New users are able to sign up on the site in order to gain access to restricted devices. By default, a new user is unverified and unable to log in until the homeowner confirms their account. The main user (the homeowner) could also then configure more specific permissions at device-level, permitting or denying access to particular devices based on the user's needs or the homeowner's trust of said user. Each user would also be able to have an associated calendar to determine when access to devices should be allowed. For this prototype, integration with a user's Google Calendar was used, along with a cached copy in the database. This allows both users to determine times when they will be away from home and therefore not accessing certain devices, as well as enables the homeowner to determine times to explicitly permit or deny access to certain devices by a user.

Table 3. Specifications of devices used in the implementation.

Brand	Device Type	Device Name	Network Interfaces	Revision	Power
Linksys	Wireless Router	N300 Wi-Fi Router	4x 10/100 Ethernet and 802.11n Wi-Fi	E1200-V2	AC to DC
Raspberry Pi Foundation	Single-Board Computer	Raspberry Pi 3	10/100 Ethernet, 802.11n Wi-Fi and Bluetooth 4.1	Model B	USB to DC
Belkin	Smart Switch	WeMo Insight Switch	802.11n Wi-Fi	F7C029V2	AC
Philips	Smart Bulb	Hue White A19	ZigBee	9290011369	AC / Edison Socket
Belkin	Camera	NetCam HD+	802.11b/g/n Wi-Fi	F7D7606v1	DC

New devices can be added to the system by connecting them to the wireless network intended for the device, and then going to the respective page. Devices are listed on the page based on the router's ARP table, acquired through SSH, and cross-referenced against the database to determine if they have already been added and configured. In addition to being able to set their name, description, and static IP, used devices can be set to only allow access from certain user accounts or be set to ignore location authentication if they are in a fixed location. IoT devices can also be configured to only permit access to certain users, along with many options, such as allowing external access, anonymous access, requiring location authentication, or restricting usage to certain devices or schedules.

All actions by users are logged in the database against the device origin and the user credentials if logged in. This includes actions such as arriving or departing a location, accessing the site or devices, and logging in or out of the site. This logging allows for heuristic analysis and pattern matching, which could potentially be used in the future as another level of authentication. A user profile could be established based on their access patterns and a notification could be sent to the homeowner upon a given variance, or access could be pre-emptively denied.

Finally, authentication is based on overall quality of the provided credentials. Users are granted access to different levels of services dependent on this overall quality. This quality is calculated by the homeowner during the setup, with assigned weights to each included authentication, for example:

location weight = 20%, calendar schedule data weight = 20%, time = 10%, username and password = 30%, and profile data (such as preferences) = 20%. Smart devices are setup with a required confidence level needed to access their services. If in the event that confidence level is too low to access a service, the user will be requested to enter traditional credentials, such as security questions/preferences, or try again when scheduled or no longer attempting to access remotely, depending on what attributes are lacking.

5. Evaluation Results

In evaluating the effectiveness of the implemented prototype, we have used the following criteria:

- The overhead (time/ms) imposed on the system by each added attribute used in the authentication process
- The authentication-assigned weights and thresholds set by the homeowner and their effects on access decision-making
- The ability for the server (Home Gateway) to handle multiple simultaneous requests without bottlenecking access to smart devices

5.1. Evaluation 1: Performance

This part of the evaluation examines the overhead (time/ms) imposed on the system by each added attribute used in the authentication processes shown in Tables 4 and 5.

Table 4. Performance of individual authentication methods.

Used Credentials	Local Access Time (ms)	Internet Access Time (ms)
No authentication	7	90
IP address-based location (network)	8	90
Bluetooth-based location (proximity)	14	97
Static credentials (username, password)	15	96
Calendar access	13	96

Table 5. Performance of authentication methods combined.

