
*Intelligent Automation and Soft Computing*, Vol. 16, No. 3, pp. 399-418, 2010


# DEVICE AUTHENTICATION IN WIRELESS AND PERVASIVE ENVIRONMENTS

**[1]GEORGIOS KAMBOURAKIS, [1]STEFANOS GRITZALIS AND [2]JONG HYUK PARK**

*[1]Laboratory of Information and Communication Systems Security*
*Department of Information and Communication Systems Engineering*
*University of the Aegean, GR-83200 Samos, Greece*
*{gkamb, sgritz}@aegean.gr*

*[2]Department of Computer Science and Engineering*
*Seoul National University of Technology, Korea*
*E-mail: parkjonghyuk1@hotmail.com*

ABSTRACT—Security can only be guaranteed as long as the hardware and other key parameters, including software components, secret keys etc, of a device remain genuine and unmodified. Under this context, device authentication must be considered as a key security issue, complementary and of equal importance to user authentication, in today's wireless and forthcoming ubiquitous realms. This paper classifies and analyses possible major solutions proposed until now towards solving the device authentication issue. We constructively argue on each solution presented examining its advantages and disadvantages. A qualitative comparative analysis for the device authentication schemes in question is also offered, probing its applicability for both infrastructure and ad-hoc deployments. Inter-domain device authentication, where applicable, and users' privacy as a side-effect are investigated as well.

unauthorized device is difficult to identify, locate and repel, when on the move, in an emergency situation.

In this paper we survey all major potential solutions and trends to the device authentication issue and examine its pros and cons. Each option is further analyzed and compared with the others based on some indicative qualitative criteria giving a comprehensive view about its applicability and robustness in terms of security. User's privacy as a complementary issue to device authentication is investigated as well.

The remainder of the paper is structured as follows: next section elaborates on several aspects of the device authentication problem highlighting its importance for today's wireless networks. Section 3 identifies and analyses possible solutions to the device authentication problem so far. Section 4 discusses privacy concerns associated with device authentication, while section 5 gives a qualitative analysis for the device authentication schemes in question. Last section draws a conclusion.

## 2. PROBLEM STATEMENT

Generally, entity authentication can be defined as the process whereby one party is assured of the identity of a second party involved in a protocol, and that the second has actually participated [1],[2],[3]. Identification and authentication are sometimes confused with one another. So, while Identification is the means by which an entity provides a claimed identity to the system, authentication is the means of establishing the validity of this claim. Simply put, an authentication protocol involves a claimant A and a verifier B. The claimant should provide certificatory evidence to the verifier about its purported identity. The aim is to confirm that the true identity of the claimant is as declared (i.e., A). In this context, an authentication protocol should guarantee that: (a) in the case of truthful parties A and B, A is able to successfully complete the protocol and authenticate itself to B, (b) B cannot reuse an authentication exchange with A so as to successfully impersonate A to a third party. This is referred to as *transferability*, (c) the probability that any third party, carrying out the protocol and playing the role of A, can cause B to complete and accept A's identity is negligible. This is also known as *impersonation*.

In the context of this work we specifically define device authentication as the entity authentication in which the objective is to securely identify and further authenticate the identity of a physical device, possibly at a specific location. In most cases the authentication procedure must also guarantee the integrity of the device in terms of hardware and/or software.

In practice, while device authentication is most easily exercised, it can be forged in many ways, as unique device characteristics can be easily copied. Currently, the most usual practice to protect IEEE 802.11 networks against unauthorized access is to perform device authentication by maintaining a list of "white" MAC addresses that are allowed to access the network. However, today, this solution is considered ineffective as the majority of end-user devices allow the user to configure its MAC address at will. As a result, after surveillance, the attacker can modify the MAC address of his rogue AP to match that of an existing authorized device and connect to the network without detection.

Bluetooth specification is another example of weak device authentication. Actually, within the present Bluetooth specification there is only device authentication. Bluetooth devices use the PIN codes for device authentication where the PIN acts as a variable in the initialization key generation process. As these codes are necessary for authentication and link security, users should ensure that Bluetooth devices use PIN codes other than the default, or lowest, setting. On the other hand, unauthorised users can easily gain access to mislaid or stolen devices. If these Bluetooth devices have been paired sometime in the past, they now have access to any device in each of the Piconets the device has previously associated with.

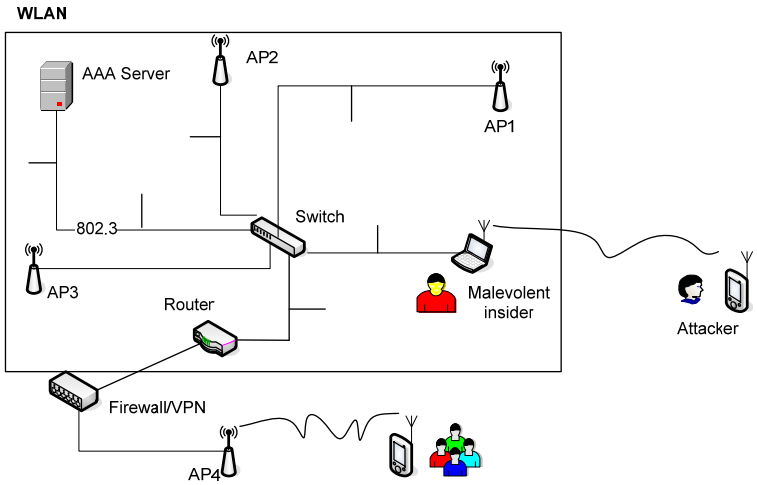



**Figure 1. Rogue device scenario (insider attack)**

Whether this sort of attack is most common to infrastructure IEEE 802.11 networks, similar problems may easily arise to MANETs, WSNs and even Radio Frequency Identification (RFID) tags, where a rogue or even a compromised or cloned device can be fatal for the overall network trustworthiness. For instance, at present most RFID devices promiscuously broadcast their static identifier with no explicit authentication procedure. This gives the opportunity to attackers to passively scan identifying data performing a *skimming attack*. Additionally, skimmed data may be used to fabricate cloned tags, thus giving more opportunities to attackers. In a *swapping attack*, for example, the adversary fabricates cloned tags, seals them inside a decoy container, and quickly swaps the fake container with the original one. Having the ability to clone a tag and prepare the decoy in advance, the adversary is able to carry out the physical swap very quickly.

Furthermore, it is well known that erratic behaviours in sensors networks seeking physical access to sensor devices are in most cases difficult to be repelled due to the anonymous and (semi)uncontrolled wireless terrain. At best, physical access to a certain sensor enables the aggressor to acquire sensor's secret keys. According to [4] a competent attacker equipped with a laptop is able to retrieve sensor keys in less than a minute given that he has physical access to it. The experiments were performed utilizing sensors integrated with the commonly used Mica family designs and especially with Mica2. For instance, CodeBlue medical framework sensors employ the Chipcon CC2420 chip, which as cited in [5] is able to support only two secret keys. Once these keys are compromised the attacker has access to the communications of the entire network.

