Three Way Authentication Protocol for Privacy Preserving and Ownership Authentication Transfer for Ubiquitous Computing Devices

PRADEEP B.H
Department of I & CT
Manipal Institute of Technology
Manipal University, Manipal-576104
INDIA
pradeep.kabbar@gmail.com

SANJAY SINGH
Department of I & CT
Manipal Institute of Technology
Manipal University, Manipal-576104
INDIA
sanjay.singh@manipal.edu

Abstract: Now a days almost everybody is having a portable communication device, be it a laptop, a tablet or smart phones. The user would like to have all the services at his fingertips and access them through the portable device he owns. The user would exchange data with the other user or the service provider or control the smart appliances at his home. The interactions between the user's device and the service provider must be secure enough regardless of the type of device used to access or utilize the services. In this paper we propose a "Three Way Authentication (TWA)" technique intended to preserve the user privacy and to accomplish ownership authentication in order to securely deliver the services to the user devices. This technique will also help the users or the service providers to check whether the device is compromised or not with the help of the encrypted pass-phrases that are being exchanged. The users use the devices to store most of the valuable information and will prove risky when the device is borrowed by the other user or when it is lost or stolen. To safeguard the user data and also to preserve user privacy we also propose the technique of Authenticated Ownership Transfer (AOT). The user who sells the device has to transfer the ownership of the device under sale. Once the ownership has been transferred, the old owner will not be able to use that device at any cost. Neither of the users will be able to use the device if the process of ownership has not been carried out properly. This also takes care of the scenario when the device has been stolen or lost, avoiding the impersonation attack. The proposed protocol has been modeled and tested with Automated Validation of Internet Security Protocols and Applications (AVISPA) and is found to be safe.

Key–Words: Ubiquitous Computing, Three Way Authentication, Ownership Authentication Transfer

1 Introduction

A ubiquitous computing (Ubicomp) is imagined as a system with large number of invisible, collaborating computers, sensors and actuators interacting with user devices. Data about individuals who are in the environment is constantly being generated, transmitted, modified and stored. The user data present in the ubiquitous environment is very sensitive and protecting private data of every user is a major concern. In the this era of the sophisticated technology and gadgets the user owns a number of portable devices like the PDAs, Laptops, Mobile Phones etc. with varying computing capabilities in order to access various types of services that are being provided by the service providers. It is very much important to secure the service interactions between the user and the service providers. If interactions or the transactions are not secure then the user will be hesitant to avail the services by providing the most sensitive data hence revenue loss for the service providers. For example a user who wants to have a secure bank transaction will not go for accessing his account by providing the username, password and also his account details if he is not sure whether the connection is secure. Hence it is important that the user's details are hidden from the third party and provide the required security.

In developing countries with slow economy, people tend to buy a used device as they can not afford for a new device. In such cases there will be scenario of selling and buying a used device for a lesser price. If a user wishes to sell his device, the ownership of the device has to be transfered. Since the device contains valuable information about the user and also will have the access to the valuable information present at the server, due care should be taken to delete the information of the old owner before the devices is sold to the new owner and store the details of the new owner in the device as well as in the server. Previously many

approaches have been proposed in this regard and despite our best efforts, we were not able to find any similar protocols, other than those cited below to have a comparison.

The main contribution in this paper are two protocols which fulfills the security requirement for the portable devices in the ubiquitous environment.

- The first protocol called Three way Authentication (TWA) [1], concentrates its functioning on the user privacy and authentication using the concept of pass-phrases. This provides a better security during the service interactions between the users and the service providers.

- The second protocol is Authenticated Ownership Transfer (AOT) [2], which is intended to work for secure and authenticated transfer of the device ownership from the old user to the new user. This provides the security for the Ubiquitous device from theft and also avoids impersonation attacks.

Rest of the paper is organized as follows. Section 2 discusses the related work followed by explanation of the proposed protocols in section 3 and 4. Section 5 presents the security analysis of the protocols. Section 6 presents the discussion on the proposed protocol and finally section 7 concludes the paper.

2 Related Work

In the recent past, number of authentication protocols have been presented but found to be generic in nature and not specifically suited for ubiquitous computing environment.

Jalal Al-Muhtadi et al [3] suggested different wearable and embedded devices such as smart jeweleries, active badges and smart watches etc which contain an ID for authentication, but the user should carry it wherever he goes. Also there are chances of the device being lost and fall into wrong hands. U.P.Kulkarni et al [4] and C. Lesniewski et al [5] has used the concept of Certifying Authority (CA) for authentication, which requires the user to register his devices and also requires the user to maintain his certificate on a regular basis. A non technical user would find it difficult to manage the certificates and it would be an unnecessary burden. If a person is having more than one device, then he needs to have independent certificates for all the devices he owns and should be managing his certificate for each device time to time and should be spending more time on this rather than actually doing his work.

Wenjuan Liu et al [6], has used the concept of information hiding used in TCP/IP packets. However this approach might not be useful at all time. It can mostly be used as the trustworthy authentication of security devices such as fire walls. Every time the information or the request is being sent, it will be encapsulated. This may lead to encapsulation of non-sensitive information. Also the encapsulation will not be able to differentiate between the sensitive and non-sensitive information. This limitation will lead to high computational and transmission overhead. Any loss in data during transmission will lead to inconsistency in the request or the information sent.

Adrian Leung and Chris J. Mitchell [7] in their work have proposed manual authentication protocol to authenticate the user and his device. This protocol authenticates the two devices using a combination of an insecure wireless channel and manual data transfer. As it uses the insecure channel, the system may be susceptible to attacks of any kind for the wireless network [8]; whatever is the information, be it sensitive information or non-sensitive can be tracked and attacked. Moreover the user has to transfer the data manually in order to be authenticated. In the current world of fast and high speed technology people tend to go towards automation of the systems to get their work done quickly; therefore it will not be good enough to go with the manual system for authentication. Also the author mentions that the user, user device and the device management entity needs to be in close physical proximity with each other during initialization phase, which actually wipes out the concept of mobility and ubiquitous computing.

Her-Tyan Yeh and Hung-Min Sun [9] in their work have mentioned that, two clients will register to two distinct servers, and the clients are subjected to indirect mutual authentication. However this cannot be incorporated in ubicomp environment because having more servers results in multiple point of failure, cost involved in physical and network security would be in multiples of that of three party authentication, trusting only a distinct server of users choice in the ubicomp environment may not be possible. The user has to trust all the servers assuming that almost all the servers are trustworthy. For computational reasons there has to be only one Central Key Server (CKS) as a data center. Multiple servers will result in excessive message exchange over the network resulting in computational overheads due to communication between several servers.