Used Credentials	Local Access Time (ms)	Internet Access Time (ms)
No authentication	7	90
Location based on both IP address and Bluetooth	14	90
Location based on both IP address and Bluetooth, and static credentials (username, password)	16	98
Location based on both IP address and Bluetooth, static credentials (username, password), and calendar access	20	98

The above tables demonstrate our performance tests on all the authentication methods, both individually selected contextual attributes and combined attributes. Both the location based on the IP address and the user's session cookie, granted by user credentials, are retrieved from the user's request itself. Nearby Bluetooth devices are retrieved from the cache, and the current time is compared against events in the user's calendar in the database. As expected, no authentication yields the fastest results, with a combination of all methods being the slowest. However, as can be seen in Tables 4 and 5, there is very little overhead on the request-level associated with the different authentication methods. While location, calendar access, and even static credentials most affect the response times, the difference is almost negligible, especially when accessing the system over the Internet.

5.2. Evaluation 2: Authentication Weights and Device Thresholds

For the second experiment, the system framework is evaluated with regards to the calculated confidence levels, based on the assigned weights and the subsequent access levels given to the various services. Table 6 shows some example weights for contexts that are assigned by default, and security levels that are assigned by the homeowner which will be applied to smart devices of varying concern. The confidence level is considered as the total of the assigned weight of the available context information.

As seen in Table 6A, when providing both credentials of username and password, the assigned weight is 40 whereas, when achieving access to only the calendar, the assigned weight is only 10. In contrast, as seen in Table 6B, the highest security level is 4 with threshold 100, whereas the lowest security level is 1 with threshold 30.

Table 6. Example of authentication weights for context and threshold for accessing devices.

(A)		(B)	
Available Parameters	**Assigned Weight/100**	**User Security Level**	**Access Threshold/100**
Username & Password	40	4	100
Location (proximity)	30	3	70
Location (network)	20	2	50
Calendar	10	1	30

Table 7 reflects some examples of calculating a confidence level by adding together the assigned weights and determining if the system should grant the user access to the requested service. As can be seen from the same table, should the confidence level (calculated by combining the available parameter weights) be sufficient to meet the security level/access threshold of the requested service, access is granted. If the calculated confidence level is insufficient to meet the security level/access threshold of the requested service, access will be denied.

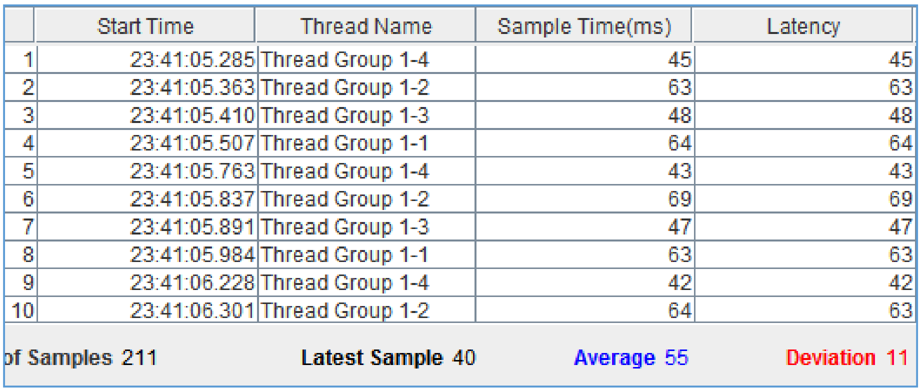
As an example, as seen in Table 7, with the ability to access calendar information as well as the location based on the network, the total calculated weight will be 30. Thus, if the user is requesting access to a service with an assigned security level of 3 or higher, the request will be denied, since the achieved security level is lower than that required. In addition, having only a username and password would allow login and access to only those services that do not require a high threshold. As an example, weight 40 will not allow access to important services by setting a higher access threshold.

Table 7. Calculated confidence levels and authentication scenarios.

Available Parameters	Weights	Confidence Level	User Security Level	Service Security Level	Access Decision
Username, password, and Bluetooth	40, 30	70	3	2	Granted
Bluetooth and on local network	30, 20	50	2	2	Granted
Scheduled and on local network	10, 20	30	1	3	Denied
Username, Password	40	40 (<50)	1	4	Denied
Bluetooth	20	20	1	2	Denied
Scheduled, Bluetooth and on local network	10, 30, 20	60 (<70)	2	2	Granted
Scheduled, Bluetooth and on local network	10, 30, 20	60 (<70)	2	1	Granted

5.3. Evaluation 3: Scalability

In this section, the Gateway is tested for its ability to handle multiple simultaneous requests without bottlenecking access to smart devices. This evaluation demonstrates that the implemented framework is able to handle several simultaneous requests simulated by Apache JMeter, without significantly affecting response times. As shown in Figure 3, the implemented framework is able to handle several simultaneous requests without significantly affecting the response times from our previous trials.