In all cases, the heart of the problem is the lack of any mutual device-to-device authentication procedure or mechanism when a certain device attempts to join the network. Also, there are many cases where an identified device may not be allowed on the network; for example, if it was reported as stolen, the metadata in the device identity or policy store would indicate that it should not be allowed.

Device authentication mechanisms enable an organization or system in general to manage both users and devices. As described in Figure 2 it is considered as a second layer of authentication, ensuring that only specific authorized devices operated by authorized users can access the organization's network. Separately, neither one can have access. This means that even

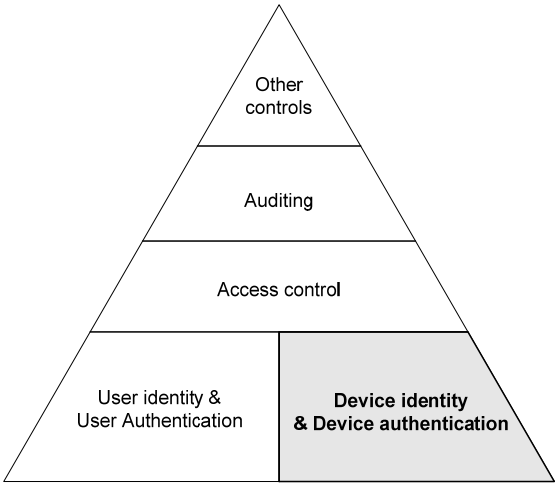



**Figure 2. Authentication & access control pyramid**

in case passwords, credentials or tokens are stolen or compromised, the network will still be well protected as long as the authorized device is not employed. It also assures that private data stored across network resources are never exposed because unauthorized devices cannot access the network, even when operated by an authorized person. Moreover, in case of infrastructure devices, e.g., APs, switches etc and other hardware that is not operated directly by humans, like sensors, device authentication can guarantee to a great deal that a device is genuine and has not been somehow compromised. Therefore, device authentication effectively enforces network access control policies in a proactive manner, that is, before they connect to the network. When applied properly, device authentication can provide the following benefits, thus reinforcing the overall network security:

(a) Adds another layer of protection to the organization's defense strategy. Device authentication allows only authorized users, employing previously enrolled devices, to enter the network and access its resources. This permits organizations or even ad-hoc deployments to effectively synchronize their user and device security policies.
(b) Consolidates the secure authentication of authorized hardware and other remote devices trying to access the network into a comprehensive security plan. Therefore, device authentication is particularly effective for securing remote access by mobile users who must access the network through a Virtual Private Network (VPN) or other remote connection.
(c) Device authentication can be a key enabling technology for e-government and other agency applications where the control of the device, in conjunction with the control of the user, is an important security concern.
(d) In foreseeable highly pervasive environments device authentication would substantially increase user's trust in the provided services. Actually, the main goal of researchers working in this field is to create a system that is pervasively and unobtrusively embedded in the environment, completely connected, intuitive, effortlessly portable, and constantly available. As a result, smooth device authentication in such environments, which are dealing primarily with machine-to-machine rather with human-to-machine model, can offer secure entity interactions, thus boosting overall system's trustworthiness.

## 3. IDENTIFICATION OF POSSIBLE SOLUTIONS

### *3.1 The IEEE 802.1X Framework*

With the advent of the IEEE 802.11i specification [6] the 802.1X [7] framework provides various Extensible Authentication Protocol (EAP)-based [8] and certificate-oriented mechanisms that can be employed both for user as well as for device authentication. Towards this direction every device must afford a device certificate bound to it to be able to prove its identity prior acquiring an IP address and joining the network. The uniqueness of each network device can be determined by a combination of its hardware and software characteristics. For example, hardware parameters may be the device's serial number, hard disk or other components serial codes and manufacturer identities, MAC address, processor type, memory capacity etc, while as software parameters may use a hash of some driver codes, start/end memory address of software portions stored in ROM and other similar attributes. A careful choice of this kind of characteristics is enough to uniquely identify and further authenticate each network device even those of the same model and type. Note however that these attributes must be static in the long run as they comprise the identity of each particular device.

Once a collection of such parameters has been decided, e.g., by the network operator, a hash of the concatenated sequence *(charact_1||charac_2||…||charact_n)* is calculated to serve as the mid or long-term identity of the device. Also, for the sake of security, a keyed-MAC (hash) can be calculated using a corresponding HMAC function. As a result, a device certificate must bind a combination, say a hash of various physical properties of the device (MAC address, serial number, driver versions etc), to a private key in the form of a X.509 certificate. After that, device-to-device authentication can be effectively exercised utilizing EAP methods (EAP-TLS, EAP-TTLS, PEAP etc), before any user authentication takes place. It is stressed that the private key of the device must be stored securely in the device in the form of a tamper resistant memory, therefore not accessible by human users or applications. By this scheme, the authentication server can utilize the same identity certificate that is always used when being authenticated by other network nodes.

However, at least for IEEE 802.11 infrastructure mode, 802.1X-based device authentication mandates several modifications concerning the current communication procedures between the AP and the authentication (usually RADIUS) server. Specifically, all APs must act as supplicants when booting-up (before acquiring an IP address) to be able to be authenticated as devices to the corresponding RADIUS server [9]. Moreover, all network devices, including APs, must support e.g., EAP-TLS functionality to support certificate based authentication at the data link layer. In addition, a well-defined and scalable (re)keying mechanism between the AP and the authentication server to encrypt the traffic between them must be somehow automated and not rely on administrators to configure it manually. This is especially true for remote network devices. Currently however, no standard automated session key derivation procedure between an AP and the authentication server exists.

Furthermore, to thwart clever attackers any solution applied must support periodic re-authentication at regular intervals, thus ensuring session freshness. Consider for example the following scenario in the 802.1X architecture: A legitimate AP acting as a supplicant authenticates itself to the responsible network authentication server through a network switch, which in this case implements the role of an authenticator in the 802.1X model. After successful authentication the switch opens the corresponding port for the AP, which is allowed to join the network and allocate a valid IP. Sometime later an insider brings a rogue AP, forges its MAC address to be identical to the MAC address of the legitimate AP and connects it to the network replacing the latter. In the absence of any re-authentication mechanism the switch has no means to detect the replacement, since its role is limited to transfer packets with a valid MAC address from one

network segment to the other and vice versa. Nevertheless, periodic session validation may presume the derivation of a session key between the involved devices during initial device authentication phase. After that, it is not possible to substitute a legitimate device, since the rogue one does not know the current session parameters, including the key.

Apart from all previously discussed issues the 802.1X approach:

(a) cannot straightforwardly be accommodated to ad-hoc network configurations as it requires infrastructure mode,
(b) mandates some sort of Public Key Infrastructure (PKI) and some rather sophisticated and maybe costly hardware and software components to be implemented,
(c) in most cases requires expensive public key operations and protocols that lightweight mobile devices is difficult to afford.

Therefore, it is only appropriate for medium to large organizations rather than for Small Office/Home Office (SOHO) environments, MANETs, or WSNs. Concluding this subsection, we can say that 802.1X-oriented device authentication, if refined and standardized sometime in the future, can provide a promising avenue towards solving the device authentication problem.