Paulo Tam and Jan Newmarch [10] in their work have suggested the concept of transferring the ownership of the device. The owner (old owner) of the device will send the message to the device itself that it is being bought by the other user (new owner). The

device will send the message to the new owner saying that its ownership is about to change to you (new user), do you accept or reject. The new owner sends the response to the device, and the object will in turn send an acknowledgment on the status of the transfer to the old owner. However when the owner of the device is selling the device to the new owner, sending the message to the device itself does not seem logical. Moreover to which device of the user, the device under sale is sending the message is not known. It is feasible if the new owner of the device has one more device under his ownership. Nevertheless if the user has no other device previously and it is his first device then there is no possibility for the device under sale to send the message to its new owner asking his consent on the ownership transfer. In ubiquitous environment the ownership transfer has to be informed to the central server instead of informing to the device under sale.

Jurgen Bohn [11] has mentioned that the user can borrow or lend the device to his friend or the stranger. The data of a particular user can be retrieved from the instant personalization server at any time and from anywhere for a specific time. Once the time limit is exceeded, the session will end and the user needs to quit the session or restart it. After using the device, the user can release the device and return it back to the owner of the device. However, the very basic idea of sharing the personal device with a friend or a stranger may cause information to be public. This could be due to other user being malicious (intentionally causing harm) by installing some kind of software which can record the data of the user or inadvertently installing malicious software which can save the users data. Moreover due attention should be paid to the fact that the device could come with old data, if the transfer is incomplete due to technical reasons such as network congestion or lack of connectivity. The owner of the device may also turn out to be malicious with respect to the other user. The user may install a software that records all the data that has been retrieved and sent from that device before encryption and after decryption. Later the user may be subjected to the impersonation attack. Moreover when the time limit is exceeded, there are chances that the user may have to end the session or restart it due to network latencies or unresponsive server when the user is trying to retrieve or release the data.

Yongming Jin et al [12] has described the transfer of RFID from the old owner to the new owner. They define a protocol to safeguard the privacy of the respective owners by putting the clean stop between the transactions of old and new owners by means of a secret. The authors have suggested the use of RFIDs for the ownership transfer. However there are many security concerns with respect to the RFID tags. One of the primary RFID security concern is the unauthorized tracking of RFID tags. The tags are read by anyone in the world; if the person who read the tag is malicious can pose a risk by either impersonating the user or trying to manipulate the user data and use it for a wrong purpose. RFIDs working at a shorter range are vulnerable to skimming and eavesdropping. Even though certain RFID tags use cryptographic features, the cost and power requirements are very high when compared to the simpler RFID tags. Thus, the cost and power limitation has constrained manufacturers to implement cryptographic tags using substantially weak encryption schemes, which are weak to resist the sophisticated attack. Moreover, the power available in the hand held devices is limited; these tags cannot be incorporated in the devices.

Abdullah M. Alaraj [13] has suggested that the users has to go to some officially designated place for buying or selling the merchandise and to complete the process of ownership transfer. He also makes an assumption that certain equipments are required for ownership transfer and tries to improve the fairness by including the transfer of money through the bank servers. However going to an officially designated place that deals with buying or selling of merchandise is suitable only for goods like cars or real estate; this scenario will not be suitable for ubiquitous computing devices. The process of ownership transfer requires only a central key server and a device meant for sale. Submitting buyers bank details to third party might be risky at the time of payment. Even if the system provides the best servers for transaction and promotes the users to submit their bank details to the device in an office meant for buying and selling of the merchandise; the device or the system in that office might turn out to be malicious.

3 Three Way Authentication (TWA)

In this paper we propose a simple and more user friendly approach to authenticate the user and his device in the ubiquitous environment. The proposed solution has the following phases:

- Initialization
- User Registration
- Connection between users
- High Level Transaction

Table 1: Notations Used

Symbol	Meaning	Symbol	Meaning
T_A	Token of the corresponding user A signed by the KDC	CKS	Central Key Server
$ID_{machine}$	Machine ID or the Device Serial Number	KDC	Manufacturer's Key Distribution Center
U_A	User A	U_B	User B
U_M	Device Manufacturer	DMN	Device Model Number
$E_{P_{CKS}}$	Encryption Using Public key of CKS	$E_{P_{KDC}}$	Encryption Using Public key of KDC
N_A	nonce generated by A	N_B	Nonce generated by B
N_U	Nonce generated by user U	N_{CKS}	Nonce generated by CKS
ID_B	User ID or User Name of the user B	ID_A	User ID or User Name of the user A
ID_M	Manufacturer's ID	T_S	Time Stamp
P_{CKS}	Public key of the CKS	PW_U	Password of the User U
$P1$	Pass-phrase which is known to central key server alone	$P2$	Pass-phrase which is known to user and the CKS
K	One Time Session Key	SR	Service Request
SP	Service Provider	Ack_D	Acknowledgment for device authentication at the time of registration
Ack_U	Acknowledgment for user registration	H	Hash of the message using MD5 or SHA1 etc.
OTP	One Time Password	$TempID$	Temporary Identity or pseudo Identity or the respective user
N_M	Nonce of manufacturer	$\|$	Concatenation Symbol
OTC	Ownership Transfer Confirmation	OTR	Ownership Transfer Request

3.1 Initialization

This phase is carried out at the site of manufacturer and the manufacturer is treated as the user here. The assumption made is that the manufacturer is already an authenticated person. The manufacturer will be registering the device with Key Distribution Center (KDC) meant only for the manufacturers by sending the machine ID, the Device Model Number (DMN) and nonce of the manufacturer (N_M). The machine ID and the DMN is encrypted by public key of the KDC. The KDC will then generate a token T which is the hash of manufacturers ID and a time stamp. This token T is encrypted using nonce of the manufacturer and sent to the manufacturer through a secure channel. The token will also be stored in the CKS for future use. This is performed before the device is sold. The token obtained by the manufacturer will be encrypted using RSA [14] and then hard coded into the trusted portability module [15][16] which is embedded in the portable device. The message exchange that takes place between KDC and manufacturer during the process of initialization is as follows:

1. $U_M \rightarrow KDC$: $E_{P_{KDC}}(ID_{machine} \| DMN \| N_M \| ID_M)$
   The manufacturer will send the unique machine ID along with the device model number, nonce (N_M) and ID of the manufacturer to the KDC to register the device being manufactured. This message is encrypted using the public key of KDC. A nonce is like one time password which is generated by the device itself for every new transaction. When a nonce is generated, a corresponding pair which acts as a private key also has to be generated for decrypting the encrypted message.

2. $KDC \rightarrow U_M$: $E_{N_M}(T)$, where $T = H(ID_M \| T_S)$
   The KDC decrypts the message received using its private key and after manufacture's authentication credential verification, generates a token T which is a hash of manufacturer's ID and a time stamp which indicates the time when the device was registered. In future the token will be helpful to recognize to which manufacture the device belongs to. Then the KDC sends the token T through secure channel by encrypting it using the nonce of the manufacturer.