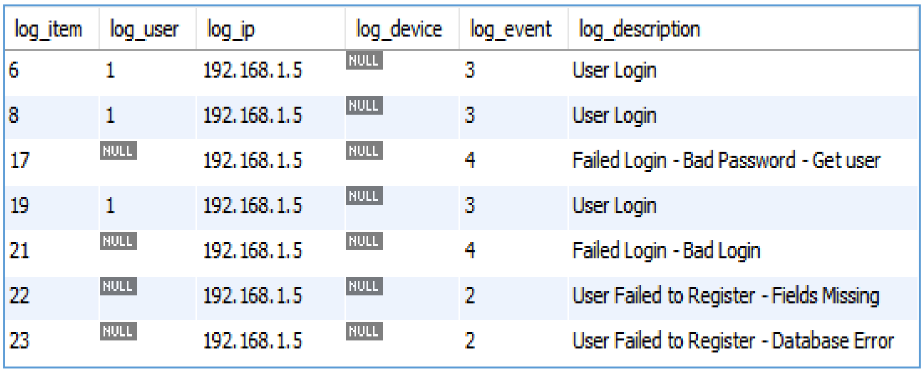


	Start Time	Thread Name	Sample Time(ms)	Latency
1	23:41:05.285	Thread Group 1-4	45	45
2	23:41:05.363	Thread Group 1-2	63	63
3	23:41:05.410	Thread Group 1-3	48	48
4	23:41:05.507	Thread Group 1-1	64	64
5	23:41:05.763	Thread Group 1-4	43	43
6	23:41:05.837	Thread Group 1-2	69	69
7	23:41:05.891	Thread Group 1-3	47	47
8	23:41:05.984	Thread Group 1-1	63	63
9	23:41:06.228	Thread Group 1-4	42	42
10	23:41:06.301	Thread Group 1-2	64	63

of Samples 211 Latest Sample 40 Average 55 Deviation 11

Figure 3. Snapshot of Apache JMeter multiple simultaneous requests.

As shown in Figure 4, the implemented framework stores all historical information that is related to users interacting in any way with the system or various services. This includes, but is not limited to, user logins (both successful and failed), accessed services (both permitted and denied), and new registrations.



log_item	log_user	log_ip	log_device	log_event	log_description
6	1	192.168.1.5	NULL	3	User Login
8	1	192.168.1.5	NULL	3	User Login
17	NULL	192.168.1.5	NULL	4	Failed Login - Bad Password - Get user
19	1	192.168.1.5	NULL	3	User Login
21	NULL	192.168.1.5	NULL	4	Failed Login - Bad Login
22	NULL	192.168.1.5	NULL	2	User Failed to Register - Fields Missing
23	NULL	192.168.1.5	NULL	2	User Failed to Register - Database Error

Figure 4. Historical data captured by the Home Gateway.

6. Security Analysis

Security analysis of possible attacks on the system are discussed here in order to validate the ability of the framework to protect against them.

6.1. Obtaining a User's Login and Password

In the event that a victim user's login and password were stolen, compromised, or even loaned out, the other security contexts would still be able to react by denying the new user access from another device to the network and home device through device identification or scheduling. In addition, the user's login and password are protected in transit through the use of HTTPS encryption and in the database, using a salted hash function.

6.2. Obtaining a User's Device

This type of attack is mitigated by considering the device's location through the IP address and proximity detection. If the device were stolen and not yet reported to the homeowner, its absence from the home network or known remote locations would be sufficient to deny access to a malicious user. In addition, in case of trying to access from a registered device using the achieved compromised credentials, weight 40 will not allow access to important services by setting a higher access threshold.