### *3.2 The 802.16 Case*

Device authentication through corresponding device (manufacturer) certificates is already part of the IEEE 802.16 standard, namely the Privacy Key Management (PKM) protocol [10]. The PKM RSA authentication protocol employs X.509 digital certificates and the RSA public key encryption algorithm that binds public RSA encryption keys to MAC addresses of MSs. Under this context, a Base Station (BS) authenticates a client Mobile Station (MS) during the initial authorization exchange. Each MS must incorporate a unique X.509 digital certificate issued by the MS's manufacturer. The digital certificate among other contains the MS's Public Key and serial number and the MS's MAC address.

When requesting an Authorization Key (AK), an MS presents its X.509 certificate to the BS. Upon reception, the BS verifies the MS's certificate, and then uses the public key that it contains to encrypt an AK, which then sends back to the corresponding MS. Under this scheme MAC spoofing attacks can be effectively repelled considering that only the legitimate MS device has the matching private key to decrypt AK and join the network. Briefly, the specification mandates that all MSs using RSA authentication shall have factory-installed RSA private/public key pairs or provide an internal algorithm to generate such key pairs dynamically. All MSs with factory-installed RSA key pairs shall also have factory-installed X.509 certificates. All MSs that rely on internal algorithms to generate an RSA key pair must offer a mechanism for installing a manufacturer-issued X.509 certificate after key generation. For mutual authentication each BS is also equipped with a digital certificate that binds its hardware characteristics with the corresponding public key as described hereunder.

*countryName=<Country of Operation>*
*organizationName=< Name of Infrastructure Operator>*
*organizationalUnitName=<WirelessMAN>*
*commonName=<Serial Number>*
*commonName=<BS Id e.g., 00:60:21:A5:0A:23>*

Note that the newest PKM version 2 protocol specification [10] supports 802.1X/EAP authentication too. This is of course a movement towards providing a unified 802.11/802.16 authentication framework, but in our case device authentication services to heterogeneous 802.11/802.16 contexts may also be applied as discussed earlier in the previous subsection. Generally, the PKM's authentication protocol establishes a shared secret (AK) between the MS and the BS. The shared secret is then used to secure subsequent PKM exchanges of temporary keys. PKM also supports periodic re-authentication / re-authorization and key refresh. Although,

the 802.16 approach is effective as far as the device authentication problem is concerned, it suffers from the same problems discussed in section 2.1. Therefore, it cannot straightforwardly be applied to low-cost pervasive devices like sensors and RFID tags.

### *3.3 The Trusted Computing Solution*

A different hardware oriented solution towards solving the device authentication problem has been examined by the means of trusted computing. Considering this option a number of hardware and software manufacturers have cooperated forming the non-profit Trusted Computing Group (TCG) (https://www.trustedcomputinggroup.org). The main aim of TCG is to develop trusted platforms by utilizing Trusted Platform Module (TPM) chips and novel hardware architectures. The TPM chip [11], also referred to as the "Fritz chip," is responsible for a number of basic functions including integrity measurement, integrity storage and integrity reporting of all critical events occurring in the trusted platform. This chip can be either embedded in a smartcard or dongle soldered onto the motherboard or will be integrated in the main processor. The latter approach offers better security because the data is not transferred on motherboard buses between the TPM and the CPU. Very recently, TCG formed the Mobile Phone Work Group focusing on the adoption of TCG concepts for mobile devices [12]. This work group will enhance TCG as needed to address specific features of mobile devices like their connectivity and limited capability.

The specification defined by the TCG [13] states that Trusted Platforms (TPs) are computing platforms that add to themselves the property of trust. In other words, they provide proper mechanisms to verify, in a secure way, that the data yielded by them is not tampered with. When a manipulation is performed, a security discrepancy is detected and reported to the user who will decide whether or not to trust the data provided by the TPs. More specifically, on booting up, the TPM takes over inspecting the integrity of boot ROM, then loading and executing it, and finally, verifying the overall system's state. It then verifies the first portion of the operating system, loads and executes it, and again attests the system's state. This procedure repeats several times for all protected software modules which in the end are loaded and become available to the system upon booting up.

Moreover, the TCG-enabled system preserves and maintains a list of approved hardware and software components. For each of them, the system must confirm whether it is approved and not revoked and whether it is digitally signed in case of software. Meanwhile, e.g., in case that some components have been upgraded and therefore the system's configuration has changed, it must go online to be recertified. The goal of this verification process is a system that has booted into an acknowledged and well-defined state and all its components either hardware or software are certified.

In a nutshell, the trusted computing approach can contribute to the following:

(a) platform integrity: Ensures that the device is running authorized software and hardware. This provides the user with warranties that they are using a secure device,
(b) device authentication: Ensures that the correct, in terms of authorisation and authentication, device and user is accessing the appropriate services,
(c) secure software download: Allows users to securely download application software, updates, firmware etc and to verify that those downloads are genuine,
(d) secure channel between device and subscriber identity modules, like Subscriber Identification Module (SIM) and UMTS Integrated Circuit Card (UICC) cards used in GSM and UMTS networks: The intention here is to provide users with secure channel transactions for data being transferred between the device and the UICC card,
(e) software use: Ensures that the applications on a device do exactly what they are intended to do e.g., they do not access data they are not supposed to,

(f) proving platform and application integrity: A method to ensure that the user is aware that they are using a secure device and verified applications which gives them a feeling of security and trustworthiness and
(g) user data protection and privacy: Protect user data (e.g., contacts / address book, electronic wallets, identity etc).

In this context, trusted computing can contribute a great deal to the vision of the "self authenticated, self protecting network" where every wireless or wired network entity that contains a TPM is self and cross authenticated before entering the network. As a result, rogue components either hardware or software can be repelled from joining the network. Nevertheless, currently the level of security provided by TPM modules highly depends on the details of design and implementation, which are not clear yet for almost all trusted computing manufacturers. Moreover, the TCG specifications has to cover some distance until it reaches a mature state and proved to be secure and trustworthy enough (not simply trusted) in the long run [14],[15],[16].

### *3.4 Other Approaches*

In this subsection we briefly survey other research works dealing either diametrically or partly with device authentication.

#### *3.4.1 Smart Cards*

In NIST report 7206 [17] the authors employ smart cards to support user and mobile devices authentication. They state that smart card authentication is perhaps the best-known example of a proof by possession mechanism when compared to other more traditional categories of authentication, including proof by ability (to do something) proof by knowledge (e.g. passwords) and proof by property (e.g. fingerprints). Towards this direction the report provides an overview of two novel types of smart card that use standard interfaces supported by most handheld devices. Without doubt, when used for user authentication, smart cards can improve the security of a device and provide additional security services too. Device authentication can also be seconded considering that it is generally more difficult to operate a rogue (compromised or stolen legitimate) device without the proper smart card. On the other hand, cloning an existing device and its matching smart card is not exactly an easy task for the attacker to accomplish.

