A Survey of Security Middleware for Pervasive and Ubiquitous Systems

Jameela Al-Jaroodi, Alyaziyah Al-Dhaheri, Fatmah Al-Abdouli and Nader Mohamed
College of Information Technology, UAE University, Al Ain, UAE
j.aljaroodi@uaeu.ac.ae, alyazia.s@gmail.com, 200404330@uaeu.ac.ae, nader.m@uaeu.ac.ae

Abstract - Recently, pervasive and ubiquitous computing evolved tremendously and has become an integral part of many fields. This is widely attributed to their efficiency in integrating with everyday components and handling tasks in a faster and better way. As a result, technologies to facilitate their development, integration and security also evolved. Middleware is an essential component in this field and recently more emphasis has been put on security middleware as an enabling component for pervasive and ubiquitous applications. This is due to the high levels of personal and private data sharing in these systems. In this paper we survey some representative security middleware approaches and highlight their various properties and characteristics. We also discuss and compare these approaches in terms of their design, target environments, and main features. The review reveals that several options are available; however, there are still many issues and problems to be addressed to achieve more efficient security middleware.

Keywords-Ubiquitous Computing; Pervasive Environments; Security Middleware; Authentication; Ad Hoc networks; p2p systems

I. INTRODUCTION

Middleware is a software layer residing among and connecting different software components or applications. It provides connectivity, abstractions, interoperability and other services like the distribution of functionality, scalability, load balancing and fault tolerance [1]. Moreover, it provides security mechanisms for many applications. Security became an important issue because most transactions and operations occur online and needs transmission of data. The applications and data involved need to be protected from malicious and unintentional attacks and also from any possible risks of exposure. Well defined access policies, encryption mechanisms and authentication models can help in providing security.

Pervasive computing refers to the ubiquitous presence of computing in both mobile and embedded environments, with the ability to access and update information anywhere and anytime [2]. Ubiquitous and pervasive devices are becoming very popular and less expensive these days with which different devices like mobiles, smart phones and PDA can interact easily. Pervasive computing can be used in many applications such as healthcare, home care, transportation and environmental monitoring. Therefore, pervasive devices may have implications for privacy, security and safety because of their ability to gather sensitive data, retrieve and use information from large databases. Furthermore, these devices contain limited resources thus having difficulty supporting the computational needs for security functions. The open wireless connectivity is another challenge in the pervasive computing systems putting the users and the system resources under susceptible threats. As a result, middleware could be used to provide security and protect users from privacy violation and data leakage.

In this paper, we explore different middleware projects for mobile and pervasive computing, summarize their contributions and discuss their approaches. The remainder of the paper is structured as follows. Section 2 provides a short overview of security middleware and some related work. In Section 3, we describe several research projects in middleware support for security in mobile and pervasive computing. Section 4, is a discussion of the different approaches used and Section 5 concludes the paper.

II. BACKGROUND AND RELATED WORK

Since middleware handles all the communication between a client and the target application, it can also mask all the underlying complexities including the security aspects. The security is enforced within the middleware by utilizing many security features such as authentication where identities are verified and credentials are handled, messages protected from unauthorized modifications and disclosure, access control policies are managed and audited for accountability [14]. These features are all integrated in a way that preserves the main functionalities of the middleware and fits into its structure. As a result of the security advantages it provides; security middleware has been widely adopted in many diversified systems such as e-commerce, web access grid computing and internet data storage. Taking for example iDataGuard [3] is a middleware providing a Secure Network Drive Interface to Untrusted Internet Data Storage; and FAME-PERMIS [4] an Authentication Middleware to Support Ubiquitous Web Access. Furthermore, security middleware has been a thriving research area to combat the continuous security issues in sensitive areas such as mobile ad hoc computing, pervasive/ubiquitous computing and peer-to-peer (p2p) computing. More details in the upcoming sections.

A recent research paper [5] surveys middleware paradigms for mobile computing. It presents several research projects that have been initiated to address the dynamic aspect of mobile distributed systems. The paper argues that traditional middleware approaches fail to provide the appropriate support for modern mobile applications. One of the several mentioned new technologies is the Event-Based...
Paradigm, which supports the development of large scaled distributed systems by facilitating a highly decoupled approach and many-to-many interactions. Our survey focuses on security middleware particular to pervasive and ubiquitous environments.

III. SECURITY MIDDLEWARE APPROACHES

In this section we discuss several examples of security middleware used for pervasive and ubiquitous applications. The list below is not complete, but it is a representative of the latest research directions if this field.