Thus, having only a username and password will not allow an illegitimate user to access important services. For protecting access from within the home network, this will be mitigated through addressing pattern analysis in future work.

6.3. Brute Force or Guessing Attacks

This type of attack is countered by monitoring and recording both login attempts and service access attempts, locking out a user's account or device until reviewed by the homeowner.

6.4. Unauthorized Modification of Contextual Information

This type of attack is countered by faking the framework through using as much data as possible to authorize clients.. When it comes to the modification of external data such as the schedule, a cached copy of the calendar is kept on the Gateway. In addition, any update in the schedule will be flagged and reported to the homeowner, and permission from the homeowner is required to update the cached copy on the Gateway.

Moreover, the assigned weight values can be set by the homeowner based on the availability of contextual information. For example, if a user does not have a calendar, the homeowner would set the weights based on other available contextual attributes. Furthermore, the time it would take to detect the availability of intrusion is the same as the re-check time. In addition, this time is determined based on the average of the previous access sessions of the user. Hence, we believe that this time is short enough to realize any unauthorized access. The weighting of the various contexts also prevents any single data source being compromised from significantly affecting the security of the system as a whole, with the calendar scheduling not explicitly permitting any user access as a prime example.

6.5. IP/MAC Address Spoofing and Data Protection

The efficacy of MAC address spoofing can be somewhat reduced through the monitoring of network traffic and clients connected to the router. An IP address conflicts or is malformed, or any non-IEEE compliant MAC addresses could, could be other indicators of possible spoofing attempts. Hence, the use of certificate-based authentication would prevent the possibility of the Gateway being impersonated by an attacker.

For the sake of privacy, no information related to users is stored on the end devices as they are susceptible to loss or theft. This ensures that if the device is compromised, the adversary cannot learn the user profile and simulate the user's behavior to gain system access. Furthermore, all data is verified and replicated to a backup location to avoid both data loss and the need to retrain user profiles, thus preventing a malicious user from rebuilding a user profile to match their own contexts.

7. Conclusions and Future Work

Ensuring the authenticity of a user beyond the point-of-entry to access a service has become a crucial issue, hence, the need for continuous authentication. This paper introduces a continuous authentication framework that utilizes contextual information. This framework has the ability to protect home devices against unauthorized access from anonymous and known users, whether locally or remotely, by routing all communication to said devices through the secure home Gateway. The users, devices and policies can all be configured to create a system which authenticates users and grants them access to services based solely on contextual information, without necessarily needing to provide any credentials for continuously authenticating users, unless a specific situation requires this. It also monitors all access-related activities, such as attempted logins, service requests, and access durations, storing them in a database for future analysis by establishing usage patterns and detecting brute force attempts and other anomalies. The implementation and evaluation show that the proposed framework provides continuous authentication in a flexible manner, without requiring additional intervention by users during the access session. It is also concluded that the security measures do not impose significant connection overhead, which amounts to the system acting almost entirely transparently.

Ultimately, results show that considerable contextual information can be retrieved in a reasonable time, and that such contextual information can be used in providing a seamless, usable, and secure authentication for the IoT. Finally, including contextual information in the authentication process should increase the level of security beyond point-of-entry by providing non-intrusive, convenient, and continuous authentication throughout the duration of the access session. This is possible by enhancing the standard knowledge-based static authentication with contextual information.

For future work, measuring the quality of the retrieved contextual information will be considered according to the techniques discussed. In addition, we plan to explore the possibility of further analyzing the historical data (long-term contextual information) retrieved from end-user devices and stored in the Gateway to generate higher fidelity user profiles using machine learning and other techniques. Being able to accurately identify a user based on usage patterns with the use of contextual information would greatly improve the system's ability to counter various attack scenarios, such as insider unauthorized access, as well as reduce the number of false positives and requests for user intervention.