On the contrary though, standard size smart cards are generally not suitable for handheld devices due to the relatively large size of the card, the need for a proper card reader, and the difficulty and cumbersomeness of embedding a reader to the device. Putting aside these obstacles, by e.g., utilizing interfaces found today in most smart card readers (as in the aforementioned report), smart card authentication may prove very profitable. Some difficulties remain however including the increased acquisition and administrative cost for the users and the organizations themselves and the fact that this solution is not suitable for small wireless devices like sensors and RFIDs.

#### *3.4.2 Location-based Access Control*

In another work [18] that partly deals with device authentication the authors examine location-based access control mechanisms. They propose a new protocol for location verification, called the *Echo protocol* and they prove its security. Location verification enables location-based access control. This means that a person carrying a specific device can be granted access to particular resources only if his location has been confirmed by employing a corresponding protocol. Naturally, when this approach is combined with physical security e.g., who's entering the building, then location verification can be used to allow wireless access to all those inside.

It is true that location-based access control has several pros. Among others, it is natural for various applications. While one simple security policy might permit wireless access of only the

printers installed in the office you are in, on the other hand might force that a wireless device must cease operating if it is detected operating outside the company building or being moved to another room. By this means, stolen, compromised or rogue devices not operated in certain premises, where they are supposed to operate according to the current policy, will be proved useless to malevolent individuals. Though, while location-based access control in human terms is straightforward, e.g., turning on the TV set in a particular room needs to have a physical presence in the room, achieving the same kind of guarantee with wireless networks, is not so easy. Location-based access control policies on networks and information resources by extension, requires a method to perform location verification, where an entity's location is securely verified to meet certain criteria: e.g., being inside a particular room.

In practice, while this approach may be effective if implemented properly (guarantee in-region verification for a high rate of legitimate location claims), requires significant administrative costs in terms of configuring and maintaining proper and strict policies for every network entity involved. On the top of that, as with 802.1X, location-based access control adapts better with infrastructure wireless networks having some sort of administrative authority to define policies rather than ad-hoc pervasive mode and nomadic computing.

*3.4.3 Electromagnetic Signatures*

A different approach that examines the feasibility of identifying wireless nodes in a network by measuring distinctive electromagnetic characteristics or "signatures" of Wireless Local Area Network (WLAN) cards is presented in [19],[20]. There the authors focus and perform preliminary experiments with IEEE 802.11 compatible cards but their conclusions can be applicable to other wireless technologies as well. Their idea originates from the remark that the physical layer of 802.11 wireless communications cannot effectively protect the identities of the communication endpoints. Thus, although the user's identity and privacy may be well protected at network-layer, they are still in danger and can be disclosed at the physical layer.

Specifically, any electromagnetic signal transmitted over the air can be passively or actively monitored, captured and analyzed at will by any properly equipped adversary located within the wireless device's transmission range. This physical layer "vulnerability" is also under investigation by several researchers in the context of the so called *template attacks*. Therefore, users' anonymity and privacy can be in danger if their device can be uniquely identified, through the measurement of distinctive radio-frequency electrical characteristics or electromagnetic signatures that it emits. The attacker's aim in this case is to correctly relate a received electromagnetic emission with a specific transmitter (device). This is possible because the electromagnetic signature unique characteristics stem from discrepancies in circuit and antenna topology from manufacturer to manufacturer and from variability in circuit performance linked to manufacturing tolerances. At frequencies, such as 2.4 GHz or 5.2 GHz, used in 802.11 networks even minor component variations in a transmitting circuit may result to a significant effect on the emitted signal. Given that we are able to detect and record distinctive electromagnetic signatures, a wireless device and its user can not only be monitored, but when combined with visual identification, can also be identified and tracked.

Due to these qualities, devices' electromagnetic emissions are worth being further investigated in the context of effective device authentication. Rogue, compromised and even cloned devices can be differentiated from the legitimate ones through their electromagnetic signature that they emit. However, this must be proven so, not only in sporadic experiments, but also in large scale, where many types and access technologies of wireless devices are employed. On the other hand, device authentication based on this scheme may be practical in corporate networks - by constructing beforehand a database of all authorized devices' electromagnetic

signatures (metadata describing the asset) and putting it in a corresponding authentication server - but seems rather unpractical for ad-hoc deployments.

*3.4.4 HB+ style AKE protocols*

The last category of solutions attempts to tackle the device authentication problem indirectly. Specifically, all AKE Authenticated Key Exchange (AKE) protocols proposed in the literature can serve as device authenticators especially for autonomous, low power computing devices like sensors and RFIDs. The operation of such devices is not user centric, thus in a sense, device authentication coincides with user authentication.

As referred in [21], an AKE protocol must satisfy two requirements. Firstly, support mutual authentication. That is, two communication entities can authenticate the identities of each other; this is specific to the device authentication issue. Secondly, guarantee secure communication between the communication entities via the establishment of a session key for securing later transmissions. AKE protocols can be classified into three major categories. The first one requires PKI while the second is based on symmetric key architecture. The last category is based on hybrid-key architecture, where the server holds a pair of matching public/private keys and the client shares a secret with the server.

However not all of the AKE protocols categories mentioned above are suitable for low power devices because they do not consider the computational cost in terms of service time and power consumption on the client side. For example, the client device may need over 1 min to execute a modular exponentiation. Moreover, keep in mind that the authentication process in an AKE mechanism is not based on unique hardware characteristics of the device itself but rather on some sort of (pre)installed credentials or tokens. This means that once the device becomes compromised (the credentials leak and being transferred to another device) or hardware modified will go undetected. For this reason in this group of solutions we should consider only protocols that require extremely low processing power and have been designed for tiny wireless devices.

Under this context, in the following, we only refer to the work been proposed in [22] and redefined later in [23] under a three party (proxy assisted) setting. The authors analyze a particular human-to-computer authentication protocol designed by Hopper and Blum (HB), and demonstrate by using RFID tags that it is practical for authenticating low-cost pervasive devices as well. The outcome of their work is a new symmetric authentication protocol, namely *HB+* that is appropriate to securely identify and authenticate wireless devices with limited power and processing capabilities. The motivation here is that low-end RFID tags and other similar pervasive devices share many limitations with human beings. For instance, just like people, RFID tags can neither remember long passwords nor keep long calculations in their working memory. Likewise, low-cost wireless devices and humans have similar advantages and disadvantages, i.e., tags are better at performing logical operations like AND & XOR. Tags are also better at generating random values than people. In this context, well-studied human authentication protocols utilized for proving human's identity to a machine, can also be applied in low-cost wireless devices.

It is true that securing low-end wireless devices is a challenging issue because of their limited resources and small physical form. Towards this direction the HB+ and other analogous protocols [21],[24] can contribute to the problem of secure device authentication. Nevertheless, while theoretically the HB+ protocol is secure against both passive and active aggressors and should be realizable for implementation in current RFID tags, a number of open questions remain before the HB+ can see practical realization [22]. Moreover, as already mentioned, do not neglect that HB+ and alike protocols proposed both for RFIDs and sensors devices lean against symmetric secrets stored inside the device, which in turn can be entirely revealed through active or physical attacks, such as electron microscope probing as discussed in [25].

