Home Network Security

Hans Scholten, Hylke van Dijk
University of Twente
Enschede, the Netherlands
{scholten@cs.utwente.nl, h.w.vandijk@eemcs.utwente.nl}

Abstract

Service discovery and secure and safe service usage are essential elements in the deployment of home and personal networks. Because no system administrator is present, setup and daily operation of such a network has to be automated as much as possible with a high degree of user friendliness. To achieve this goal many systems sacrifice security and privacy such, that services can be discovered and used unauthorized or a person's privacy may be breached. In this paper we present a security mechanism that seamlessly integrates with service discovery and usage. Exchange of keys and certificates is combined with messages used for service discovery. Services messages themselves are encrypted and authenticated, and casual receivers cannot read them. Although encryption and decryption of messages takes extra time, the combined protocol poses minimal communication overhead and hence can be used even in small devices.

1 Introduction

Advances in network technology enable the large scale deployment of always connected devices at low costs. In the not so distant future the Internet of Things [1] will give us the possibility to *"Google our shoes"* in the morning if we cannot find them immediately. Somewhere in the network a service exists that knows where your shoes are or knows where to search for them. This service only needs to be found by prospective clients, which is the task of a service discovery mechanism. It is clear that –automatic– service discovery is mandatory in a dynamic environment where no system administrator is present, our home being the perfect example.

A home network is not solely used anymore for connecting personal computers, but also to convey music and video, to switch lights, etc. Also, the home network isn't a single (e.g. IP based) network string anymore. It may be extended with other network segments like WLAN, Bluetooth, X10 home automation, sensor networks (e.g. ZigBee), etc. These new uses pose new challenges concerning security. Though a home network seems, and is considered as such by many, a closed non-open environment without the need for security other than at the perimeter (i.e. gateway and firewall), it is not. Some scenarios to illustrate this:

- A wireless network is not confined by the walls of an apartment, so neighbors can not only monitor what is going on, they might be able to intrude and use services they are not meant to, like switching lights.
- We may grant house guests access to web browsing through their own PDA, but restrict them otherwise.
- The remote control can present different functionality to its current user, or even restrict functionality, depending on who is using it. So parents are allowed to watch a different set of TV channels than their children.
- Some devices will be carried wherever the owner goes, e.g. mobile phone or PDA. When used at home, where they double as remote control, they will reveal themselves for the purpose to be discovered by the system. But once outside, they should be discrete or else others may infer information their owner wants to keep private, e.g. the presence at the location and time of a robbery. This is not as farfetched as it may sound: already the presence of GSM mobile phones in certain base cells is used as evidence in court. The use of short range radio makes localization even more fine grained. Or, in the Netherlands trials are taking place where people swallow smart pills to measure internal temperature. Without doubt this is essential to monitor one's medical condition (at home), but a bank could reject an application for a loan because the –secretly discovered– presence of such a pill may indicate a serious illness and be considered a risk.

From these examples it becomes clear that a home network is not open. Access to it should be controlled at all times, based on the current context. Context depends on the service being asked for, the device and person asking for the

service, location, time of day, etc. Secondly, every service discovery message can breach a person's privacy, even if it doesn't reveal this person's identity.

Service discovery is necessary, but it should be used with care, balancing ease of use and security and privacy. Secure service discovery protocols must ensure confidentiality and data integrity. Authentication and access control are methods to do so. This is not the case in service discovery protocols presently available. If they offer security at all, the imposed overhead prevents deployment in small unattended resource-lean devices. In most cases security is delegated to higher layers or to the application. Although at the application level this may ensure access control and encryption of data, at lower levels messages are exchanged freely, giving away information that should have been kept private.

This paper proposes a secure service protocol for home networks. Security is tightly coupled with service discovery. It uses messages exchanged by the existing service discovery protocol and so no extra messages are needed.

In the remainder of this paper we will give an overview of service discovery principles and protocols. After that the secure service discovery protocol is presented, followed by a discussion of future work and the conclusion.

2 Contributions

Service discovery and secured service usage are not new. After a service is discovered, layers higher in the network stack may ensure security by exchanging encrypted messages. However, the discovery process itself is open and receivers can monitor and analyze the nature of services on offer or those searched for. In addition, other information like the owner's identity and her whereabouts can be deduced. The security mechanism we propose seamlessly integrates with service discovery and service usage. The exchange of keys and certificates is combined with messages used for service discovery. Service discovery messages optionally can be encrypted and authenticated, and casual receivers cannot read or interpret them. A tamper proof trusted module embedded in the device provides certificates for authentication and does encryption. The public keys initially used in the message exchange are distributed by the home gateway to trusted devices only.