Author Contributions: Writing-Original Draft Preparation, Y.A.; Writing-Review & Editing, D.K.; Supervision and Writing-Review & Editing, Q.H.M.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ashibani, Y.; Mahmoud, Q.H. Cyber Physical Systems Security: Analysis, Challenges and Solutions. *J. Comput. Secur. Elsevier* **2017**, *68*, 81–97. [CrossRef]
2. Ashibani, Y.; Mahmoud, Q.H. An Efficient and Secure Scheme for Smart Some Sommunication Using Identity-Based Signcryption. In Proceedings of the IEEE 36th International Performance Computing and Communications Conference, IPCCC, San Diego, CA, USA, 10–12 December 2017; pp. 1–7. [CrossRef]
3. Jeong, J.; Chung, M.Y.; Choo, H. Secure User Authentication Mechanism in Digital Home Network Environments. In *Embedded and Ubiquitous Computing. EUC 2006*; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Berlin/Heidelberg, Germany, 2006; Volume 4096, pp. 345–354. [CrossRef]
4. Forget, A. A World with Many Authentication Schemes. Ph.D. Thesis, Carleton University, Ottawa, ON, Canada, 2012.
5. Li, Y.; Wang, H.; Sun, K. Personal Information in Passwords and Its Security Implications. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2320–2333. [CrossRef]
6. Dropbox Hackers Stole 68 Million Passwords. Available online: http://www.telegraph.co.uk/technology/2016/08/31/dropbox-hackers-stole-70-million-passwords-and-email-addresses/ (accessed on 8 October 2018).
7. Covington, M.J.; Fogla, P.; Zhan, Z.; Ahamad, M. A Context-Aware Security Architecture for Emerging Applications. In Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC '02), Las Vegas, NV, USA, 9–13 December 2002; pp. 249–258.
8. Covington, M.J.; Sastry, M.R.; Manohar, D.J. Attribute-Based Authentication Model for Dynamic Mobile Environments. In *Security in Pervasive Computing. SPC 2006*; Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Berlin/Heidelberg, Germany, 2006; Volume 3934, pp. 227–242. [CrossRef]
9. Benzekki, K.; El Fergougui, A.; Elalaoui, A.E.B. A Context-Aware Authentication System for Mobile Cloud Computing. *Procedia Comput. Sci.* **2018**, *127*, 379–387. [CrossRef]
10. Kim, S.H.; Choi, D.; Kim, S.H.; Cho, S.; Lim, K.S. Context-Aware Multimodal FIDO Authenticator for Sustainable IT Services. *Sustainability* **2018**, *10*, 1656. [CrossRef]
11. Ehatisham-ul-Haq, M.; Awais Azam, M.; Naeem, U.; Amin, Y.; Loo, J. Continuous Authentication of Smartphone Users Based on Activity Pattern Recognition Using Passive Mobile Sensing. *J. Netw. Comput. Appl.* **2018**, *109*, 24–35. [CrossRef]