*3.4.5. The way ahead: The Jini Technology*

Jini technology (www.jini.org) is a promising Service Oriented Architecture (SOA) that defines a programming model which both exploits and extends Java technology to enable the construction of secure, distributed systems consisting of coalitions of well-behaved network services and clients. Jini technology can be used to build adaptive network systems that are scalable, evolvable and flexible as typically required in dynamic computing environments. Jini allows for immediate recognition of new devices in a network. In the Jini architecture, each new device that joins the network instantly defines itself to the network device registry. When a user connects a device such as printer, storage device etc, it becomes immediately available to others. Security is really strong and supported by the Java Cryptographic Extension (JCE). The early version of Jini had several security issues [26], but Jini v2 rectified most of them. In the future, Jini may serve as a form of Trusted Third Part (TTP), operating on a host device, e.g., laptop computer or PDA, to authenticate devices on the network. Jini may also monitor device usage by tracking device authentication and controlling network access. As Jini is tied to the Java programming language (e.g., the actual security mechanism of Jini is hidden away from the specification by Java Class) and it is still an emerging technology will not be further addressed within the scope of this paper.

## 4. PRIVACY CONCERNS

Naturally, device authentication is strongly related to user's (or device's) privacy. The chief question here is how to preserve privacy, that is, logically disassociate the user from the device that he operates; in other words how to correctly identify and authenticate a device without disclosing user's private information, thus preserving anonymity, context privacy, location identity etc. Unfortunately, the location of a device, for example when it is identified through a corresponding device certificate, as in the 802.16 case, can be exploited to infer highly private information.

Upon acquiring the device's unique ID one has the ability to permanently associate the device with its human operator and consequently track its (i.e., his) current and past location. Therefore, the main concerns here are anonymity and location privacy. Note, that this observation pertains not only to the user's privacy but also to the privacy of the device itself, when referring to devices that are not directly operated by humans, like sensors and RFID tags. For example, in a battlefield sensor devices should not reveal their identity or location to the enemy. The same applies for sensors that gather confidential information, like wild animal movements, as in this case sensor's ID and location can be exploited by poachers. Wearable sensors or RFIDs may be subject to the same problem. Hence, device authentication mechanics must be carefully designed and implemented, lest they may become a ubiquitous surveillance system.

The privacy problem is expected to grow with the advent of foreseeable 4G highly pervasive environments [27],[28]. In such an environment supposing that the administrative domains collude, they can track the whole movement of any device (user) simply by observing the use of its static device (or user) identifier. Furthermore, even when administrative domains do not collude there can be a location privacy breach, since every single domain can recognize an old device (user) that returns to it. It is thus, more than obvious, that systems' logistic files can be anytime processed to disclose information about the entire history of movements of a specific device (user).

Currently, many wireless systems strive to protect the anonymity of their users via the employment of pseudonyms. For instance, GSM and UMTS mobile systems employ Temporary Mobile Subscriber Identities (TMSI) to protect users' permanent identities known as International Mobile Subscriber Identity (IMSI). Network Access Identifier (NAI) [29] utilized in several

homogenous or heterogeneous network realms that belong to different operators is also classified into this category of solutions. For example, in the case of IEEE 802.11, 802.16, UMTS networks users may be anonymously identified by the network by using a temporary NAI in the form of: *temp_user_id@domain_id*. This temporary NAI employs as user_id a random unused string, which only the home domain is able to associate with the true identity of the user, and as domain_id the domain_id of the user's home network. Each temporary user_id can be used once for every single domain and by one user at a time.

Under these circumstances the user is authenticated anonymously by employing methods like PEAP [30] and EAP-TTLS [31]. Both aforementioned EAP methods execute in two different phases where the first one is actually a one-way TLS handshake; only the server reveals its identity to the user's device, while the user sends its temporary NAI. After the first phase has been successfully executed, both ends have constructed a secure TLS channel. During the second phase and under the protection of the TLS channel the user sends his true identity to the authentication server. Note that all the aforementioned procedures are a native part of the 802.1X authentication framework. As a result the same mechanics can be used for device authentication as well, thus preserving privacy.

The same applies for 802.16 networks at least for those that support PKM version 2 [10], which in turn is fully compatible with 802.1X. The TCG solution can be also considered as privacy preserving as long as no unique information of the TPM chip is revealed during device authentication procedure. The smart cards option is privacy secure too assuming that the corresponding smart card remains always in the possession of its legitimate owner. On the contrary, location-based control mechanisms are by nature susceptible to surveillance. Consequently, they must be cautiously employed and only for applications that really require this feature. Privacy is rather infeasible to preserve with the electromagnetic signatures scheme. So, this solution fits better to closed environments with small user population and on an opt-in basis for the users to join. Lastly, AKE protocol solutions can support privacy if implemented properly. This means that: (a) the authentication protocol should not reveal device or user specific information to potential attackers and (b) the authentication credentials are securely stored in the device and being frequently refreshed.

## 5. DISCUSSION

### *5.1 General Issues*

We shall stress that, complementary to user authentication, device authentication is of utmost importance in today's wireless networks, giving the fact that: (a) because of their small size, small devices are easily lost, stolen, compromised or replaced by rogue ones. This is further assisted by the intrinsic nature of wireless communication. (b) User authentication may be disabled, which is sometimes the default mode, (c) even if user authentication is enabled, the authentication mechanism may be weak or easily bypassed, (d) in cases where authentication is enabled, changing the authentication information regularly is rarely performed, (e) processing and memory limitations of the device per se may weaken defenses to potential inroads. (f) In open, distributed and highly ubiquitous realms authentication should not be limited to authenticating human users, but rather it should be able to authenticate roaming devices, applications and mobile code that can run within the corresponding realm as well.

At present, there exist several software-based ways to safeguard mobile devices VPNs, firewalls, upper layer data encryption software, device management solutions, to name just a few. These types of solutions typically protect the data and/or the operating systems of the devices from attacks, but cannot guarantee the integrity and authenticity of the hardware platform on

which they are running. For example, while SIM or UICC cards are employed in wireless cellular networks to authenticate users, they cannot ensure that the computing platform on the mobile equipment is trustworthy too. Also, many applications of cryptographic authentication protocols are vulnerable against adversaries who perform real time active attacks. For instance, when identifying a physical device like a wireless AP, common authentication schemes can be bypassed by faithfully relaying all messages between the communicating participants. This attack is well known in the literature as *mafia fraud*. Several researchers have already manifested that is not possible to thwart mafia frauds simply by using cryptographic schemes [32]. Furthermore, the aforementioned class of solutions does not contribute much in protecting the unique identity of a handheld device such as a mobile phone. When intercepted, these identities can be further utilized to install rogue network components in absence of effective access control mechanisms.