3. $KDC \rightarrow CKS$: $E_{P_{CKS}}(ID_{machine} \| DMN \| T)$
   Once the token has been generated and sent to the manufacturer, the work of the KDC is done. Now KDC will send a encrypted message to CKS using the server's public key. The message consists of machine ID, DMN and the token T. This is done because the CKS needs this information to carry out the device authentication in future transactions.

The above explained process of device initialization is summarized in Fig. 1.

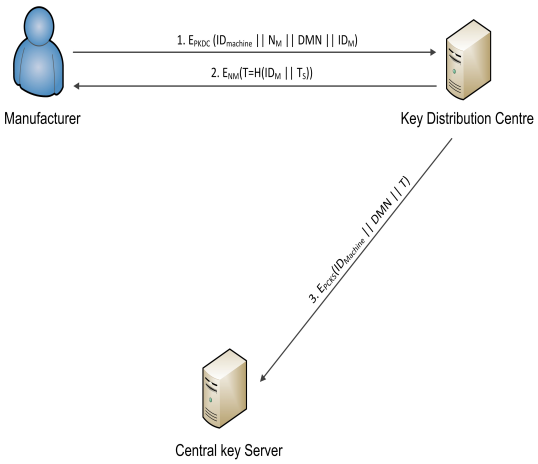

Figure 1: Diagram Showing Device Initialization Process.

3.2 User Registration

This phase is carried out only after the device is sold. The user who is the owner of the device needs to register to the CKS. He sends the message containing the machine ID and the token encrypted using the public key of CKS; the CKS will decrypt the message received from the user by its private key. The token and the machine ID sent by the user is compared with the token and the machine ID which has been stored in the server's database during the initialization phase. If both are same then the server sends the acknowledgment (Ack_D) indicating that the authentication of the device was successful and also an One Time Password (OTP). Now the user needs to register himself by providing the user ID and the OTP to CKS. Once the registration is done, the server sends an acknowledgment (Ack_U) to the user along with the Temp ID. Now the user is ready to avail any service through his device. The messages that are exchanged during this phase is given below:

1. $U \rightarrow CKS$: $E_{P_{CKS}}(ID_{machine} \| T \| N_U)$
   Once the device is bought, the use needs to register himself to the CKS. To do so the device has to be authenticated. So the user sends a message containing the machine ID, his nonce (N_U) and the token T, encrypted using the public key of

CKS. CKS decrypts the contents of the message using its private key and compares them with the contents in its database. Only when the contents match, the CKS will send the success message.

2. $CKS \rightarrow U \ : \ E_{N_U}(Ack_D \| OTP)$
CKS will send the device acknowledgment (Ack_D) indicating that the device has been successfully authenticated, which informs the user that he needs to register him to the CKS using the One Time Password (OTP) received by him, where $Ack_D = H(T \| DMN)$. This whole message is encrypted by using the nonce of the user.

3. $U \rightarrow CKS \ : \ E_{P_{CKS}}(ID_U \| OTP \| N_U)$
Now the user sends a message containing his credentials like User ID, OTP and a nonce. The message is encrypted using the public key of the server.

4. $CKS \rightarrow U \ : \ E_{N_U}(Ack_U \| TempID)$
Once the message from the user is received by the CKS, it will decrypt it using its private key and register the user. At the end of this phase, the server sends an user registration acknowledgment (Ack_U) and a TempID which is encrypted using the nonce sent by the user. The user after receiving the message will decrypt it and retrieve his TempID.

The process of user registration is shown in Fig. 2.

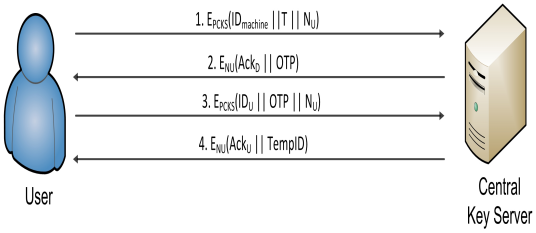

Figure 2: Diagram Showing the User Registration Process.

3.3 Connection between the Users

Every user will be given a TempID by CKS during the user registration phase. In this phase and in the high level transaction phase, user needs to send his TempID along with the nonce that is generated in the device. To connect the device to the fellow user's device, the user needs to be connected through the Internet. The user, who wants to connect to a particular user, sends the token and his Temp ID along with the details of the device he wants to connect, to the CKS in the network. The key server compares the token and his ID with that present in its database. If they are same, then it requests the called user to send his details and compares it. If the called user credentials are verified, then the CKS sends a one time session key to both the devices. If any of the user's details is not matching with his details in CKS's database then CKS sends the message to the other user that the device is not an authorized one and he is trying to connect to a malicious entity. The scenario of the connection between the two users is described as follows:

1. $U_A \rightarrow CKS \ : \ E_{P_{CKS}}(ID_{U_B} \| T_A \| N_A \| TempID)$
The user who is interested in establishing a connection with the other user of his interest sends the target user ID (ID_B in this case) along with his token, a nonce and his TempID. This message is encrypted using the public key of the CKS.

2. $CKS \rightarrow U_B \ : (ID_B \| ID_A)$
CKS decrypts the received message and compares it with the contents in its database. It check for the users details in CKS's database, if found, the user is authenticated. It also checks for the device details and if they match with the details in its database, then the CKS sends the message to the other user saying that the user A is trying to connect to user B.

3. $U_B \rightarrow CKS \ : \ E_{P_{CKS}}(T_B \| N_B)$
Once the message from the CKS is received by the user B, he will send the details like token and a nonce by encrypting them using the CKS's public key.

4. CKS decrypts the message sent by the User B. If authenticated, both the users will be given a one time session key (K) to encrypt and decrypt the messages that are being exchanged between them. There should be some time bound for the session keys, after which it must get expired by itself. It will avoid the replay attack. The session key sent to both the users will be encrypted using their respective nonce.
$CKS \rightarrow U_A \ : \ E_{N_A}(K)$
$CKS \rightarrow U_B \ : \ E_{N_B}(K)$

The process of connection between the two users is summarized in Fig. 3.