3 Service Discovery Overview

Service discovery may be considered the third generation of name discovery systems. ([2]). A name discovery system allows to discover objects in a distributed system. Properties of an object are stored as attributes associated with the name of the object. The first generation name discovery systems were real name services: name based queries return the attributes associated with the name. Directory services belong to the second generation: attribute based queries return the name associated with the attributes. Their use generally is limited to a static infrastructure of directories and servers in a computer based environment. The third generation name discovery –service discovery– accommodates dynamic environments that contain not only PCs, but all types of (consumer) devices and embedded systems. Here, service discovery should provide self-configuration and self-healing properties to the network with a minimum of manual administration and maintenance. In this context a service is a certain functionality offered by servers to users and applications. A service doesn't necessarily need to be associated with one device. It may migrate from one device to another, or it may be offered by a changing group of devices (as is often the case in wireless sensor networks).

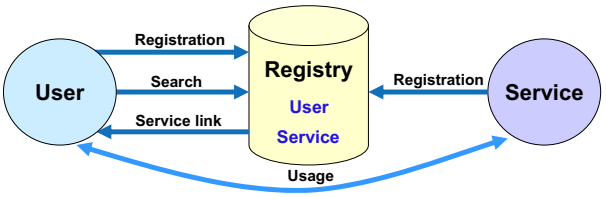

Figure 1. Registry based service discovery

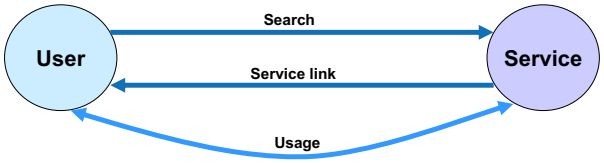

Figure 2. Non-registry based service discovery

3.1 Service discovery architecture

Service discovery comes in two basic architectures, registry (Fig. 1) and non-registry (Fig. 2) based. In the registry based architecture a registry (also: repository or directory) contains all information on users and services in the network. Every user and service has to register with the registry when it becomes active in the network (*"Registration"*). When a user wants to use a service it asks the registry if the service is available (*"Search"*). If the service is registered, the registry replies with a reference to the service (*"Service link"*). The user will access the requested

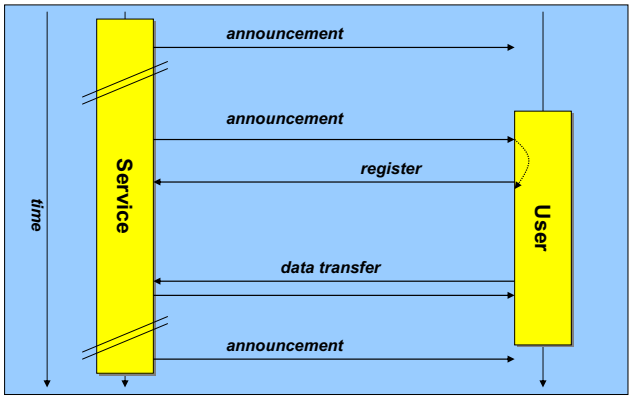

Figure 3. Active service - reactive user

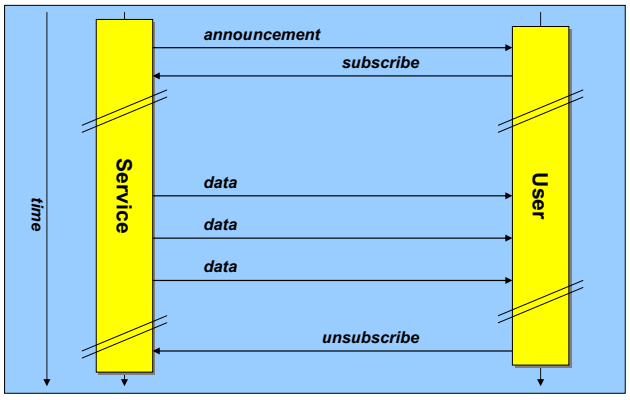

Figure 4. Subscription and notification

service directly via the given reference without any further intervention from the registry (*"Usage"*).