12. Lee, W.; Lee, R. Implicit Sensor-Based Authentication of Smartphone Users with Smartwatch. In Proceedings of the Hardware and Architectural Support for Security and Privacy, Seoul, Korea, 18 June 2016; ACM: New York, NY, USA, 2016; Volume 9. [CrossRef]
13. Fuentes, D.; Maria, J.; Gonzalez-Manzano, L.; Ribagorda, A. Secure and Usable User-in-a-Context Continuous Authentication in Smartphones Leveraging Non-Assisted Sensors. *Sensors* **2018**, *18*, 1219. [CrossRef] [PubMed]
14. Saevanee, H.; Clarke, N.L.; Furnell, S.M. Multi-Modal Behavioural Biometric Authentication for Mobile Devices. *IFIP Adv. Inf. Commun. Technol.* **2012**, 465–474. [CrossRef]
15. Ashibani, Y.; Mahmoud, Q.H. A Behavior Profiling Model for User Authentication in IoT Networks Based on App Usage Patterns. In Proceedings of the 44th Annual Conference of the IEEE Industrial Electronics Society (IECON), Washington, DC, USA, 21–23 October 2018; pp. 2841–2846.
16. Ashibani, Y.; Mahmoud, Q.H. A User Authentication Model for IoT Networks Based on App Traffic Patterns. In Proceedings of the 9th Annual IEEE Information Technology; Electronics and Mobile Communication Conference (IEEE IEMCON), Vancouver, BC, Canada, 1–3 November 2018; pp. 1589–1595.
17. Mahalle, P.N.; Anggorojati, B.; Prasad, N.R.P.; Prasad, R. Identity Authentication and Capability Based Access Control (IACAC) for the Internet of Things. *J. Cyber Secur. Mobil.* **2013**, *1*, 309–348.
18. Yaici, M.; Oussayah, A.; Takerrabet, M.A. Trust-Based Context-Aware Authentication System for Ubiquitous Systems. *Procedia Comput. Sci.* **2018**, *134*, 35–42. [CrossRef]
19. Chitalia, U.; Sanghavi, M.; Iyer, S.; Shah, S.; Jyotinagar, V. Single Sign On (SSO) Application for Websites. *Int. J. Adv. Eng. Sci. Technol.* **2014**, *2*, 207–212.
20. Al Abdulwahid, A.; Clarke, N.; Stengel, I.; Furnell, S.; Reich, C. Continuous and Transparent Multimodal Authentication: Reviewing the State of the Art. *Clust. Comput.* **2016**, *19*, 455–474. [CrossRef]
21. Aloul, F.; Zahidi, S.; El-Hajj, W. Two Factor Authentication Using Mobile Phones. In Proceedings of the IEEE/ACS International Conference on Computer Systems and Applications, AICCSA 2009, Rabat, Morocco, 10–13 May 2009; pp. 641–644. [CrossRef]
22. Mock, K.; Weaver, J.; Milton, M. Poster: Real-Time Continuous Iris Recognition for Authentication Using an Eye Tracker. *CCS* **2012**, 1007–1009. [CrossRef]
23. Tsai, P.W.; Khan, M.K.; Pan, J.S.; Liao, B.Y. Interactive Artificial Bee Colony Supported Passive Continuous Authentication System. *IEEE Syst. J.* **2014**, *8*, 395–405. [CrossRef]
24. Miettinen, M.; Nguyen, T.D.; Sadeghi, A.-R.; Asokan, N. Revisiting Context-Based Authentication in IoT. In Proceedings of the 55th ACM/ESDA/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 24–28 June 2018; pp. 1–6. [CrossRef]
25. Zhou, K.; Ren, J. PassBio: Privacy-Preserving User-Centric Biometric Authentication. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 3050–3063. [CrossRef]
26. Belguechi, R.; Alimi, V.; Cherrier, E.; Lacharme, P.; Rosenberger, C. An Overview on Privacy Preserving Biometrics. In *Recent Application in Biometrics*; InTech: London, UK, 2011; pp. 65–84.
27. Karim, N.A.; Shukur, Z. Review of User Authentication Methods in Online Examination. *Asian J. Inf. Technol.* **2015**, *14*, 166–175. [CrossRef]
28. Chaim Gartenberg. Hacker Beats Galaxy S8 Iris Scanner. Available online: https://media.ccc.de/v/biometrie-s8-iris-en#video&t=21 (accessed on 13 July 2018).
29. Li, F.; Clarke, N.; Papadaki, M.; Dowland, P. Behaviour Profiling for Transparent Authentication for Mobile Devices. In *Proceedings of the European Conference on Information Warfare and Security, Tallinn, Estonia*; Academic Conferences International Limited: Sonning Common, UK, 2011; pp. 307–315.
30. Shi, E.; Niu, Y.; Jakobsson, M.; Chow, R. Implicit Authentication through Learning User Behavior. In Proceedings of the Conference on Information Security, Boca Raton, FL, USA, 25–28 October 2010; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6531, pp. 99–113. [CrossRef]
31. Damopoulos, D.; Menesidou, S.A.; Kambourakis, G.; Papadaki, M.; Clarke, N.; Gritzalis, S. Evaluation of Anomaly-Based IDS for Mobile Devices Using Machine Learning Classifiers. *Secur. Commun. Netw.* **2012**, *5*, 3–14. [CrossRef]
32. Niinuma, K.; Park, U.; Jain, A.K. Soft Biometric Traits for Continuous User Authentication. *IEEE Trans Inf. Forensics Secur.* **2010**, *5*, 771–780. [CrossRef]
33. Agudo, I.; Rios, R.; Lopez, J. A Privacy-Aware Continuous Authentication Scheme for Proximity-Based Access Control. *Comput. Secur.* **2013**, *39 Pt B*, 117–126. [CrossRef]

