ADVANCED SECURITY MANAGEMENT ON MOBILE DEVICES IN AD HOC NETWORKS

H. BUCHHOLZ, S. PREUSS, I. SEDOV, C. CAP
University of Rostock, Dept. of Computer Science,
Chair for Information and Communication Services
Albert-Einstein-Str. 21, 18059 Rostock, Germany
Email:{hz, spr, igor, cap}@informatik.uni-rostock.de

M. HAASE, M. SCHMALISCH, D. TIMMERMANN
University of Rostock, Dept. of EE and Information Technology,
Institute of Applied Microelectronics and CS
Richard-Wagner-Str. 31, 18119 Rostock, Germany
Email:{marc.haase, matthias.schmalisch, dirk.timmermann}@etechnik.uni-rostock.de

ABSTRACT
This paper describes an approach to centralize security management and security functions on mobile devices into a central component to enable a more natural, tacit interaction in ad hoc networks. Existent wireless communication technologies provide embedded security algorithms to protect wireless links, however the necessary key management task, e.g. key exchange, key administration, is mostly transferred to the user. Thus spontaneous networking is hampered by user intervention to accomplish secure links. In this paper we introduce the *Advanced Security Manager (ASM)* a security management extension for mobile devices that enables authorization, authentication, encryption, non-repudiation based on user, device, or application requirements.

KEY WORDS
Wireless Security, Applied Cryptography, Access Control

1 Motivation

Mobile ad hoc networks (MANET) are self-organizing networks in which mobile wireless devices communicate with each other independent of stationary infrastructure. Nowadays, the term ad hoc is frequently used for home, office or other networks, where mobile devices organize spontaneous networks for the period of the communication session. An example could be a conference or a meeting, where participants want to exchange documents between their mobile devices using a spontaneously organized wireless network. In this paper we will cover this type of ad hoc networks.

Wireless communication technologies provide on the one side security algorithms to protect the wireless channel, however there are various examinations proving, that they are insufficient to establish secure ad hoc connections.

Firstly, algorithms and protocols used for securing data transmissions are vulnerable. Bluetooth encryption is not strong enough [1], IEEE 802.11b WEP has broken [2], and IrDA [3] does not provide any data protection mechanisms at all. Secondly, integrated protection mechanisms of wireless technologies are limited to authentication and data encryption on link layer. They do not consider key exchange, user authentication, non-repudiation and further security requirements. Finally, users have limited control over communication interfaces of the mobile device and are overemployed with administration and configuration of security parameters. Thus spontaneous networking with mobile devices is hampered and the setup of security parameters of wireless links is sofar unsatisfactory.

To solve the above mentioned problems, software developers implement own security mechanisms into the application layer. This complicates the application development process, consumes additional memory and processor resources and leads to compatibility problems between different applications. Furthermore, applications developers often have only basic knowledge about security paradigms in ad hoc networks and do not take all properties of spontaneous networking into account.

An efficient solution is the encapsulation of security functions into a central component below the application layer. This reduces application size and complexity, and provides security services, e.g. authentication, authorization, and encryption to the application by an consistent API. The central security component provides an interface to existing security mechanisms of wireless technologies, to software based algorithms coming from cryptographic libraries, and to hardware based cryptographic algorithms. The main advantage of this architecture is the ability to find a good compromise between the required security for data to transmit and the limitations of mobile devices in terms of computational power and energy resources.

In this paper we present the *Advanced Security Manager (ASM)*, a central security component for mobile de-

vices realizing the above mentioned security requirements. It provides adaptive security mechanisms for ad hoc communication scenarios covering initial key exchange, application definable security profiles, no need for manual administration and device or user authentication. The major challenge of the ASM is to preserve the ad hoc character during connection establishment of a secret link.

The paper is organized as follows. Section 2 presents the concept of the Advanced Security Manager followed by a summary of its security services in Section 3. Section 4 describes the implementation of the ASM on current available mobile devices (PDA) utilizing Linux as operating system and Bluetooth as wireless communication technology. Section 5 concludes this paper with an outlook on future research activities.