A. TMAHP2P

This is a middleware providing security for ad-hoc p2p applications using a trust-based approach and WSFEP (Wireless and Secure File Exchange Protocol) [6]. It is used for securing digital content that can be shared between pervasive devices and also used for providing flexible security services to the applications used in these devices.

There are many advantages associated with this middleware. This middleware performs an autonomous trust management and eliminates the complexity of establishing a new trust-based relationship with other devices. Moreover, it reduces the dependency on a central server and eliminates the need of manual settings. Therefore, any device can participate in a p2p application in a secure way.

TMAHP2P’s design is open and extensible and its architecture consists of three layers: the trust, security, and application layers. All three layers are connected through an access control based on XACML-compliant policies that will grant or deny the access to the resources. The trust layer contains the authentication manager, communication API, and secure file exchange. The application layer contains an application developed using WSFEP. The application goes through discovery and exchange phases. File exchange is done through either efficient or secure mode. In efficient mode, messages are transmitted without any security mechanism. However, in the secure mode messages are signed, encrypted, or both.

B. SGSC

Secure Group communication Service is a middleware service for mobile ad hoc networks [7]. This middleware provides flexible secure group management and supports the development and execution of distributed applications. Mobile ad hoc networks pose more requirements on secure group communication due to the limited capabilities of the members and their mobility. SGSC was implemented using a context-sensitive middleware approach and consists of three components: SGCS Daemon (to create and manage services), Group Services (created on the devices to provide core functionality) and SGSC API (used by the distributed applications to utilize group functionality). A distributed application is developed with secure group communication using RCSM middleware and members of each secure group are identified. After running the application, the secure groups will be automatically managed by the SGCS.

C. SMMU

A security management middleware designed for ubiquitous computing devices [8]. It allows the administrator to define the needed security policies and provides management services to monitor and control the interconnected devices. This middleware’s architecture is focused on providing trust management service and supporting real-time mobile application scenarios. Some of the advantages are providing authentication between connected devices, checking the permissions and monitoring the activities of the connected devices. Its architecture consists of five components which are the policy managers used to define the rules for access control, confidentiality, and availability of resources and stored information; the object manager used to monitor and control target objects (devices); the context manager used to collect the contexts and aggregate context data to be sent to the status monitoring manager; the Status monitoring manager collects the status information to send it to the database manager; and finally the authentication manager uses X.509 credential for identification.

D. SSMAP

Security-supportive middleware architecture designed to serve mainly heterogeneous pervasive devices [9]. It is provided with trust a manager that offers dynamic reconfiguration to fulfill security requirements of heterogeneous service providers and consumers. The main advantages are that its structure depends on reconfigurable and on-demand assembled security components that cooperate with each other; and trust management is linked with many services like authentication, trust negotiation, trust maintenance and access control techniques that govern the relationship between devices in pervasive computing systems. The architecture relies on a virtual server where a user can find all the provided services. The application layer lies on top and it includes both the service consumer and provider applications while, the proposed security middleware resides in the middleware layer. It provides on-demand assembled security components like encryption and intrusion detection. Trust management is dedicated to support establishing and maintaining trust between applications and different devices.

Authentication is only deployed when the user has an existing profile stored in the pervasive computing space from previous transactions to claim and prove identity. Therefore, a stranger will get lower level of access to the resources. However, access control follows the traditional mandatory access control approach (MAC) where each service is assigned with a sensitivity level that dynamically changes according to the variation of environment. To use the service, the user level has to be equal or higher than the requested service level. Trust negotiation works in parallel with authentication when a profile is not established or not available. Trust maintenance is after authentication and trust
negotiation are completed to handle monitoring and managing trust between user and service.

E. MSHAC

A middleware designed for secure home access and control for home automation [10]. It helps broaden the proliferation of ambient technology. The main obstacle with these technologies is the lack of trust and acceptance from users due to possible intrusion and unauthorized access to private data. Therefore, ‘Secure Service Middleware’ provides secure management of user and context data that gives access to services just for the authorized users and devices.

A distributed architecture has been chosen as well as the adoption of SSL/TLS based communication to guarantee security aspects. Furthermore, mutual authentication between communication partners is required with the use of a challenge-response protocol that verifies a component identity based on its public keys. The four components of the architecture are: 1) Devices and Services, where devices request services and services contact Trust Manager to check for authorization. 2) Trust Manager which is used to issue authorization with the help from Credential and Authentication manager this is done using a DSA signature scheme to prevent forgery and tampering. 3) Authentication Manager that handles the authentication of devices in order to meet the requirements of the different security levels. 4) Credential Manager to retrieve credentials in the form of access control list from the credentials database with respect to the security level for a specific user and service combination. The implementation of this middleware is based on the Profile Database Schema and JCCM (Java Card Certificate Manager). A flexible data schema has been designed that supports several data types and formats, incorporates security features (data in the database is all encrypted) and allows distribution among different databases.