Without doubt, device authentication is a hard problem to deal with, as it involves some sort of bootstrapping trust between the access control mechanism and the stranger device or between several stranger devices in ad-hoc mode. For instance, ad-hoc networking and security are contradictory, as these networks have to be open to additional devices. Thus, authentication of devices and/or users is vital to sustaining a considerable security level. The issue of secure device authentication is becoming even more complicated considering: (a) the heterogeneity of the wireless access technologies that currently exist and (b) the diversity of network providers reflected in their security policies.

As already stated, another important parameter here is that most existing security infrastructure is targeting to traditional human-to-machine interaction, while device authentication in pervasive environments, as with web services, involves scalable machine-to-machine interaction. For instance, conventional authentication and access control methods require frequent user interaction in the form of manual logins and file permissions. However, by doing so, the vision of non-intrusive ubiquitous computing is often infringed. Moreover, in many cases the security requirements of an intelligent space may differ according to the context of the realm itself. Traditional security mechanisms do not adapt their security policies to a changing context. So, context awareness and automated reckoning to the identification and authentication of devices/users and access control to resources and services is also desirable.

### *5.2 Comparison of Schemes*

In the previous section we investigated several device authentication schemes and discussed their pros and cons. Generally, schemes based on symmetric cryptography have obvious performance advantages over public-key cryptography; they fit much better to low-end wireless devices and ad-hoc modes, but usually suffer from complex key management. They also mandate some sort of trust across the entire network as a device moves from one wireless domain to another. Admittedly, schemes based on public-key technology offer less computation for more communication rounds, but are still too costly to be practical for at least non-infrastructure wireless networks that involve low-power computing devices. However, in order to have a more complete view about the applicability and robustness of each scheme it is necessary to classify them based on several key factors.

Table I depicts an aggregated, comparative view of all the anticipated schemes considering eight basic criteria in accordance with the analysis provided in section 3: (a) supports infrastructure and/or ad-hoc deployments, that is, centralized and/or distributed, (b) requires symmetric and/or asymmetric key technology, (c) effectiveness and robustness in terms of security, (d) scalability, (e) Resilience, i.e., resistance against node capture, (f) practicality to implement, (g) supports heterogeneity in terms of access technologies and trust relationships between network providers and (h) can preserve privacy.

**Table I. Device authentication schemes comparison**

| Scheme Description | Infrastructure /Ad-Hoc | Symmetric/ Asymmetric | Effectiveness / Robustness | Scalability | Resilience | Practicality | Supports Heterogeneous Environments | Preserves Privacy |
|---|---|---|---|---|---|---|---|---|
| IEEE 802.1X | Infrastructure | Asymmetric | High | High | High | Moderate | Mostly | Fully |
| IEEE 802.16 | Infrastructure | Asymmetric | High | High | High | Moderate | Partly | Fully (PKM ver. 2) |
| Trusted Computing | Mainly Infrastructure | Both | High | High | High | Moderate | Mostly | Depends on the implementation |
| Smart Cards | Mainly Infrastructure | Both | High | Moderate | High | Fair | Partly | Yes |
| Location-based Access Control | Mainly Infrastructure | Not applicable | Moderate | Moderate | High | Moderate | Partly | Yes, if designed properly |
| Electromagnetic Signatures | Mainly Infrastructure | Not applicable | Moderate | Moderate | High | Moderate | Partly | No |
| HB+ and other similar protocols (AKE) | Both | Symmetric | Moderate | Fair | Depends on the implementation (key distribution mechanism) | High | Partly | Yes, if designed properly |

Briefly, we note that the first four solutions score high in the effectiveness / robustness criterion due to their asymmetric and hardware oriented nature, while the remaining three should be classified as “moderate”. This is due to their symmetric nature and independency from device (hardware) parameters in the case of AKE protocols, and the complexity in managing hardware and defining proper policies in the case of the other two. Considering the scalability criterion the first three asymmetric solutions are characterized as highly scalable, while the symmetric AKE protocols score low due to the complex key management they require. Contrariwise, the latter approach is straightforward to implement because of its symmetric nature. 802.1X and trusted computing frameworks are mostly adaptable to hybrid network environments, in terms of access technology and security policies in force. This is due to the open character of the IEEE 802.1X standard and standardized network components that employs (e.g. AAA entities) from the one hand, and the uniformity that the TCG specifications can guarantee, if implemented correctly, on the other. Naturally, device authentication across multiple networks may also increase the handover delay time. Assuming that the network domains involved, implement the same (or compatible) device authentication scheme, the aggravation in terms of performance is expected to be similar to that of a secure hand-off optimization scheme [33]. Post-authentication is always an option in order not to disrupt active services during handoff.

Last but not least, all but one solution can be considered as resilient against node capture. In case a device been stolen and cloned all the other devices are not affected and preserve their ability to securely authenticate themselves to others. This is not always true for the last scheme, where its resilience depends on the implementation per se. For example, if some network nodes, or worst all of them, share a common master or transient key, then compromise of security credentials stored in one node or exchanged over radio links, reveals security information about other nodes/links in the corresponding network realm too.

As a general remark it seems that the trade-offs between security robustness and lightweightness in terms of processing power and accompanying infrastructures on the one hand, and between ad-hoc and infrastructure modes on the other is not easy to fulfil. Currently, the trusted computing approach and the IEEE 802.1X framework blended with emerging technologies like the Jini one seem to be the most promising solutions towards tackling the device

authentication problem. On the downside, these options are rather impractical for small to medium scale ad-hoc deployments, WSN and RFID networks, due to the PKI and Authorization, Authentication, Accounting (AAA) entities that they mandate and the associated cost that goes with them. The IEEE 802.16 solution although fully compatible with 802.1X is more or less custom-tailored to Wi-Max networks. All the other approaches are very interesting still they have to prove their effectiveness in the long run in terms of security robustness, scalability, key administration and ease of materialization.

Moreover, putting aside basic properties like computational and communication efficiency, involvement of a third party (if any), storage of secrets, nature of security guarantees etc, authentication protocols provide assurances attesting the identity of an entity only at a given snapshot in time. As already mentioned in section 3.1, usually the continuity of such an assurance is required, so additional techniques are necessary to cope with active attacks; the adversary insinuates into the communications line after the successful authentication of the legitimate device. Countermeasures to thwart this threat include:

(a) Performing re-authentication periodically, or at the beginning of each event / transaction. However, the adversary may disconnect the rogue device every time re-authentication is performed before re-connecting.
(b) Binding the authentication process to a constant integrity service. This means that the identification and authentication phases should be intertwined with a key establishment mechanism for each particular session. The established key will be used by a subsequent integrity scheme.

In our opinion one global universal solution is at present difficult to form. As already pointed out some of the aforementioned solutions (e.g., 802.1X, trusted computing, Jini etc) are not yet complete or mature, while others (AKE protocols, location-based access control, electromagnetic signatures) have been implemented to deal with particular wireless access technologies, requirements and needs. It is thus better to orientate ourselves into choosing one of the aforementioned schemes, according to our particular needs and interest or alternatively, if applicable, develop a custom-made or hybrid solution.