2 Concept

The *Advanced Security Manager (ASM)* is a universal security management component for ad hoc networks. The large variety and the frequent change of communication peers, mobile computing devices are exposed to, requires protection schemes, that automatically adapt to the current context and do not need prior configuration. The ASM adds this missing functionality to mobile devices by providing *authentication* and *authorization*, based on the identity of the user or the device, as well as *encryption, integrity*, and *non-repudiation* of the transmitted data. Therefore the ASM enables the establishment and maintenance of protected spontaneous connections.

The ASM is hooked into the transport layer and observes connection establishment and network traffic (see Figure 1). Its exact position depends on the network technology it is applied to. It administrates information about applications, users, devices, access rights and established connections in several databases. A connection request on transport layer is observed by the ASM that decides the completion of the connection establishment. Furthermore the transport layer has to bypass every data packet for a established connection through the security engine of the ASM. The restriction of the ASM is, that it supports only connection oriented protocols, e.g. L2CAP and TCP, because a unique relation between communication peers is required in order to set up the security context.

The ASM core is made up by the *security engine*. It intercepts network traffic and applies security algorithms at packet level. The engine contains different security services, e.g. authentication, authorization, and encryption, that utilize pluggable security primitives implemented as software algorithms, as hardware based cryptographic modules, or as native security algorithms from wireless technologies, e.g. Bluetooth encryption or WEP.

The algorithms, effectively applied to the transmitted data, are negotiated between peers during connection establishment, and depend on capabilities and requirements at each side. Those requirements are defined by *security policies* that can be separately assigned for users, devices, applications, and connection type.

ASM enabled devices remain backward-compatible. Applications which are not aware of the ASM are supported by the *General Management Entity (GME)*. It possesses a user interface to setup a default security policy or a application specific policy. After this manual security configuration step, legacy applications can be executed as usual. The ASM handles the security setup fully transparent for this application. ASM aware applications utilize the ASM API to setup their specific security policy without user intervention.

In the next subsections, we describe the functionality of the ASM architecture in detail.

2.1 Security Engine

The security engine is the core module of the ASM. All network connections are redirected to pass the security engine, which in turn applies connection specific security algorithms to datagrams. According to the negotiated protection level, outgoing datagrams can be encrypted, signed, or protected against modification using an HMAC. At the other side the corresponding actions are carried out, datagrams are decrypted, their signature is checked, and their integrity is verified. If everything is correct, a plain text datagram leaves the security engine and is delivered to the upper layer. The set of security primitives is not fixed. It can be extended or customized by additional security primitives according to the devices capabilities (pluggable security concept).

2.2 Application Registration

Applications, that either offer a service for remote usage or want to use a remote service, have to register at the ASM. Without a prior registration of an application, no connection to this application respectively from this application can be secured by the ASM. An application registers its name and identifier as well as its security requirements, e.g. security service (encryption) and security primitive (3DES). Furthermore, the ASM supports the dynamical creation of security parameters, e.g. encryption keys, which are only valid for one connection. Legacy applications can either be treated with a default security policy or supported by the GME, that registers specific requirements on behalf of them. Applications security policies are stored in a database (see Figure 1). After closing an application its information remains into the database and is updated upon restart in order to apply changes in the security requirements.

2.3 Connection Establishment

A secured connection establishment between two applications is divided into several phases. Assuming that we have