F. TDAMU

A trust-based distributed authentication middleware that supports role-based trust and accountability in distributed mobile environments [11]. It uses trust-based frameworks that add a security layer for data sharing, permission management, access control and right delegation issues. The middleware has many advantages such as its support for the role-based trust in which roles can be assigned to a group of mobile objects which will ease the management of security policies. The role-based trust model consists of four objects used for authentication: Truster module which represents the party who will make a decision to whether or not to trust another (trusted). Trustee is the party who will be under the evaluation of trust. Trust credential in which each trustee has certifications signed by other parties. Trust policy is used to assure that trustees’ credentials comply with the trust policy. Moreover, authorization is considered as an advantage by having an identity for each node which can be public key, email address, CPU, hard drive serial number or an IP address. Another advantage is the separation of the trust-based authentication module logic and the application logic which will facilitate the customization of both logics.

The Framework used to implement the middleware consists of the middleware authentication layer, secure communication layer and trusted proof layer. In the middleware authentication layer, when a mobile middleware wants to communicate with another mobile middleware they have to be authenticated by going through this layer which will establish an interaction between the trust proof layer and the requester to filter trusted requests from other requests. While in the secure communication layer, only authenticated middleware can communicate and abstraction is used to handle the complexity and heterogeneity of different parties. The trusted proof layer handles the main tasks such as modifying the trust policies, searching credentials and trust proofing.

G. DMW

A middleware for digital rights management (DRM) developed to enforce DRM in p2p-like networks that consist of mobile and stationary devices [12]. It tackles content management challenges in p2p networks. DMW addresses DRM enforcement aspects and content sharing through super-distribution. The proposed middleware is based on a multi-tiered service-oriented architecture. DMW is a network oriented middleware that resides in each peer device; it acts as a facilitator for value-added services with access control and copy control during content rendering and distribution. DMW follows a balanced decentralized approach to meet the objectives of p2p content sharing and revenue generation with reasonable amount of restriction; it also has the ability to address issues related to interoperability, content manageability, compatibility, trustworthiness and support for protected content distribution. The DMW multi-tier architecture is as follows: Upper Tier: Application level that provides services to an application for interpreting the rights and rendering the secure content. Middle Tier: Operating system (OS) level that provides DMW specific extensions to the OS that help track illegal use of content. Lower Tier: Hardware level used for rights enforcement.

H. S-MARKS

This is a secure middleware for portable devices in a pervasive environment [13]. It incorporates security in the middleware design to address important issues such as device validation, discovering resources, malicious recommendations and privacy violations, which according to the authors, available middleware have failed to tackle in a one stop solution. The importance of this approach lies in that sharing services among portable devices has been subject to distrust with the available security concerns in pervasive computing environments. That’s why a single solution is proposed that may enhance the adoptability of mobile devices. The four main models present in this approach are: (1) Valid Device Discovery, which restricts the interaction to only valid devices with the use of a challenge-response mechanism. (2) Trust Based Resource Discovery based on a trust table in each device containing the services available by the neighboring nodes with a corresponding trust value that
can be upgraded dynamically. (3) Handling Malicious Recommendation to minimize their effects on overall recommendations. (4) Avoiding Privacy Violation related to resource sharing.

I. AMUWA

An approach for an authentication middleware designed to supports ubiquitous web access [4]. It is a middleware extension used to secure ubiquitous data access through Web services. It addresses security by using a combination of different types of authentication methods according to different places. Therefore, the middleware uses different levels of authentication based on the place where the user is at. This middleware tried to tackle some issues related to the support of heterogeneous authentication and authorization. Access control policy plays a great role in the strength of the authorization and authentication provided to access specific services. Moreover, the support of user roaming is necessary to allow a user who is working away from home to be authenticated without the need of a technology that strongly authenticates the user.

The middleware is based on FAME-PERMIS (Flexible Access Middleware Extensions to PERMIS) which is intended to ease multi-factor authentication and authentication strength that supports a wide collection of authentication methods including IP addresses, username and password pairs, certificate-based soft tokens, Java cards and biometrics. As a result, the user will have the freedom to choose any combination of authentication methods to reach high level of assurance (LOA). The middleware extension is divided into two parts the FAME and the PERMIS. These two parts are linked to SHIBBOLETH. The FAME will handle the user identification and authentication via web browsers, and the derivation of the LOA according to the authentication method used in the identification/authentication process. The LOA will be sent to the authorization infrastructure through the SHIBBOLETH SAML message. Then SHIBBOLETH will decide on which privilege to be given to the user. The middleware tries to follow a standard that will define the strength and the level of authentication method using NIST that define four levels of authentication strength.