In this context, the middle course solution is epitomized by the 802.1X framework through the employment of EAP authentication methods. This option is already compatible with several wireless standards (802.11, 802.16, 3GPP), but above all is flexible and extensible giving the fact that new authentication EAP methods (specific to the device authentication issue) can be easily defined and amended. It is also potentially privacy preserving, when proper EAP methods are utilized as discussed in section 4. For example, infrastructure devices can be authenticated by utilizing a method like EAP-TTLS, while low-end wireless devices employ a more computational and communication efficient EAP method. In all cases however, the solution(s) employed should bind authentication credentials with unique characteristics of the device and should guarantee session freshness as described earlier in this subsection. Also, bear in mind that even if a certificateless EAP method is selected the need for some rudimentary infrastructure (i.e., AAA server) remains. Therefore, the device authentication problem must be treated differently in case of ad-hoc or other types of infrastructureless networks and sensor or RFID realms, i.e., by employing a proper AKE protocol.

### *5.3 An Example: Device Authentication and Smart Home Security*

Nowadays, device authentication becomes very prominent to smart home security. Generally, a smart home environment can be seen as a usual ubiquitous environment consisting of a network of computerized electronic appliances. In this context, the appliances should provide security services such as device/user authentication, and access control. So far, several works have investigated the authentication parameter in smart home environments. However, most of them

deal only with user authentication within the smart home, rather with device authentication. In the following we briefly present those that cope with device authentication too.

The authors in [34] have proposed the Context-Aware Security Architecture (CASA) for emerging applications, which is a security platform on middleware level. Specifically, the CASA system, implemented in Java, is responsible for transparently providing a security service such as access control and granting authority under the mobile computing system that regularly interact with various network accesses, services and devices. Context-aware authentication, build on Generalized Role-based Access Control (GRBAC) model, identifies user's ID, location and a role, and finally authenticates the user using biometric technology or a sensor like active badge.

The work in [35] presents Cerberus, a core service in the Gaia project that integrates identification, authentication, context awareness, and reasoning. Cerberus enhances the security of ubiquitous applications that are built using Gaia. Although Gaia authentication device module(s) is able to authenticate a user with various authentication methods, i.e., Kerberos, SESAME, username/password, and a challenge-response through shared secret, it cannot control differentiated accesses. The author can also refer to [36] which attempts to offer a more advanced security service for smart homes based on [34],[35].

The ZigBee home network service presented in [37] copes with device authentication. Clients can connect to a webpage for controlling ZigBee devices. ZigBee device information is stored into a database while the server program manages ZigBee devices. The web server is responsible for all the transactions taking place between the client and ZigBee home network services. In this context, device authentication is used to authenticate ZigBee devices for users. In other words, only authenticated ZigBee devices can be controlled by users through the ZigBee smart home webpage. If a user tries to authenticate towards the website, the server compare authentication credentials with that of database. The user can control the ZigBee devices after successful comparison. For doing so it is assumed that every ZigBee device has assigned a unique authentication number. While the authors correctly identify the need for device authentication they do not provide detailed information on how it is achieved. Nevertheless, their scheme seems to be quite simple and easy to bypass as it appears to be similar to “white-black list” employed in 802.11 networks.

A really robust user/device authentication scheme for smart homes is proposed in [38]. The authors introduce a user-centric secure multimedia service system suitable for intelligent home. In addition, a method of improving security, user efficiency, and home device authentication is proposed. The authors clearly state that within an “intelligent home, flawless service should be provided by ensuring the reliability of public resources through device authentication, along with existing user authentication process. In order to achieve this, authentication services such as intelligent device authentication carrying out synchronization, device loss, and robbery prevention are needed”. A recognition algorithm for detecting the location of a moving node is also suggested dealing with location based access control as discussed previously. The system employs a standard certificate method that uses PKI for user authentication. Device authentication is performed through a different service provider acting as a Certification Authority (CA), such as an Internet Service Provider (ISP). In a nutshell, the proposed method can be classified as 802.1X compatible. Therefore, while effective and robust it inherits the problem of decreased speed as it requires a public key method in each session of user authentication, plus an additional authentication process through the CA every time a new device is added.

Last but not least, the authors in [39] present an authentication protocol using the Simple Password Exponential Key Exchange (SPEKE) strong password method to address information-security deficiencies found in most smart home automation systems. The Home Automation Authentication Protocol (HAAP) introduced in this work is specifically designed to provide

sufficient authentication facilities to be used for home automation (HA) devices. Thus, this solution can be classified to AKE solutions discussed in subsection 3.4.4.

## 6. CONCLUSIONS

The term authentication is used in a very broad sense and is usually specific to a particular security goal, e.g., access control, entity authentication, key or message authentication. In this work we define device authentication as the entity authentication in which the objective is to securely identify and further authenticate a physical device possibly at a specific location. Beyond doubt, this issue is expected to gain more and more attention in the forthcoming era of ubiquitous computing.

In this context, a constructive analysis of the current potential solutions and trends to the device authentication issue has been given. Each scheme was briefly presented and some comments including implementation problems and research challenges have been provided. Privacy issues as a side-effect of device authentication were investigated as well. Finally, a comparison of the presented schemes was conducted based on several criteria.

## ACKNOWLEDGEMENTS

The third author was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2008-0174).

## REFERENCES

1. C. Boyd and A. Mathuria, “Protocols for Authentication and Key Establishment,” *Information Security and Cryptography Text and Monographs*, Springer, 2003.
2. A.J. Menezes, P.C. Van Oorschot, and S.A. Vanstone, “Handbook of Applied Cryptography,” CRC Press Series on Discrete Mathematics and Its Applications, 2001.
3. NIST, “An Introduction to Computer Security: The NIST Handbook,” Special Publication 800-12, 1997.
4. C. Hartung, J. Balasalle, and R. Han, “Node Compromise in Sensor Networks: The Need for Secure Systems,” Technical Report CU-CS-990-05, Department of Computer Science University of Colorado, Boulder, 2005.
5. N. Sastry and D. Wagner, “Security Considerations for IEEE 802.15.4 Networks,” *WiSE’04*, pp. 32-42, Philadelphia, Pennsylvania, USA, 2004.
6. IEEE Std 802.11i-2004, “Amendment to IEEE Std. 802.11, 1999 Edition, Amendment 6: Medium Access Control (MAC) Security Enhancements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications,” IEEE Press, 2004.
7. IEEE 802.1X-2004 IEEE Standards for Local and metropolitan area networks - Port-Based Network Access Control, 2004.
8. B. Aboba, “Extensible Authentication Protocol (EAP),” IETF RFC 1661, 2004.
9. B. Aboba and P. Calhoun, “RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP),” IETF RFC 2869, 2003.
10. IEEE P802.16e/Draft12, “IEEE Standard for Local and metropolitan area networks, Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands,” published 2005.
11. TCG, “TPM Main Part 1 Design Principles Specification Version 1.2 Revision 85,” 2005.
12. TCG, “TCG Mobile Trusted Module Specification, version 0.9, Revision 1,” 2006, DRAFT.
13. TCG, “TCG Specification Architecture Overview,” Specification Revision 1.2, 2004.