3.4 High Level Transactions

The user does not just try to connect his devices to the other users devices, but also goes for the transac-

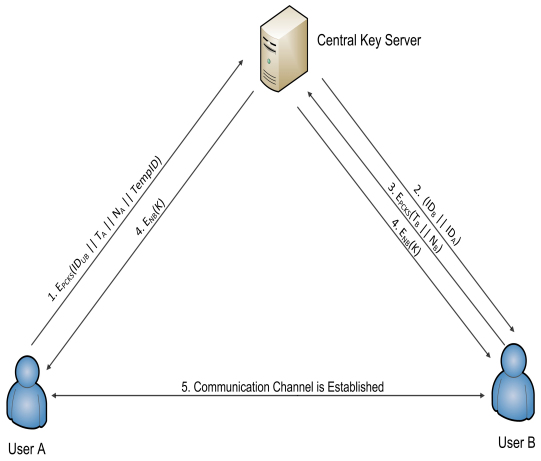

Figure 3: Diagram Showing Connection Between the Users.

tions wherein the user needs to submit his most important and sensitive details to the service providers. For example he may try to access his bank accounts from his device where he needs to submit his account details; this kind of transactions are generalized as high level transactions. The user who wants to have a high level transaction will send the request to the central key server to grant access to a particular service provider. This technique is analogous to Kerberos [17], where CKS acts as both the authentication server and the ticket granting server. The service provider is the server to which the user wants to gain access. In case of the bank transaction there is no need to search for the server as a bank has a dedicated server. In case of other transactions like online shopping, the service providers are more than one and the best service provider has to be chosen. In this technique two different pass-phrases (P1 & P2) are used. P1 is what the only central key server knows and P2 is what the central key server and the user knows. The pass- phrase P2 is entered manually by the user and pass-phrase P1 is generated by the central key server. The pass-phrases act as a secret key or word adding up an extra layer of security in ubiquitous computing. The proposed technique can be explained in different steps as follows:

1. $U \rightarrow CKS$: $E_{P_{CKS}}(SR \| P2 \| T \| N_U \| TempID)$
   The user sends a service request (SR), pass-phrase P2, token (T), TempID and nonce encrypting it using the public key of CKS. The CKS decrypts the message and compares the contents with that in the database and saves P2. If the user is authenticated, the CKS will check the service request from the user and sends the ID of the service provider who is best suited to provide the service that has been requested by the user.

2. $CKS \rightarrow U$: $E_{N_U}(K \| E_{N_{CKS}}(P1) \| ID_{SP} \| P2)$
   CKS sends a session key (K), a pass-phrase P1 encrypted using nonce of the CKS and ID of the service provider. This message is encrypted using nonce of the user. The user cross checks the pass-phrase P2 and if it matches with what he has send in step 1, then he stores the session key.

3. $U \rightarrow SP$: $ID_{SP} \| E_{N_{CKS}}(P1) \| ID_U$
   Once the user receives the message, he decrypts it. The user sends the message containing ID of the service provider (ID_{SP}) and the encrypted pass-phrase P1 to the service provider.

4. $SP \rightarrow CKS$: $E_{N_{CKS}}(P1) \| E_{P_{CKS}}(N_{SP} \| ID_U)$
   The service provider forwards the encrypted pass-phrase P1 along with its nonce and the ID of the user from whom the message was sent by encrypting it using the public key of CKS to the CKS. CKS checks for the contents of the message with that in its database, only if they match the service provider is authenticated.

5. $CKS \rightarrow SP$: $E_{N_U}(P2 \| K \| ID_{SP}) \| E_{N_{SP}}(K \| ID_U)$
   CKS nows sends the pass-phrase P2, session key, and the ID of the service provider encrypted using the nonce of the user along with other message containing session key and the user ID encrypted using the nonce of the service provider.

6. $SP \rightarrow U$: $E_{N_U}(P2 \| K \| ID_{SP})$
   The service provider forwards the pass-phrase P2, session key, and the ID of the service provider encrypted using the nonce of the user, to the user. User decrypts the message from the SP, and checks P2, if it is the same what he had sent to the CKS and also checks whether the session key is the same as that received earlier. If both P2 and K match with their respective values, the user is now confident that the service provider is not a malicious entity.

7. Once the User and the service provider get the session key, they will exchange the messages which are encrypted using the session key.
   $U \rightarrow SP$: $E_K(Message)$
   $SP \rightarrow U$: $E_K(Message)$

The process of high level transaction is summarized in Fig. 4.

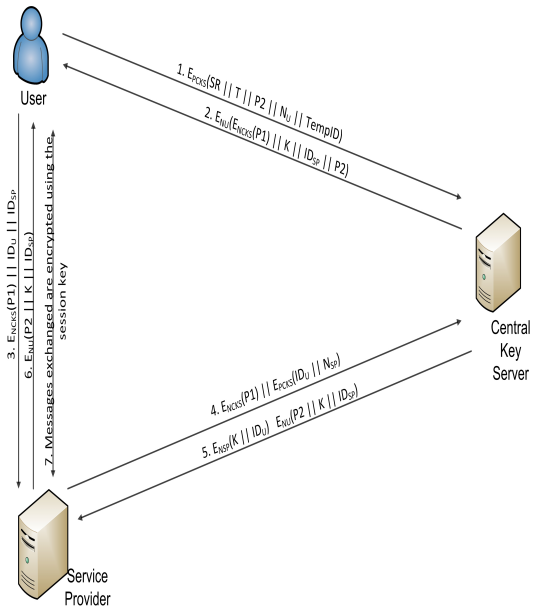

Figure 4: Diagram Showing the Process of High Level Transactions.

4 Authenticated Ownership Transfer (AOT)

Assumption: The value or the price of the device has been agreed upon between the users before transferring the ownership of the device.

Requisite: The users should be in physical proximity and the whole process has to be carried out in the device which is under sale.

The old user should introduce the new owner of the device to the CKS, in other words user A must transfer the ownership authentication credentials to the new user B. Once the new owner is introduced, the CKS will delete the authentication credentials of the old owner and saves the authentication credentials of the new owner in its database for the same device. Once the ownership rights has been transfered to the new user, the old user at any cost will not be able use the device. If the whole process of ownership transfer as mentioned below is not completed, neither of the users will be able to use the device. This also takes care of the scenario wherein the devise has been stolen, the thief cannot use the device. The ownership transfer for a given ubiquitous device is described below.

1. $U_A \rightarrow CKS$: $E_{P_{CKS}}(ID_A \| PW_A \| N_A \| OTR)$
The user A (Old User) sends the message to the CKS. The message consists of the user A ID, password of the user A, nonce of the user A and Ownership Transfer Request (OTR). This message is encrypted using the public key of the CKS. OTR consists of the ID of the user selling the device, ID and nonce of the user buying the device. OTR is also encrypted using the public key of the CKS, where $OTR = E_{P_{CKS}}(ID_A \| ID_B \| N_B)$. In this step the user A will introduce user B to the CKS. The nonce of the user B will be generated by the device when a request for ownership transfer has been initiated.

2. $CKS \rightarrow U_A$: $Ticket$
In response to the user A's request for ownership transfer, the CKS sends a ticket to the user A. The Ticket consists of the acknowledgment for ownership transfer to the user B. The ticket is encrypted using the public key of the CKS.