IV. DISCUSSION

As we explored the different approaches in security middleware, it was clear that several issues and directions have been addressed by different researchers. However, in the particular area of pervasive and ubiquitous computing, there are some important aspects security middleware shares. In this section we discuss the different security middleware approaches and their properties in light of their application in pervasive environments versus ubiquitous environments.

A. Pervasive computing

The nature of pervasive computing is to blend-in computing devices within the background environment of our everyday life, but due to its limited system resources and dynamic nature it has been susceptible to many security issues. S-Marks [13] is one of the projects intended to tackle this problem. This middleware starts by validating the device discovery mechanism and moves on to build a trust-based resource discovery. Then it handles false recommendations when validating a new device and finally handles the issues arising while sharing the devices. Another project proposed to solve the security constraints in the pervasive computing environment is the Security-Supportive Middleware Architecture (SSMAP) [9] is based on reconfigurable and on-demand assembled security components. These components reside on distributed virtual servers and are aggregated on demand to serve a specific transaction. These components consist of various security services such as the Trust Management which adopts authentication, trust negotiation and maintenance to provide trust relationships.

The difference between these two approaches is that SSMAP builds an active space architecture that confines all the security services in one place as well as providing a virtual execution environment for the client to access these services. This approach eliminates the need to communicate with various heterogeneous devices in order to get the security services. As for S-Marks, this middleware focuses on addressing security concerns when sharing services among portable devices by building security functionalities within the middleware design. S-Marks adopts an open architecture and supports context and situation awareness services. It also involves the trusted devices of the network in some critical decision making such as the validation of new device when trying to join the network. In this case, every other node in the network sends a challenge to the new node and according to the response they get, the nodes provide their recommendation whether to accept or reject the joining of this new device. S-Marks have completed the first prototype of the middleware by fully implementing the device validation and trust based resource discovery while the other functions are to be completed in the next stage. As for SSMAP, the paper demonstrated an architecture that is not implemented yet, the next stage will see them handle the detail design issues and when completed it will be implemented on a virtual executable environment named User Mode Linux.

Another challenge in the dynamic networks is the management of content sharing between the pervasive devices. To address this problem, TMAHP2PA [6] demonstrates a trust-based middleware for secure digital content sharing between pervasive devices in ad-hoc p2p networks. The main motivation is to provide flexible security services to the p2p applications and to have an autonomous trust management that eliminates the complexity in the establishment of new relationships, the dependence on a central server and the need for frequent manual setting to build trusted relationships between devices. Another project trying to solve a similar problem is the DMW [12] middleware, which proposes a multi-tiered architecture for digital right management (DRM) in p2p-like networks. This middleware resides on user devices to provide a trusted environment for content sharing and DRM enforcement.

The difference between the two approaches is that project TMAHP2P considers the trust management as the basis of security in p2p applications; hence it plays the role of
enhancing the security aspects of the operating systems of these devices in order to protect the access to resources, prevent attacks, make use of local common knowledge and manage new relationships. Also, this middleware aims in minimizing the resource consumption and user intervention making it easier to be integrated into the security architecture of the pervasive devices. A p2p digital content sharing application has been developed to test this middleware, which was successful on two PDAs on both Linux and Windows XP. In DMW, the heterogeneous nature of a p2p networks has led to the adoption of a multi-tier approach to build a trusted system. A generic OS will not be able to satisfy this option due its limited capabilities in detecting and handling the content violation; therefore, DMW extends this OS with additional services that will fill-in the security gaps in the application, OS and hardware layer to establish a trusted system.

Smart home environments are another application domain requiring the services of pervasive computing. This domain relies heavily on ambient intelligence transforming the ordinary home environment into an electronic home that is both responsive and sensitive to the presence of people. Consequently, this technology requires gathering huge amount of personal data which raises many concerns from the consumer point of view. Therefore, a Middleware for Secure Home Access and Control [10] protects personal data and access control by only allowing the relevant stakeholders to benefit from the services. This security middleware is based on two main components one is the JCCM (Java Card Certificate Management) which moves part of the semantics of Cryptoki inside a Java Card to handle the certificate management. Also, it uses generic database schemas with security features that implicitly obfuscate user data. What differentiates this project is that services are decoupled from security issues, which therefore dismisses the tradeoff between performance and security for the application developer where the later is handled by the security middleware modules.