When no registry is present (in a non-registry based architecture), the user sends a broadcast to search for a service. If the service is available and it receives the broadcast, it will send the reference to the requested service to the user.

When a user becomes active in a registry based network, it has to know the location of the registry. One way to do so is to give the registry a fixed name or address known to all. But note that the registry is a service that can be searched for as in the non-registry based case. Both methods, sometimes combined, are applied in current service discovery systems.

3.2 Service discovery communication

The previous section introduced the two classes of service discovery architecture and their basic interaction. Actual communication is more complex, because all parties can be pro-active, reactive or both. A party is pro-active if it sends –unprovoked– messages, e.g. to announce its presence. It is reactive if it only responds to other party's announcements or requests. An example with a pro-active service and a reactive user is shown in figure 3. The service broadcasts its presence at regular intervals to the network. When a user enters the network, or becomes active in the network, it waits until it receives an announcement of the service it is looking for. Only then it will unicast, not broadcast, a message to the service. In the example the service is a registry to which the new user wants to register. Note that before the announcement the user did not know where to find the registry.

A service or user can be both pro-active and reactive. E.g. if a potential registry becomes active it can wait during a certain time for announcements from an already active registry. If it does not receive such an announcement it will declare itself the registry and announce its presence. This is a characteristic situation when a network is initialized or recovering from a failure.

Two phases can be distinguished in the interaction between services and users. The first one is the discovery phase just described, in the second phase the service is actually used. As in the discovery process, services and users can be pro-active as well as reactive. A pro-active user requests for a service every time it wants to use it, e.g. polling the temperature sensor in the room. In contrast to this polling of services is the subscribe/notify mechanism. After finding the service of its choice, the user subscribes to the service. Depending on the type of subscription, the service sends a notification to the user at regular intervals or when parameters change. E.g. the service sends the subscribed user the new value of the room temperature if it has changed. This is shown in figure 4.

Parties in a service discovery process can be pro-active, reactive, or both. When a party is pro-active it actively broadcasts its messages, announcing its presence, requesting services, etc. But even if a device is reactive it may give its presence away, because it is mandatory in some protocols to react to certain messages e.g. [3]. If such a device is associated with a person, indiscriminatively broadcasting and reacting on messages, the person's whereabouts can be deducted even if communications are encrypted. New service discovery systems therefore must not only ensure data confidentiality by means of encryption, they also must be prudent in their communication to ensure privacy, and actively control access to devices and services.

4 Secure Service Discovery

Security and protection of privacy are essential features of a modern service discovery system. They are nevertheless lacking in many service discovery protocols in use today. Systems or protocols which have security features include Jini [4], UPnP [3] and Bluetooth [5]. However these

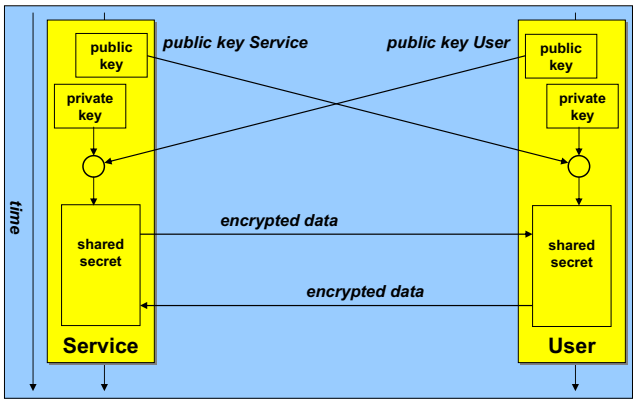

Figure 5. Diffie-Hellman key exchange

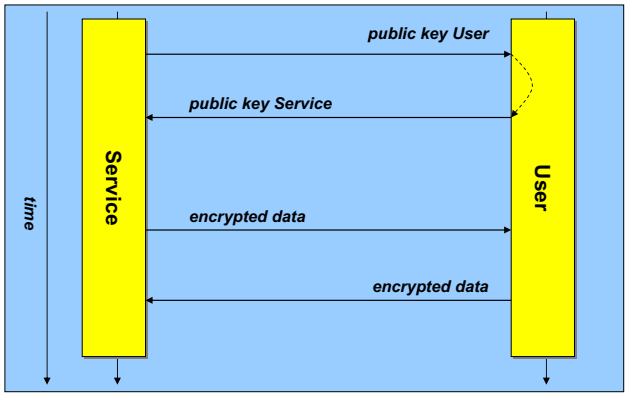

Figure 6. One party initiated Diffie-Hellman key exchange

are not very lightweight and suitable for resource-lean devices at home, or are tightly coupled to one network technology. In the following we will show how the Diffie-Hellman key exchange [6] can be mapped onto and tightly coupled with the service discovery protocol and subsequent usage of services.