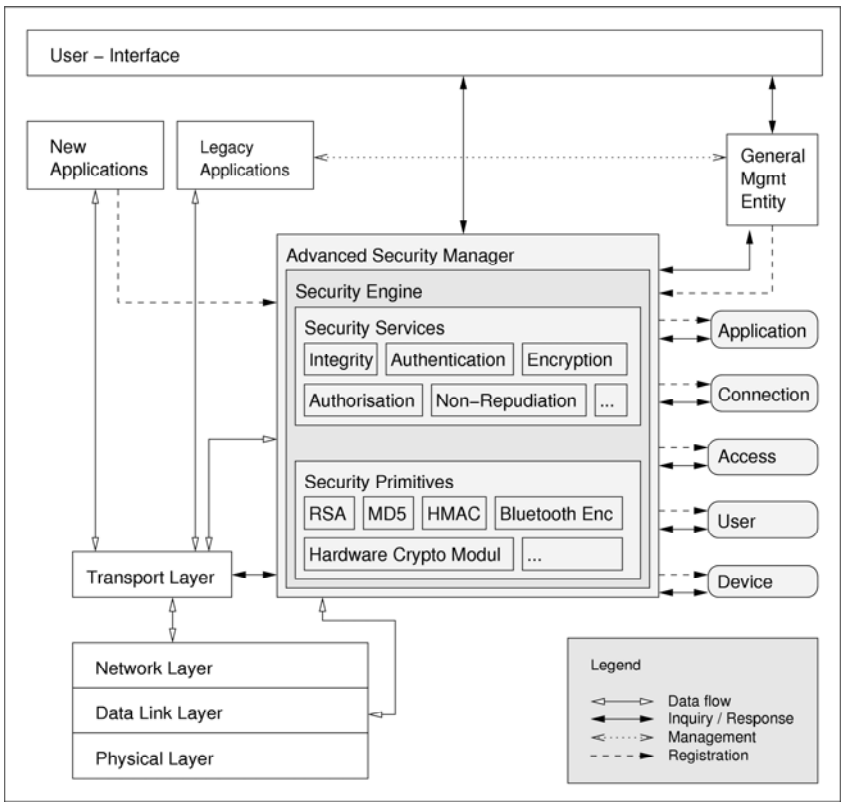

Figure 1. Communication model of the Advanced Security Manager

two devices A and B. At first, the application on device A requests a connection to an application on device B. The transport layer on device A forwards the connection request to the local ASM, to check, if the application is registered. Otherwise the connection request is discarded. Then, the ASM enables the transport layer to connect device B by utilizing the underlying protocol layers. The transport layer of device B receives the connection request and forwards them to the ASM of device B. The ASM checks the security policy of the target application and if necessary establishes a separate connection to the ASM of device A. The communication between both ASM follows the ASM protocol. At the end of the ASM protocol the security requirements for the application connection are exchanged and can be applied. Finally the ASM on device B assigns the transport layer to continue with the application connection establishment. For the created application link, every packet will be forwarded by the transport layer to its ASM security engine, to apply the negotiated security algorithms.

2.4 ASM Protocol

The ASM protocol provides the protection of the separate ASM connection and the negotiation of security parameters for the application connection. The major challenge is to preserve the ad hoc character of the ASM connection establishment. One way that we propose, is to use a fast secret key algorithm, e.g. 3DES, for traffic encryption and the Diffie-Hellmann algorithm for deriving the encryption key. Unfortunately, the Diffie-Hellmann algorithm is vulnerable by a man-in-the-middle attack. So for higher security needs, the authenticated Diffie-Hellmann algorithm can be applied for key exchange. This one provides a higher protection level because the authenticity of the communication peer can be determined on the basis of a certificate. If the application requires beside other security parameters, authentication and authorization, these are executed first (see Figure 2). Afterwards further security parameters are negotiated.

3 ASM Security Services

Before summarizing the security services of the ASM and considering possible attacks, we specify the basic security assumptions that were made in the design of the ASM.

Firstly, the ASM can only provide that level of security, the requesting application or user has chosen itself. That means that the application or the user is responsible to determine the security strength of the communication channel by selecting appropriate authentication, authorization, encryption, non-repudiation and integrity algorithms. The ASM itself attends to establish the communication channel, that corresponds to the selected security level.