3. $U_B \rightarrow CKS$: $E_{P_{CKS}}(ID_B \| Ticket \| N_B)$
The user A will now hand over the device to the user B. Now the user B sends his credentials to the CKS. The user needs to send user ID, nonce and the ticket received from user A. It is to be noted that the ticket will be in the device itself.

4. $CKS \rightarrow U_B$: $E_{N_A}(\text{OTC})$
Once the CKS receives the credentials of User B, the CKS sends the Ownership Transfer Confirmation (OTC) to user B by encrypting it using nonce of user A. This message consists of the information about the money to be transferred and the account details of the destination account.

5. $U_A \rightarrow CKS$: $E_{P_{CKS}}(\text{OTC})$
The user B will hand over the device to the user A and the user A will decrypt the message, read the acknowledgment and then he sends the acknowledgment back to the CKS by encrypting it using the public key of CKS. By sending the acknowledgment back to the CKS, he confirms the ownership transfer of the device. Signing a particular message twice is required to strike a fairness in the deal. There may be some chances wherein either of the users may turn out to be malicious. This is done in order to obtain a confirmation from the user who is selling the device.

6. $CKS \rightarrow U_B$: $E_{N_B}(TempID)$
On receiving the message, CKS completes the ownership transfer of the device by sending the temp ID to the user B. The temp ID is encrypted using the nonce of the user B.

The above explained process of device ownership transfer is summarized in the Fig. 5.

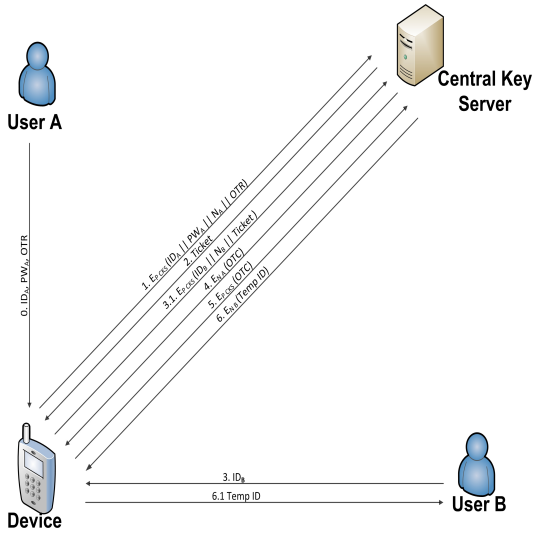

Figure 5: Diagram Showing Authenticated Ownership Transfer Process

5 Security Analysis of TWA and AOT Protocol

In order to analyze the TWA and AOT techniques, we have modeled the TWA and AOT protocols in terms of five HLPSL specifications, representing five different phases corresponding to the four execution phases of TWA and one scenario of AOT. The protocol is verified using the Automated Validation of Internet Security Protocols and Applications (AVISPA) [18][19][20] and is found to be safe.

5.1 Initialization Phase

The main task in this phase is to register the device under the KDC and CKS. We assume that the manufacturer is already an authenticated entity. The manufacturer uses a public key of the KDC to encrypt the messages that are sent to the KDC. The man in the middle attack is not possible in this case as the attacker will not be aware of the corresponding private key to decrypt the message from the manufacturer. The intruder attack if any is harmless as he just forwards the messages that are intercepted by him. The manufacturer sends the message to KDC by encrypting the message using KDC's public key. The intruder may try to intercept the message but he will not be able to decrypt it as he will not be having the corresponding key to decrypt the message. Now the KDC sends the message to the manufacturer. The message sent from KDC will be encrypted using nonce of the manufacturer. Since the intruder will not be having the corresponding key to decrypt the message, the attack fails. The KDC will also send the message to CKS by encrypting it using the public key of the CKS. The intruder attack fails in this step too, as the intruder will not be having the corresponding private key for the public key of CKS. He will not be able to decrypt the messages he intercepts. The first HLPSL specification formalizes initialization model of TWA, in which there are three protocol roles: Manufacturer (M), Key Distribution Center (K), and Central Key Server (C). The intruder (I) behaves either as M or K or as C, thus deceiving the other entities in the network. However this kind of attack is harmless as the intruder will not be able to read the contents of the messages being exchanged. Our HLPSL specification formalizes all this as follows.

1. $M \rightarrow I("K") : E_{P_{KDC}}(ID_{machine} \| DMN \| N_M \| ID_M)$
2. $K \rightarrow I("M") : E_{N_M}(T), where T = H(ID_M \| T_S)$
3. $K \rightarrow I("C") : E_{P_{CKS}}(ID_{machine} \| DMN \| T)$

Thus the man in the middle attack or the intruder attack fails. The intruder attack in the initialization phase is shown in the Fig. 6.

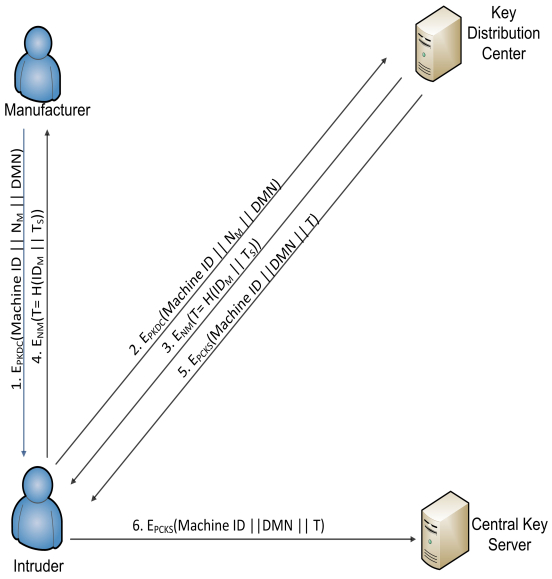

Figure 6: Diagram Showing Intruder attack during Initialization Phase

5.2 User Registration Phase

The user who has bought the device needs to register himself as the user with CKS. The user sends the

credentials from the new device to the CKS. The message is encrypted using the public key of the CKS and the attacker who is trying to decrypt the message will not succeed, as he is not aware of the corresponding private key. The message sent to the user will be encrypted using the nonce of the user. The user will be having a corresponding private key for his nonce to decrypt the message. Thus, the man in the middle attack fails here. Since the nonce changes after once usage, this eliminates the possibility of re-usage of the secret key. The CKS also checks the token that is sent from the device, compares it with the corresponding token in its database. Thus, this provides a security against spoofing of the devices. Since the device is from the authenticated manufacturer, there is no chance of a malware being present in the device. The intruder here just forwards the message as he fails to cause any harm. He will not be able to know the contents of the message as he does not possess the corresponding keys to decrypt the messages. The second HLPSL specification formalizes User Registration model of TWA, in which there are two protocol roles: User (U) and Central Key Server (C). The intruder (I) behaves either as U or as C, thus deceiving the other entities in the network. However this kind of attack is not harmful as the intruder will not be able to read the contents of the messages being exchanged between user and CKS. The HLPSL specification formalization is as follows.