4.1 Diffie-Hellman Key Exchange

The Diffie-Hellman key exchange is a public key cryptographic system in use in well known protocols like SSH and SSL. It allows two parties to anonymously establish a shared key. The key exchange is executed whenever there is a need to secure an outgoing message and there is no security context available between the two communicating parties. Figure 5 shows the sequence of messages. First of all both Service and User will publicize their public key. Based on each other's public key and their own private key Service and User calculate a shared secret. Note that the calculated

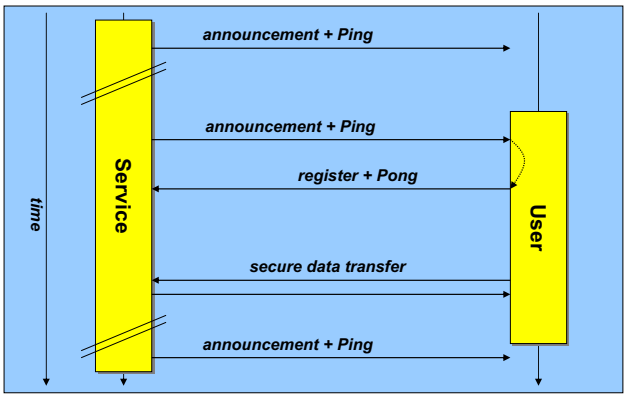

Figure 7. Service discovery combined with Diffie-Hellman

shared secrets for both parties are the same. From that moment on the shared secret can be used to send encrypted messages to the other party. Because the use of asymmetric keys is a slow process, often the first encrypted message is used to exchange a faster symmetric key to be used in the remainder of the session.

This original form of Diffie-Hellman is vulnerable to man-in-the-middle attacks. Therefore certificates can be included in the messages to certify that the used public keys are indeed genuine. The authenticated variant of the Diffie-Hellman key exchange is known as the Station-to-Station protocol. It establishes a shared key with mutual authentication of the parties and mutual explicit key authentication

We did not elaborate on how public keys are disseminated. They could be broadcasted, or sent to a central authority, where they are available for others on request. In a home network the obvious place to store public keys etc. would be some central server, e.g. the residential gateway. If we assume that one party wants to communicate with an other, the sequence of messages could be as depicted in figure 6. Service initiates a session by sending its public key to User. This message is referred to as the "Ping" message. User then responds by sending its own public key (possibly encrypted with the previously received Service's public key), also known as the "Pong" message. From now on Service and User can exchange data encrypted with the common shared secret ("Pung").

4.2 Service Discovery with Diffie-Hellman

Comparing figure 6 with figure 3 shows that the service discovery message sequence is very similar to the Diffie-Hellman key exchange. Figure 7 demonstrates that both protocols can be combined indeed. The announcement message from Service is combined with the Diffie-Hellman

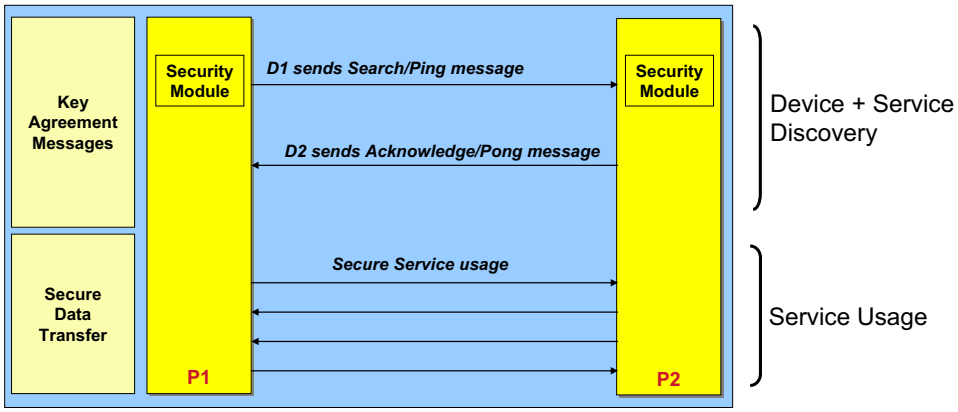

Figure 8. Secure service discovery with security module

Ping message, while the Pong message is combined with the register message from User. Combining service discovery messages with security messages is not only valid for announcement and registration, but for all other service discovery messages as well.