Secondly, the security of the ASM relies on the secu-

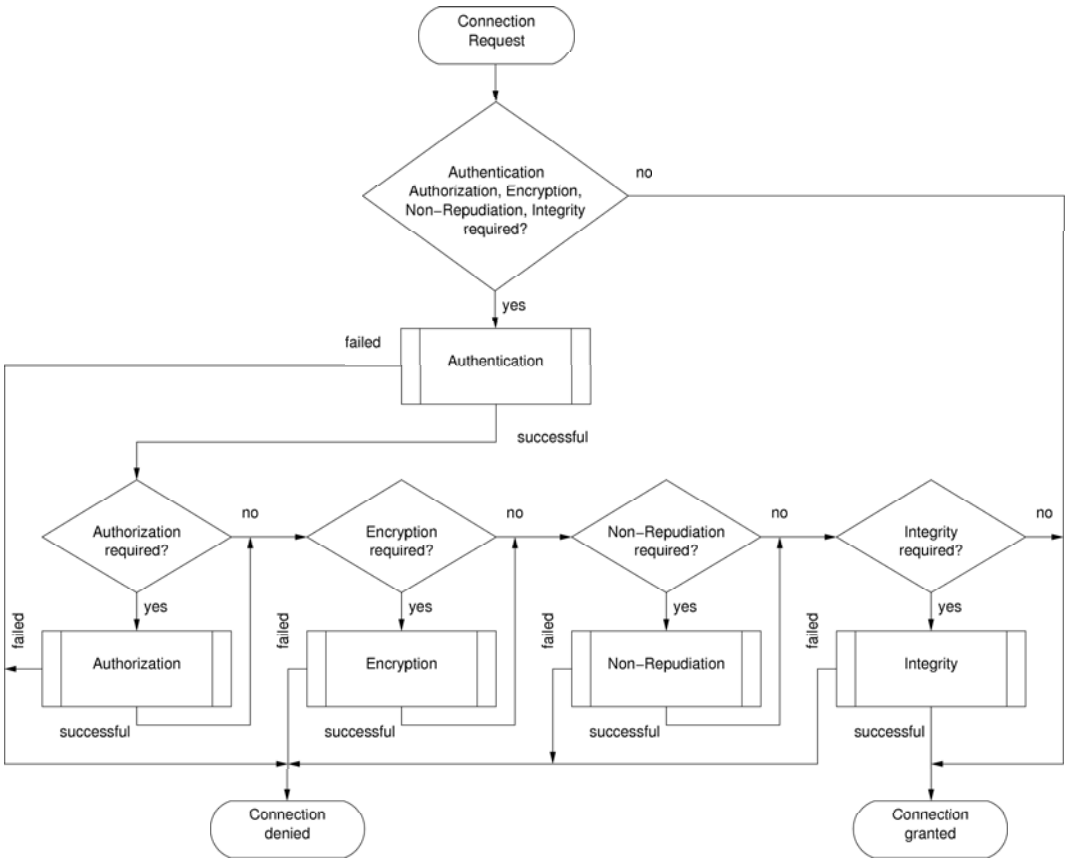

Figure 2. ASM protocol flow

rity of the used cryptographic algorithms. The ASM cannot protect from breaking a cryptographic algorithm. Therefore, the algorithms must be strong enough against passive and active attacks. If an algorithm has broken, the application or the user is responsible to change the cryptographic algorithm for future sessions.

Thirdly, to increase the security level the ASM supports public key cryptography. This requires beside secure handling of public keys, private keys, and certificates, access to a certificate authority (CA) or whose root certificates. The introduction of public key cryptography, in turn, reduces the flexibility and the ad hoc character of the whole system, because a reasonable use of certificates requires a public key infrastructure (PKI). To resolve this problem the authors in [4] propose a distributed public key management service for ad hoc networks. Capkun et al. introduce in [5] a self-organized public key management for mobile ad hoc networks. Stajano and Anderson [6] as well as Balfanz [7] propose to exchange bootstrap cryptographic information over location-limited side channel. In this case the attacker should be able to perform an active attack not only in the wireless medium, but also in the location-limited side channel. That is significantly more complex for an attacker. Another good choice are identity-based cryptosystems [8], which eliminate the need for protected key exchange channels and certificate checks. The main idea behind these systems is the usage of public available information, e.g. node name and node address, for the generation of the public key of an asymmetric encryption algorithm. While the public key is (re)producible from the identity of a user or device, the private key has to be created once out of the identity and a secret by a trustworthy third party in order to prevent attackers from easily generating secret keys. The only administrative task to be fulfilled here, is the installation of the secret key. This is an acceptable compromise.

To conclude this short survey we assume that the ASM utilizes a revocation list if a connection to a CA can not be accomplished. The revocation list is stored in the ASM database and is refreshed, when reaching infrastructure rich environments, when establishing a short-term chargeable global wireless connection to a CA, or when synchronizing the mobile device with the home pc connected to the Internet. Furthermore, certificate verification can be neglected, when the communication partner is trusted personally.