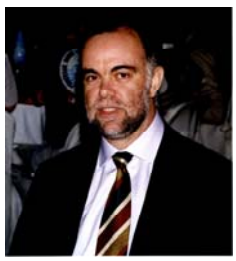
14. D. Bruschi, L. Cavallaro, A. Lanzi, and M. Monga, "Attacking a Trusted Computing Platform, Improving the Security of the TCG Specification," Technical Report RT 05-05, 2005.
15. J. Hendricks and L. van Doorn, "Secure Bootstrap is Not Enough: Shoring up the Trusted Computing Base," *Proceedings of the 11th SIGOPS European Workshop*, ACM SIGOPS, Leuven, Belgium, 2004.
16. Y. Zheng, D. He, W. Yu, and X. Tang, "Trusted Computing-Based Security Architecture For 4G Mobile Networks," Proc. of the Sixth International *Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'05)*, pp. 251-255.
17. W. Jansen, S. Gavrila, C. Séveillac, and V. Korolev, "Smart Cards and Mobile Device Authentication: An Overview and Implementation," NIST, NISTIR 7206, 2005.
18. N. Sastry, U. Shankar, and D. Wagner "Secure Verification of Location Claims," *ACM WiSE'03*, pp. 1-10, California, USA, 2003.
19. K.A. Remley, et al., "Electromagnetic Signatures of WLAN Cards and Network Security," *IEEE International Symposium on Signal Processing and Information Technology*, pp. 484-488, 2005.
20. D Henrici and P. Muller, "Hash-based Enhancement of Location Privacy for Radio-Frequency Identification Devices using Varying Identifiers," in *Pervasive Computing and Communications (IEEE PerCom '04)*, IEEE Computer Society, pp. 149–153, 2004.
21. H.-A. Wen, C.-L. Lin, and T. Hwang, "Provably secure authenticated key exchange protocols for low power computing clients," *Computers & Security*, Vol. 25, pp. 106-113, Elsevier Science, 2006.
22. A. Juels and A. Weis, "Authenticating Pervasive Devices with Human Protocols," in Proc. of *ASIACRYPT '01*, Bart Preneel (Ed.), Springer-Verlag, LNCS 2248, pp. 149-153, 2001.
23. D.-N. Duc and K. Kim, "Human Authentication Protocol for Distributed Computing Environment," Proc. of *WISA '06*, pp.367-372, Jeju Island, Korea, 2006.
24. S.A. Camtepe and B. Yener, "Key Distribution Mechanisms for Wireless Sensor Networks: A survey," Technical Report, 2005.
25. R. Anderson and M. Kuhn, "Low Cost Attacks on Tamper Resistant Devices," in Int'l *Workshop on Security Protocols ('97)*, vol. 1361, Springer LNCS, pp. 125–136.
26. P. Eronen and P. Nikander, "Decentralized Jini Security," in *proceedings of Network and Distributed System Security Symposium (NDSS)*, 2001.
27. A.R. Beresford. and F. Stajano, "Location Privacy in Pervasive Computing," *IEEE Pervasive Computing*, Vol. 2, Issue 1, pp. 46- 55, 2003.
28. G. Karopoulos, G. Kambourakis, and S. Gritzalis, "Privacy Preserving Context Transfer in All-IP Networks," in proceedings of the *MMM-ACNS '07 conference*, St. Petersburg, Russia, Springer LNCS, 2007.
29. B. Aboba, M. Beadles, J. Arkko, and P. Eronen, "The Network Access Identifier," RFC 4282, 2005.
30. A. Palekar, D. Simon, J. Salowey, H. Zhou, G. Zorn, and S. Josefsson, "Protected EAP Protocol (PEAP) Version 2," IETF Internet Draft, draft-josefsson-pppext-eap-tls-eap-10, expired, 2004.
31. P. Funk and S. Blake-Wilson, "EAP Tunneled TLS Authentication Protocol (EAP-TTLS)," IETF Internet Draft, draft-ietf-pppext-eap-ttls-01, 2002.
32. A. Alkassar, C. Stüble, and A-R. Sadeghi, "Secure object identification: or: solving the Chess Grandmaster Problem," *Proc. of the workshop on New security paradigms*, pp. 77-85, Switzerland, 2003.
33. G. Karopoulos, G. Kambourakis, and S. Gritzalis, "Survey of Secure Hand-off Optimization Schemes for Multimedia Services over all-IP Wireless Heterogeneous

Networks," *IEEE Communications Surveys and Tutorials (COMST)*, Vol. 9, No. 3, pp. 18-28, 2007, IEEE Press.

34. M. J. Covington, P. Fogia, Z. Zhiyuan, and M. Ahamad, "A Context-Aware Security Architecture for Emerging Applications," *Annual Computer Security Applications Conference*, pp. 249-258, 2002.
35. J. Al-Muhtadi, A. Ranganathan, R. Campbell, and M.D. Mickunas, "Cerberus: A Context-Aware Security Scheme for Smart Spaces," *Pervasive Computing and Communications*, pp 489-496, 2003.
36. J.-S. Cho, S.-H. Park, Y.-J. Han, and T.-M. Chung, "CAISMS: A Context-Aware Integrated Security Management System for Smart Home," in proceedings of *ICACT '07*, pp. 531-536, 2007.
37. C. Lee, J. Lee, W. Park, and S. Yang, "End to End ZigBee Home Network Security Solutions," in proceedings of *ITC-CSCC '07*, Busan, Korea, pp. 2-6.
38. J.-H. Park, S. Lee, D.-H. Lee, J. Lim, I.-H. Hong, and L. T. Yang, "Design and Implementation of the UsMSS: User-centric secure Multimedia Service System in Intelligent Home," in proceedings of *International Conference on Multimedia and Ubiquitous Engineering (MUE'07)*, pp. 1233-1238, 2007.
39. S.S. Leong and C.H. Vun, "Design and implementation of an authentication protocol for home automation systems," *IEEE Transactions on Consumer Electronics*, Vol. 44, Issue 3, pp. 911-921, 1998.

## ABOUT THE AUTHORS

**G. Kambourakis** is a Lecturer at the Department of Information and Communication Systems Engineering, University of the Aegean, Greece. His research interests are in the fields of Mobile and ad-hoc networks security, VoIP security, security protocols, PKI and mLearning and he has more than 50 publications in the above areas.



**S. Gritzalis** is the Head of the Department of Information and Communication Systems Engineering, University of the Aegean, Greece and the Director of the Laboratory of Information and Communication Systems Security. His research work includes several books on Information and Communication Security technologies topics, and more than 170 journal and international conference papers.

**J. Hyuk Park** received his Ph.D. degree in Graduate School of Information Security from Korea University, Korea. Before September, 2007, Dr. Park had been a research scientist of R&D Institute, Hanwha S&C Co., Ltd., Korea. He is now a professor at the Department of Computer Science and Engineering, Kyungnam University, Korea. Dr. Park has published many research papers in international journals and conferences. Dr. Park has been served as Chairs, program committee or organizing committee chair for many international conferences and workshops. Dr. Park has won a Best Paper Award of the 2nd International Conference on Information Security and Assurance (ISA 2008).