The security features heavily rely on the trustworthiness of the proofs of registration and the confidentiality of cryptographic key material. Devices should therefore be equipped with a temper-proof security module (e.g. smartcard) that handles all security-critical operations. The security module also represents the identity of the device in which it is installed (Figure 8).

In order to facilitate the zero-configuration of devices, the device manufacturer that produces devices with a hardware security module, will initialize the security module with its initial credentials, i.e. the manufacturer will initialize the signing key pair of the security module, and will make sure that the device has a valid certificate. The service discovery protocol precedes any other communication in the system and therefore all security parameters must be initialized during discovery of new devices and services. The initial key agreement ”piggybacks” the message exchanges of the service discovery protocol. This means that two devices, P1 and P2 perform the key agreement protocol as part of the service discovery protocol, thus making the service discovery protocol a secure service discovery protocol.

During the secure service discovery process six distinct actions can be distinguished:

1. Send Ping: Device P1 sends out an initial message, the Ping message. This Ping message includes two parts: a message part which will contain service discovery information and cryptographic key material which will allow the receiver of the Ping message to establish a common secret. Next to the message part, the Ping message contains an authenticity proof. This proof protects the integrity of the Ping message, and enables its receiver to verify that the sender of the Ping message is indeed P1.

2. Receive Ping: The receiver of the Ping message, device P2, verifies the authenticity of the message, and processes its message part.

3. Send Pong: If P2 decides to produce a response to the message part, it has the possibility to compute a cryptographic key K which P2 will share with P1 as soon as P1 has processed P2’s Pong message. The Pong message is a reply message to the Ping message.

4. Receive Pong: If P1 receives the Pong message, it first verifies the authenticity of the Pong message. If the Pong message comes from a device that P1 trusts, or which it is allowed to communicate with, P1 can proceed. Once the information has correctly been decrypted, P1 can proceed with the service discovery protocol.

5. Send Data: If the service discovery protocol necessitates a second message sent from P1 to P2, P1 uses the shared secret to derive encryption and integrity-protection keys to protect the confidentiality and integrity of the information, respectively. If the confidentiality of the information is to be protected, P1 encrypts the information for P2, and authenticates it with the freshly calculated integrity-protection key. This information is then sent to P2.

6. Receive Data: If P2 receives an encrypted and/or authenticated message, it retrieves the correct shared key, derives from this key the decryption and integrity-protection keys, and uses them to validate the authenticity of the incoming message, and to decrypt its payload.

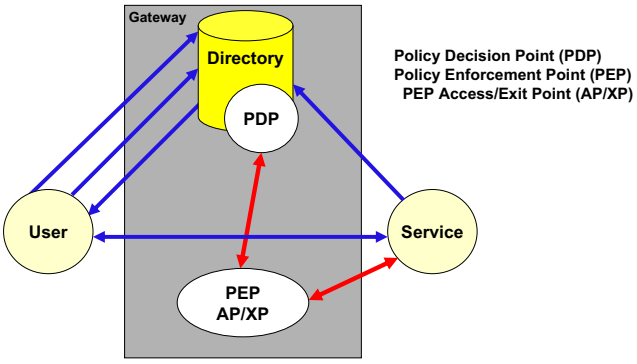

Figure 9. Service discovery with policy management

After receiving the Ping, P2 may decide not to react and thus not to send the Pong. This decision can vary over time, depending on context, even with the same devices P1 and P2. E.g. P1 and P2 are a TV and a remote control belonging to the same household. If the remote is used by the parents it will allow other TV channels than when it is used by one of the children. (P2 "knows" who is using it because a PIN is used, or some biometric characteristic.)

When a newly acquired device is brought into the house for the first time it is still untrusted and will not work. Only after learning each other's identity and setting –default– policies, the house will trust the new device and the device will trust the house. This process of "imprinting" is a (manual) one-time action –"Touch-and-Go". If after some time the device is removed from the house (e.g. sold) the imprinting needs to be reversed. The device will become untrusted again and can be imprinted once more by an other house. This way of authentication is the Resurrecting Duckling policy described by Stajano and Anderson [7].