Finally, protection against denial of service attacks is not covered by the ASM.

Authentication The Impersonation Attack represents a grave security risk in all fields of ad hoc networks. If there is no sufficient authentication of communication peers

compromised nodes can join the network as supposed trustworthy members. To prevent this attack, the ASM provides device authentication and user authentication. It is also possible, that user and device authentication are applied at the same time. Because authentication is only valid for a single connection, it has to be executed prior to each connection establishment. To increase the security level and to avoid attacks, e.g. the man-in-the-middle attack, applications can perform mutual and certificate based authentication. Because authentication is pre-requisite for all other security mechanisms, it is executed first. Currently, the ASM handles authentication based on certificates, username and password, location limited side channels and native communication technology dependent methods from, e.g. *Bluetooth* or *802.11b*.

Authorization ASM authorization can be applied at user level, device level or both. Application specific access rights are stored in the access database (see Figure 1). Granting access to an application (service) requires an appropriate entry in the access database. These entries can either be pre-configured or created at connection establishment. The further can be applied to unattended devices, e.g. access points. The latter authorization method is based on interaction with the device owner. He decides whether to accept or reject a connection. Unauthorized service usage is prevented by the application of authentication and constitutive authorization. Both must be repeated bon every service usage. Access is denied when authentication fails. Through the use of authorization it is possible to define user and device specific access control to services.

Encryption The inherent vulnerability of wireless networks is, that they can be easily eavesdropped. An adversary tries to filter sensible information, e.g. private documents, passwords or keys from received data streams to get access to a system or service. Therefore, encryption of transmitted data is recommended to prevent eavesdropping. The ASM provides hardware based and software based encryption services. These can be customized in terms of algorithm, key length, and encryption mode. Beside the software based encryption algorithms, hardware based algorithms from Bluetooth chip sets, cryptographic co-processors or microprocessor specific cryptographic extensions (3DES, RSA) can be utilized. Moreover the ASM handles key derivation and key exchange protocols, that wireless communication technologies do not provide.

Data Integrity In order to prevent unauthorized modifications of data during transmission, Message Authentication Codes (MAC) or Hash algorithms in conjunction with digital signatures can be applied to datagrams. Keys for checking the validity of received datagrams are exchanged between communication peers during connection establishment, using the ASM protocol. The keys are generated randomly and only used for a single connection. For the period of a connection, these keys are stored in the connection database. MD5 and SHA1 are algorithms useable for creating datagram fingerprints witch can be signed using with either an asymmetric or a symmetric algorithm. The ASM protocol itself is secured by integrity algorithms.

Non-Repudiation In e-business scenarios it is often necessary to prove the origin of data in order to uniquely assign contracts or transactions to a specific source. Using the ASM, it is possible to authenticate a user or a device before data is transmitted. Hence contract signatures or transactions cannot be repudiated later. In order to ensure the authenticity of transmissions, each datagram is digitally signed. For instance, the Digital Signature Algorithm (DSA) can be applied at this point. The ASM cannot sign documents because it is unaware of the actions carried out in the application layer, thus the authenticity of data is ensured for the whole connection. That leads to a higher overhead and a deceleration of the transmission which can be observed with encryption and data integrity protection as well.

Enhancing User Privacy Wired and wireless communication technologies utilize mostly uniform identifiers for the connection establishment. The MAC address for Ethernet or the Bluetooth device address are two examples of persistent identifiers. A potential adversary can log passively these identifiers and can generate movement or behavior profiles of a mobile device out of it. These profiles can be assigned to an user of a mobile device. Anonymity of a device or a user can not be guaranteed by the ASM model itself. Therefore the ASM can be connected to an identity based or credential based management service. However the ASM can control the activity of the communication interface by switching the communication interface into power save modes.