### B. Ubiquitous Computing

Ubiquitous computing environments refer to computers embedded in any device that can communicate with other devices over ad-hoc wireless networks. Ubiquitous devices users can access different resources and services; they expect resources and services availability anytime and anywhere. Security issues can be a concern here, thus a comparison of security middleware approaches in this field is discussed. AMUWA [4] is a security middleware focusing on providing authentication and give it a higher priority assuming different authentication methods are needed for users at different locations. This approach is based on FAME-PERMIS (Flexible Access middleware Extensions to PERMIS) which is open source solution to support inter-institutional sharing of web resources and uses SOAP-based token Request/Reply messages. This middleware supports a wide range of authentication methods like IP addresses, username and password pairs, certificate-based soft tokens, Java cards and biometrics. As in most of web services, username and password are the basic authenticators to access the services, but providing a third authenticator like IP address will eliminate non-authorized users from another organization from compromising the resources.

On the other hand, SMMU [8] presents a design for security middleware focused in providing more than one security service in addition to authentication. This middleware is supported with trust management that handles access control, delegation rights, monitoring and controlling devices. This middleware has a policy manager that deals with authorization, delegation and obligation policies. All these policies are implemented using XML and stored in a repository. Having obligation policy allows the middleware to be event-driven and supports real-time scenarios where if a set of conditions occur; a specific action will be established as reaction for that event. As for the authentication, credentials from the user have to be sent and verified in the service repository to access the service. This middleware was implemented using CORBA technology because CORBA services were helpful in implementing some needed functionalities. Furthermore, AMUWA [11] follows a good authentication approach, but it has some limitations that might not allow an authorized user to access web services. As been said if the user was far from his/her workplace and wants to get some important documents through the web service, he/she will not be able to because the policy disallows users inside the Intranet of the organization to access the service. Therefore, there is a need for a control process to manage authentication what techniques are to be used and in which situation. SMMU follows an access control based approach that will need a well defined delegation rights. Delegation rights control is very important in the matter in giving un-authorized user or objects a right to access services and resources.

As for TDAMU, it focuses on providing support for role-based trust and accountability. Even though TDAMU offers flexibility in access control and provides strong authentication, it has some limitations in authorization and rights delegation. For example if one user grants permission for a specific role to a delegated user and another user grants negative permission for the same user, this could result in a conflict problem in authorization and delegation rights. Therefore, an unpermitted user can have delegation rights to something he/she is not authorized to.

SGSC [7] is somehow different than other projects we discussed since it provides an efficient and automated security service for secure group management and it is implemented in a reconfigurable context sensitive middleware. By using the middleware service, distributed applications can send or receive secure messages among members of a secure group, but before that the members of the secure group have to be identified. Then in application execution, the secure group will be managed automatically according to the previously identified members by the middleware service. Other researches worked in providing security for a group communication using different methods using network protocol, group key generation and distribution techniques. These methods failed in scaling well with the group size and were costly in the matter of rekeying and key distribution across the group. SGSC uses various
data about the surrounding environment to control the interactions in distributed applications in mobile ad hoc networks. Situation-Aware Interface Definition Language (SA-IDL) is developed to describe situations and situation-aware components. SGCS deals with securing only members or devices that are within the same group. Therefore, if another group wants to join the secure group, then all members from these two groups can form a large secure group using the same group key. SGCS was shown to be best suited for mobile ad hoc networks than the other approaches because it has fewer operations from encryption/decryption in a single device. Moreover it has a better communication management for adding and deleting devices or members from a secure group.

V. CONCLUSION

In this paper we examined several projects addressing the design and implementation issues of security middleware. We noticed that different projects impose the security in the middleware in different ways. Some consider security by providing an efficient access control and authentication methodology. While others emphasize providing the protection for the content shared on the network rather than the access. In addition some projects present security middleware for group communication and P2P networks to grant smooth interaction between the connected devices. Different technologies were used in implementing the security middleware such used CORBA, XML, SOAP and JAVA. Even though, most projects succeeded in implementing their proposed security middleware, it is not easy to provide a fully-secured middleware that can combine all security components in one stop solution. At this stage, each approach addresses a part of the problem and there is no single security middleware that could claim full protection for a pervasive or ubiquitous environment. It is feasible to claim that due to the nature of pervasive computing environments and their devices that usually have limited resources, it is difficult to have an optimal security middleware.

ACKNOWLEDGEMENTS

This work was partially supported by a UAE University research grant #01-04-9-11/08.

REFERENCES