Figure 9 is just one possible implementation of access control for a typical registry-based home network, other configurations are possible as well. P1's and P2's actions are checked and enforced by an access control mechanism based on the current set of policies. The access control mechanism consists of two parts: the Policy Decision Point (PDP) and the Policy Enforcement Point (PEP). In the example the registry, PDP and PEP are situated in the same residential gateway. Both devices (User and Service) are imprinted and registered at the registry. When User wants to use a certain service it will ask the registry for a service. If such a service exists the registry will check with the PDP whether User is allowed to use the service. If so, it will send User a link to the service. If not, it will reply that such a service is not available. After receiving the link to the service (located at Service), User sends a request for the service to Service. There, the usage of the service is controlled by the PEP, who will check current policies at the PDP.

4.3 Implementation

The secure service protocol is developed as part of the TEAHA platform [8] [9]. TEAHA implements a heterogeneous registry based home network. It allows interoperability between different types of network. Secure service discovery is one of the main features of the system.

Experiments showed that performance of the secure service discovery protocol is mainly determined by the performance of the security engine. It is the security engine that stores credentials. It signs and encrypts messages on request. A part of the secure engine is a tamper proof module. In our experiments we used an integrated USB smart-card and reader for this purpose (Gem eSeal @ 5 MHz.). The initial connection to the card has a typical delay of 400 ms. Reading credentials (8 Bytes) from the card takes about 22 ms., while 3-DES encrypting of 64 bits takes 35 ms, excluding the data transfer. RSA signature generation is around 250 ms.

Some delays seem to be long (initial connection: 400 ms.; RSA signature generation: 250 ms.). However, these actions are only necessary sporadically. For instance, RSA signature generation is only required during session set up and service registration. During a session the overhead of encrypting, decrypting and checking credentials is only small. Because the security messages piggyback service discovery messages, these messages need to be extended with some extra data fields. The overhead of sending these slightly longer messages is negligible.

5 Conclusion

In this paper we have shown that home networks are becoming more and more vulnerable for attacks from outside. And if taken outside the home, devices may be a threat to the privacy of the person who is carrying them. The majority of current service discovery systems do not implement security measures at all, or only in a limited way. They may restrict access but will still expose their presence by sending service discovery related messages.

We have introduced a secure service discovery system, based on the authenticated Diffie-Hellman key exchange, security modules and imprinting. It will ensure device authentication, data authentication and data confidentiality. Policy decision points and policy enforcement points control access to devices and usage of services. Also they control when and how to use and react to service discovery messages, thus protecting the privacy of the person who is using a device.

The secure discovery protocol and access control are implemented in the TEAHA platform. PDPs and PEPs are

fully functional, however a full policy management system is lacking and will be subject of future research. Policies in the current implementation are simple rules set manually.

6 Acknowledgements

We thank the members of the TEAHA project, specifically Danny de Cock from Katholieke Universiteit Leuven, for their comments on the first drafts of this paper.

This work is sponsored in part by the European Commission (IST-507029 priority 2.3.1.8)

References

[1] "The internet of things," http://video.google.com/videoplay?docid=-3857739359956666768&pr=goog-sl.

[2] V. Sundramoorthy, "At home in service discovery," Ph.D. dissertation, Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede, Netherlands, September 2006.

[3] *UPnP™ Device Architecture 1.0*, UPnP™ Forum, Dec. 2003, http://www.upnp.org/.

[4] *Jini™ Technology Core Platform Specification, version 2.0*, Sun microsystems, June 2003, http://wwws.sun.com/software/jini/specs/.

[5] "Bluetooth specification, version 1.1," http://www.bluetooth.org/.

[6] W. Diffie, P. C. V. Oorschot, and M. J. Wiener, "Authentication and authenticated key exchanges," in *Springer Science+Business Media B.V*, vol. Volume 2, Number 2, 1992, pp. 107–125.

[7] F. Stajano and R. Anderson, "The resurrecting duckling: Security issues for ad-hoc wireless networks," pp. Springer–Verlag London, UK, 172–194, 1999.

[8] "Teaha web site," http://www.teaha.org.

[9] H. W. van Dijk, J. Scholten, A. Tobalina, V. G. M. noz, S. Milanini, and A. Kung, "Open home networks: the teaha approach," in *Sixth International Conference on Networks (ICN2007), Sainte-Luce, France*, C. Dini, Z. Smekal, E. Lochin, and P. Verma, Eds. Piscataway: IEEE Computer Society Press, April 2007, p. 053.