4 Implementation and Evaluation

We implemented the ASM using Linux as operating system and Bluetooth as wireless communication technology [9]. The cryptographic algorithms are provided by the OpenSSL crypto library. Currently, the ASM supports asymmetric RSA encryption and symmetric 3DES encryption with modes ECB, CBC, CFB, OFB as well as symmetric Bluetooth link layer encryption. Data integrity will be supported by the MD5 and SHA1 algorithms. The authentication of users is provided by using the user name and password or by the utilization of certificates with a RSA key. The latter are also used for device authentication. Moreover, for device authentication the Bluetooth challenge-response authentication scheme is supported. Currently we work on an extension of the ASM protocol by using the resurrecting duckling security scheme [6].

To evaluate the ASM performance we measured the data throughput between a resource limited mobile device

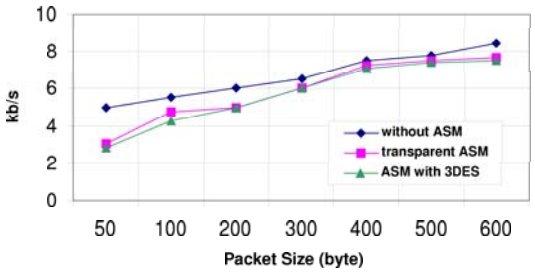

Figure 3. Comparison of the ASM influence

(PDA) and a resourceful PC system. Figure 3 shows the influence of the ASM in transparent mode (no security functions are requested by the applications) and with active 3DES encryption compared to a Bluetooth data transmission without encryption. The taken measurement shows, that the ASM does not significantly influences the data throughput. Moreover by using the ASM concept the communication technology specific security mechanisms still can be used. That is the main advantage of the ASM concept compared to SSL and IPSec. However, the overhead of the data transmission coming from applying additional security services to the data payload, reduces the nominal data rate.

5 Conclusion

In this paper we presented the ASM technology, that provides adaptive security mechanisms for ad hoc communication scenarios covering initial key exchange, application definable security profiles, no need for manual administration and device or user authentication. The ASM extends established wireless communication technologies by a flexible security management system. The security engine of the ASM supports pluggable security primitives. That enables the option to utilize security algorithms coming from software libraries, cryptographic hardware extensions and from the wireless communication technology itself. Especially technologies which possess considerable vulnerabilities in their security mechanisms, e.g. IEEE 802.11b WEP, benefit from the ASM. This cooperative security approach enables adapted security solutions for resource restricted mobile devices.

The ASM provides application level security, because applications can register their specific security requirements at the ASM. The ASM applies the requested security level to the ongoing link. The application utilizing the ASM does not need own security algorithms, that downscales the application size.

To verify the approach we implemented the ASM for the wireless communication technology Bluetooth. Our performance results prove, that enhanced security on mobile devices in ad hoc networks can only achieved, if hardware-based and software-based cryptographic functions are used together sensibly, and are controlled by an central security management component. This eliminates permanent user intervention, when moving in ad hoc network scenarios.

References

[1] M. Jakobsson and S. Wetzel. Security Weaknesses in Bluetooth. In *CTRSA: CT-RSA, The Cryptographers' Track at RSA Conference, LNCS*, Murray Hill, NJ 07974, 2001. Lucent Technologies - Bell Labs Information Science Research Center.

[2] N. Borisov, I. Goldberg, and D. Wagner. Intercepting mobile communications: The insecurity of 802.11. `http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html`, 2001.

[3] The Infrared Data Association. *IrDA Standard Specification*. `http://www.irda.org/standards/specifications.asp`.

[4] L. Zhou and Z. J. Haas. Securing ad hoc networks. *IEEE Network*, 13(6):24–30, 1999.

[5] S. Capkun, L. Buttyan, and J. Hubaux. Self-organized public-key management for mobile ad hoc networks, 2002.

[6] F. Stajano and R. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. pages 172–194.

[7] D. Balfanz, D. Smetters, P. Stewart, and H. Wong. Talking to strangers: Authentication in adhoc wireless networks, 2002.

[8] A. Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editor, *Advances in Cryptology: Proceedings of CRYPTO 84, volume 196 of Lecture Notes in Computer Science*, pages 47–53. Springer-Verlag, 1984.

[9] H. Buchholz. Security in ad hoc networks. Diploma thesis, University of Rostock, Dept. of Computer Science, April 2002.