1. $U \rightarrow I(\text{"}C\text{"}) : E_{P_{CKS}}(ID_{machine} \| T \| N_U)$
2. $C \rightarrow I(\text{"}U\text{"}) : E_{N_U}(Ack_D \| OTP)$
3. $U \rightarrow I(\text{"}C\text{"}) : E_{P_{CKS}}(ID_U \| OTP \| N_U)$
4. $C \rightarrow I(\text{"}U\text{"}) : E_{N_U}(Ack_U \| TempID)$

The intruder attack in the user registration phase is shown in the Fig. 7.

5.3 Connection between the Two Users Phase

The CKS cross checks the token received from the user. Only if the token matches with the token in the CKS's database, the user will be given the permission to communicate with the other user. The attacker will fail to decrypt the message, as he will not be having the corresponding private key for the nonce that is used to encrypt the message. The attacker will not be able to spoof the device or the user because the user ID of the user is registered under the token of the particular device and the token will be encrypted using RSA and hard coded in the device. Thus, the intruder fails to retrieve any information from the messages that are

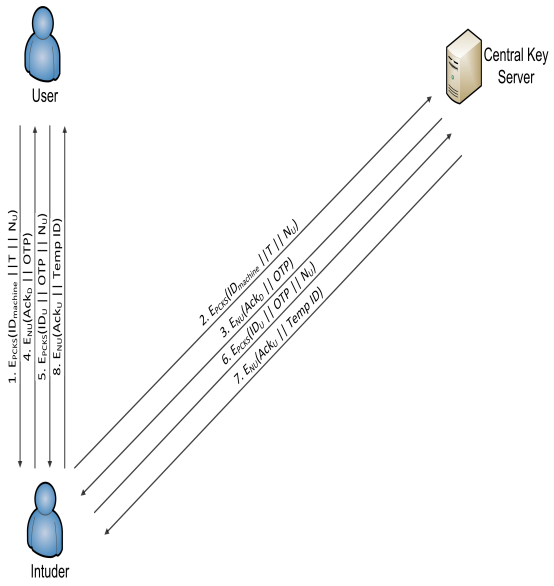

Figure 7: Diagram Showing Intruder attack during User Registration Phase

being exchanged, but will have to blindly forward the messages to other entities. The third HLPSL specification formalizes connection between Users model of TWA, in which there are three protocol roles: User A, User B and Central Key Server (C). The intruder (I) behaves either as A or B or as C, thus deceiving the other entities in the network. But this attack from the intruder is harmless as the intruder will not be able to read the contents of the messages being exchanged.

1. $A \rightarrow I(\text{"}C\text{"}) : E_{P_{CKS}}(ID_{U_B} \| T_A \| N_A \| TempID)$
2. $C \rightarrow I(\text{"}B\text{"}) : (ID_B \| ID_A)$
3. $B \rightarrow I(\text{"}C\text{"}) : E_{P_{CKS}}(T_B \| N_B)$
4. $C \rightarrow I(\text{"}A\text{"}) : E_{N_A}(K)$
5. $C \rightarrow I(\text{"}B\text{"}) : E_{N_B}(K)$

The intruder attack in the user registration phase is shown in the Fig. 8.

5.4 High Level Transaction

In this phase, the user submits a service request to the CKS along with his authentication credentials and a nonce. This nonce is used to encrypt the session key that is exchanged between the user and the CKS. The concept of pass-phrase has been used in order to provide an extra security feature. The user as well as the CKS will crosscheck their corresponding pass-phrases

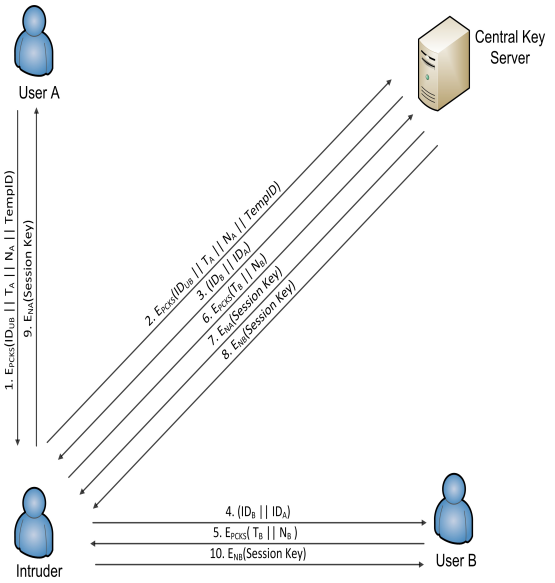

Figure 8: Diagram Showing Intruder attack during Connection between the Two Users Phase

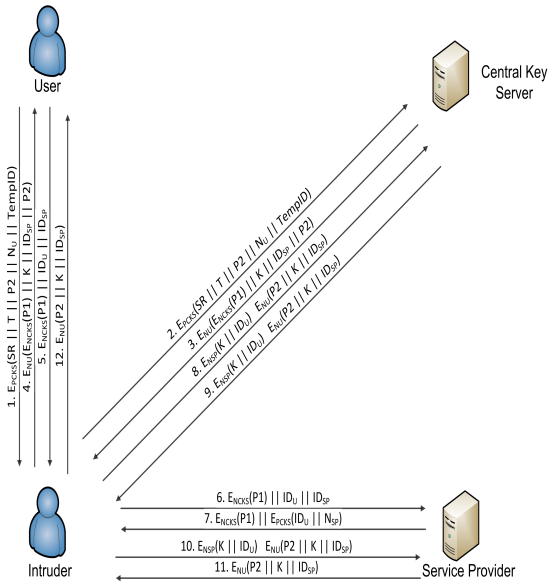

Figure 9: Diagram Showing Intruder attack during High Level Transaction

to ensure if there are any manipulations. If the pass-phrases are manipulated it is confirmed that some malicious entity is present in the network. The intruder will only be able to forward the message but will not be able to know the contents of the messages. Even if he tries to manipulate the messages, the changes due to manipulation will be reflected through the pass-phrases. Thus the intruder will fail to pose any kind of attack. The fourth HLPSL specification formalizes high level transaction model of TWA, in which there are three protocol roles: User (U), Service Provider (SP) and Central Key Server (C). The intruder (I) behaves either as U or SP or as C, thus deceiving the other entities in the network. However this kind of attack is not harmful as the intruder will not be able to read the contents of the messages being exchanged.