34. Corner, M.D.; Noble, B.D. Protecting Applications with Transient Authentication. In Proceedings of the 1st International Conference on Mobile Systems, Applications and Services, San Francisco, CA, USA, 5–8 May 2003; pp. 57–70. [CrossRef]
35. Mhamed, A.; Abi-char, P.E.; Mokhtari, B.E.M. A Dynamic Trust-Based Context-Aware Authentication Framework with Privacy Preserving. *Int. J. Comput. Netw. Secur.* **2010**, *2*, 87–102.
36. Santoso, F.K.; Vun, N.C.H. Securing IoT for Smart Home System. *Proc. Int. Symp. Consum. Electron. ISCE* **2015**, 5–6. [CrossRef]
37. Ashibani, Y.; Kauling, D.; Mahmoud, Q.H. Poster: A Context-Aware Authentication Service for Smart Homes. In Proceedings of the 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 8–11 January 2017; pp. 587–588. [CrossRef]
38. Ashibani, Y.; Kauling, D.; Mahmoud, Q.H. A Context-Aware Authentication Framework for Smart Homes. In Proceedings of the IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Windsor, ON, Canada, 30 April–3 May 2017; pp. 1–5. [CrossRef]
39. Trnka, M.; Cerny, T.; Stickney, N. Survey of Authentication and Authorization for the Internet of Things. *Secur. Commun. Netw.* **2018**, 1–17. [CrossRef]
40. Qin, W.; Zhang, D.; Shi, Y.; Du, K. Combining User Profiles and Situation Contexts for Spontaneous Service Provision in Smart Assistive Environments. In *Ubiquitous Intelligence and Computing. UIC 2008*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2008; pp. 187–200. [CrossRef]
41. Henricksen, K. A Framework for Context-Aware Pervasive Computing Applications. Ph.D. Thesis, The School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane, Australia, 2003.
42. Schilit, B.; Adams, N.; Want, R. Context-Aware Computing Applications. In Proceedings of the First Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, USA, 8–9 December 1994; pp. 85–90. [CrossRef]
43. Perera, C.; Member, S.; Zaslavsky, A.; Christen, P. Context Aware Computing for The Internet of Things: A Survey. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 414–454. [CrossRef]
44. Nazário, D.C.; Tromel, I.V.B.; Dantas, M.A.R.; Todesco, J.L. Toward Assessing Quality of Context Parameters in a Ubiquitous Assisted Environment. *JISTEM-J. Inf. Syst. Technol. Manag.* **2014**, *11*, 569–590. [CrossRef]
45. Wrona, K.; Gomez, L. Context-Aware Security and Secure Context-Awareness in Ubiquitous Computing Environments. *Ann. UMCS Inf.* **2006**, *4*, 332–348.
46. Manzoor, A.; Truong, H.L.; Dustdar, S. On The Evaluation of Quality of Context. In Proceedings of the European Conference on Smart Sensing and Context, Zurich, Switzerland, 29–31 October 2008; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5279, pp. 140–153. [CrossRef]
47. Buchholz, T.; Küpper, A.; Schiffers, M. Quality of Context: What It Is And Why We Need It. In Proceedings of the 10th International Workshop of the HP OpenView University Association (HPOVUA), Hewlet-Packard OpenView University Association, Geneva, Switzerland, July 2003; pp. 1–14.
48. Linksys E1200 N300 Wireless Router. Available online: http://www.linksys.com/ca/p/P-E1200/ (accessed on 15 July 2018).
49. DD-WRT Firmware. Available online: http://www.dd-wrt.com/site/index (accessed on 15 July 2018).
50. Welcome | Flask (A Python Microframework). Available online: http://flask.pocoo.org/ (accessed on 15 July 2018).
51. Welcome to Paramiko!—Paramiko Documentation. Available online: http://www.paramiko.org/ (accessed on 15 July 2018).

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).