1. $U \rightarrow I(\text{"}C\text{"}) : E_{P_{CKS}}(SR \| P2 \| T \| N_U \| TempID)$
2. $C \rightarrow I(\text{"}U\text{"}) : E_{N_U}(K \| E_{N_{CKS}}(P1) \| ID_{SP} \| P2)$
3. $U \rightarrow I(\text{"}SP\text{"}) : ID_{SP} \| E_{N_{CKS}}(P1)$
4. $SP \rightarrow I(\text{"}C\text{"}) : E_{N_{CKS}}(P1) \| E_{P_{CKS}}(N_{SP} \| ID_U)$
5. $C \rightarrow I(\text{"}SP\text{"}) : E_{N_U}(P2 \| K \| ID_{SP}) \| E_{N_{SP}}(K \| ID_U)$
6. $SP \rightarrow I(\text{"}U\text{"}) : E_{N_U}(P2 \| K \| ID_{SP})$

The intruder attack in the high level transaction phase is shown in the Fig. 9.

5.5 Authenticated Ownership Transfer

In this phase the old user requests the central key server to start the ownership transfer of the device. Both the users new and old will use the same device to interact with the CKS. The users will feed the details into the device that has to be sent to the central key server. The messages that are going out of the device may be intercepted by the intruder, but the intruder will not be able to decrypt the message and know its contents as he will not be having the corresponding keys to decrypt the messages that are being exchanged. Thus again the intruder attack fails and there is no harm caused from the intruder to any of the authenticated entities in the ubiquitous environment. The fifth and final HLPSL specification formalizes AOT, in which there are two protocol roles: Device (D) and Central Key Server (C). The intruder (I) behaves either as D or as C, thus deceiving the other entities in the network. But this kind of attack is not harmful as the intruder will not be able to read the contents of the messages being exchanged.

1. $D \rightarrow I(\text{"}C\text{"}) : E_{P_{CKS}}(ID_A \| PW_A \| N_A \| OTR)$
2. $C \rightarrow I(\text{"}D\text{"}) : Ticket$
3. $D \rightarrow I(\text{"}C\text{"}) : E_{P_{CKS}}(ID_B \| Ticket \| N_B)$
4. $C \rightarrow I(\text{"}D\text{"}) : E_{N_A}(\text{OTC})$
5. $D \rightarrow I(\text{"}C\text{"}) : E_{P_{CKS}}(\text{OTC})$
6. $C \rightarrow I(\text{"}D\text{"}) : E_{N_B}(TempID)$

The intruder attack is shown in the Fig. 10. In this figure, we have shown only the device under sale instead of the two users for better understanding.

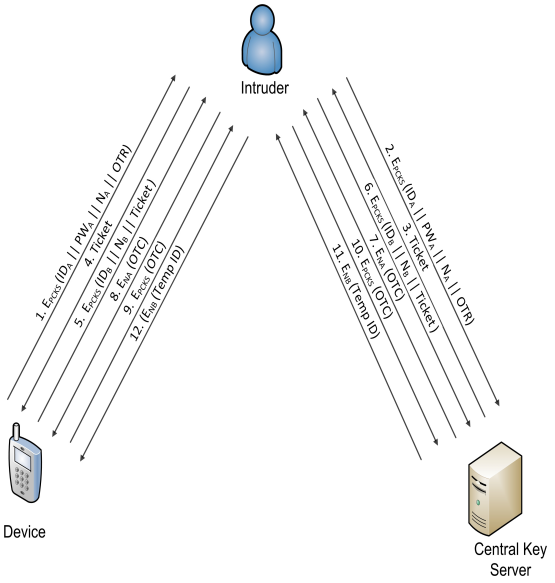

Figure 10: Diagram Showing Intruder attack on Authenticated Ownership Transfer of the Devices

6 Discussion

The Ubicomp devices are capable of establishing different types of connections as mentioned in the earlier part of the paper. The device goes through the phase of initialization where the machine ID is registered with the KDC to make sure that the malicious device does not try to access the services or the information from the other users. In the user registration phase the device is first authenticated and then the user is prompted for registering himself with central key server. This phase adds up to the security aspect as user with a non-authenticated device will not be allowed to register unless his device is authenticated. During phase of connection between the two users, the devices are authenticated by their respective tokens. The devices read the exchanged tokens which has the signature of the KDC which helps them for mutual authentication. Moreover the users submit their TempID's and other credentials which helps the CKS to authenticate the users. The concept of nonce and session key is used which adds up to the security. By making use of the nonce, the burden on the server to maintain the public keys of all the users and the service providers associated with it is avoided.

In the high level transaction phase, the two pass-phrases used are known only to the user and the central key server. However the crucial information which the user shares with the central key server is not available to the service provider or to any third party. From this technique the user will be assured that the service provider is also a trusted entity and so the service provider will be having the confirmation that the user who is requesting for its service is not the malicious entity. Since the central key server is the entity in between the user and the service provider, it will also be sure that the true user will be establishing the connection with the trusted service provider. In this technique, the user will be accessing the services with the help of his Temp ID and his actual details will not be available to the service provider. So this technique will protect the privacy of the user and will not allow illegal sharing of user's sensitive data. The technique will also ensure that the compromised device does not get into contact with the service provider. This is done with the help of the pass-phrases. So the Three Way Authentication technique will be a handy technique in the ubiquitous computing environment.

A malicious entity between the two users may intercept their messages and send the same messages to the server impersonating to be one among the two users. The malicious entity sends a number of SYN messages to the server. After receiving the SYN-ACK message from the server, the malicious entity will not send the expected ACK message to the server. The malicious entity sends a number of connection requests to the server and tries to create a bottle neck in the network. This is called the SYN flood attack [21]. SYN flood attack is type of denial of service attack where the TCP three way handshake is not completed as the client will not send the expected ACK messages. The technique presented in this paper avoids the SYN flood attack to a great extent. As mentioned before, the messages exchanged between the users are encrypted using a one time session key. Since the messages exchanged between the users are in encrypted form, the malicious entity will not be able to know the contents of the messages even though he has the access to the encrypted messages. Moreover the messages are exchanged through the server, only the server will know to whom the particular message has to be forwarded and from which user the message is transmitted. The messages are of no use to the malicious entity as he is not able to read the content of the message. Even though the malicious entity attacks the network by SYN flood attack, it can be avoided by using SYN cookies which eliminate the resources allocated on the target host. As this paper deals only with the user privacy and authentication, more detailed study and analysis with respect to the attack on the network will be carried out as part of our future work.

The Ubicomp device contains sensitive user data

and also an easy access to the data which are stored in the CKS, it is important that the user's data is protected. It is not secure for the user to borrow or lend his device for any kind of transaction. The proposed concept of ownership authentication transfer provides the transfer of ownership devices securely for the users interested in buying the old devices. If either of the users do not provide accurate information, the transfer will be aborted. At this point, a question may arises as to how often the ownership transfer can be done and can there be any temporary ownership transfer. How often the ownership is transferred depends on the user of that device. In developing countries people may tend to buy second hand devices, as the price of the devices will meet their budget. The ownership transfer can be done any number of times based on the interest of the users. The mobiles or the hand held devices are sold and the ownership is transferred permanently; there is nothing like temporary change in the ownership. Even though there is a need to change the ownership temporarily, it depends on the mutual agreements between the new and the old user. Temporary ownership transfer is purely at the users risk and it is not a part of this protocol. This concept also takes care of the scenario wherein the device has been lost or stolen. No person will be able to use the device other than its authentic owner. Thus it avoids the impersonation attack.

The protocol is verified using Automated Validation of Internet Security Protocols and Applications (AVISPA). AVISPA provides a language known as the High Level Protocol Specification Language (HLPSL) to describe the security protocols and to specify their intended security properties, as well as a set of tools to formally validate them. Experiments that were conducted on the vast library of Internet security protocols have indicated that the AVISPA Tool is a leading edge tool for Internet security protocol analysis. There are no other tool that exhibit the same level of scope and robustness providing the excellent performance and scalability to the best of our knowledge. The proposed protocol was modeled and tested using AVISPA tool and was found to be safe.

7 Conclusion

The security in ubiquitous computing along with authentication and preserving the user privacy is more important in the present world wherein people often get their work done at anytime and from anywhere through their portable devices.

This paper proposes a new technique of authentication which ensures the privacy of the user and makes sure that the user is convinced about the security he is looking for in order to submit the most sensitive data with the intention to avail the services from the service providers. This technique authenticates the user and the service provider to the central key server and also the user and the service provider are mutually authenticated with the help of the central key server. There is no burden on the non technical users to maintain his certificates or need for the user to carry authentication devices with them.

By incorporating the concept of ownership authentication transfer in the ubiquitous environment, we provide more security with respect to the owner's sensitive data and also to his device. The device can only be accessed by its authentic owner. If the device is stolen or lost the device cannot be accesses by anyone unless the process of ownership transfer is completely followed. This adds up the security in the ubiquitous environment. The future work of the proposed solution also includes the inspection of device connections like a social network, which brings to mind the trust management, where the trust quotient of the device should be calculated regularly and quickly.

References:

[1] P. Hanumanthappa and S. Singh, "Privacy preserving and ownership authentication in ubiquitous computing devices using secure three way authentication," in *Innovations in Information Technology (IIT), 2012 International Conference on*, march 2012, pp. 107 –112.

[2] B. Pradeep and S. Singh, "Ownership authentication transfer protocol for ubiquitous computing devices," in *Computer Communication and Informatics (ICCCI), 2013 International Conference on*, 2013, pp. 1–6.

[3] J. Al-Muhtadi, A. Ranganathan, R. Campbell, and M. Mickunas, "A flexible, privacy-preserving authentication framework for ubiquitous computing environments," in *Proc. IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW02)*, Jan 2002, pp. 771–776.

[4] U. P. Kulkarni, J. V. Vadavi, S. M. Joshi, K. C. Sekaran, and A. R. Yardi, "Distributed multi level security token based authentication for ubiquitous objects -dmsa," in *Proc.(IEEE) Ad Hoc and Ubiquitous Computing, 2006. ISAUHC '06. International Symposium*, Jan 2006, pp. 52–55.

[5] C. Lesniewski-Laas, B. Ford, J. Strauss, R. Morris, and M. F. Kaashoek, "Alpaca: Extensi-

ble authorization for distributed services," in *Proc.Proceedings of the 14th ACM Conference on Computer and Communications Safety (CCS-2007)*, Alexandria, VA, October 2007.

[6] W. Liu, X. Fu, S. Ouyang, J. Lin, and S. Teng, "Information hiding for pervasive trusted authentication," in *Pervasive Computing (JCPC), 2009 Joint Conferences on*, Dec. 2009, pp. 653 –656.

[7] A. Leung and C. Mitchell, "A device management framework for secure ubiquitous service delivery," in *Information Assurance and Security, 2008. ISIAS '08. Fourth International Conference on*, Sept. 2008, pp. 267 –274.

[8] L. Phifer, "A list of wireless network attacks," [Available Online] http://searchsecurity.techtarget.com/feature/A-list-of-wireless-network-attacks, 2009.

[9] H.-T. Yeh and H.-M. Sun, "Password authenticated key exchange protocols among diverse network domains," *Computers & amp; Electrical Engineering*, vol. 31, no. 3, pp. 175 – 189, 2005.

[10] J. N. Paulo Tam, "Protocol for ownership of physical objects in ubiquitous computing environments," in *IADIS international conference E-Society 2004*, 2004, pp. 614–621.

[11] J. Bohn, "Instant personalization and temporary ownership of handheld devices," in *Mobile Computing Systems and Applications, 2004. WMCSA 2004. Sixth IEEE Workshop on*, Dec. 2004, pp. 134 – 143.

[12] Z. C. Yongming Jin, Huiping Sun, "Hash-based tag ownership transfer protocol against traceability," in *IEEE International Conference on e-Business Engineering- 2009*, 2009, pp. 487–492.

[13] A. Alaraj, "Ownership transfer protocol," in *Internet Technology and Secured Transactions (ICITST), 2010 International Conference for*, nov. 2010, pp. 1 –6.

[14] M. R. A. Huth, *Secure Communicating Systems: Design, Analysis, and Implementation*, 1st ed. USA: Cambridge University Press, 2001.

[15] *Trusted Platform Module (TPM) Quick Reference Guide*, Intel Corporation, USA, 2007.

[16] J. Molina, H. Lee, S. Lee, and Z. Song, "A mobile trusted platform module (mtpm) architecture," [Available Online] www.flacp.fujitsulabs.com/ zsong/work/mtpm.pdf, 2012.

[17] K. R. C. Neuman, S. Hartman, "The kerberos network authentication service (v5)," [Available Online] http://www.ietf.org/rfc/rfc4120.txt, July 2005.

[18] "Hlpsl tutorial," [Available Online] avispa-project.org/package/tutorial.pdf, June 2006.

[19] A. A. et al, "The avispa project," [Available Online] http://www.avispa-project.org/, 2012.

[20] L. Vigano, "Automated security protocol analysis with the avispa tool," *Electronic Notes in Theoretical Computer Science*, p. 6186, 2006.

[21] W. Eddy, "Tcp syn flooding attacks and common mitigations," [Available Online] http://tools.ietf.org/html/rfc4987, Aug. 2007.