Mobile and Wireless Communications Security

Bart Preneel
Katholieke Universiteit Leuven
Dept. Electrical Engineering-ESAT,
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
Bart.Preneel@esat.kuleuven.be

This article presents an overview of security issues for mobile and wireless communications. It describes the security requirements and architectural options for these networks. Subsequently three technologies are discussed in more detail: mobile phones (GSM and 3GSM), Wireless LAN (IEEE 802.11) and Personal Area Networks (Bluetooth). A critical evaluation is given of the strengths and weaknesses of the security solutions.

1. Introduction

The introduction of wireless data communications at the beginning of the 20th century has resulted in an increasing interest in cryptology [DL98]. A first reason is the growth of both business and military communications as a consequence of this technology, which allowed for global communications in seconds rather than weeks. In addition, it is obvious that wireless communications are as easy to intercept for an adversary as for the legitimate receiver. This resulted in a wide deployment of mechanical and electromechanical cryptographic devices in the first half of the 20th century and a growing interest in their cryptanalysis. From the 1960s, computer networks were built up for data communication over fixed wired networks; the protection of these communications was mainly restricted to military and financial communications. The popularity of the Internet and the world wide web resulted in broad use of cryptography for e-commerce and business applications. The underlying enabling technologies are inexpensive fast software cryptography and open security protocols such as TLS (SSL), SSH and IPsec as introduced in the second half of the 1990s. In spite of this development, only a small fraction of the Internet traffic is encrypted as of today. At the beginning of the 21st century, we observe a real explosion of wireless data communications with Wireless LANs (WLAN, IEEE 802.11), Personal Area Networks (PANs such as Bluetooth or IEEE 802.15, Zigbee or IEEE 802.15.4, and Ultrawideband or IEEE 802.15.4a) and Wireless Metropolitan Area Networks (WiMAX or IEEE 802.16). All these technologies have been introduced with cryptographic security from the beginning, even if the solutions are far from robust. In addition mobile data communication is growing

on the evolving GSM mobile phones using technologies such as GPRS and EDGE as on the third generation mobiles phones such as 3GSM.

For voice communications, the introduction of security has been significantly slower. The main reason has been technological limitations, but there is also a significant legal barrier, since governments want to maintain the capability to perform wiretap for law enforcement purposes. Analog voice scramblers do not offer a very high security level: protecting analog information effectively turns out to be hard. Secure digital voice encryption was available to Roosevelt during the 1945 Yalta conference but the devices were voluminous and expensive and the voice quality was not very good. Efficient digital coding of voice for mass market products had to wait until the 1980s: secure digital phones (e.g. the STUs) became available, but outside the government and military environment they were never successful. However, one can expect that with Voice over IP (VoIP) technologies, end to end security based on software encryption will become widespread. The first analog mobile phones provided no or very weak security, which resulted in serious embarrassment (e.g., the private conversations of Prince Charles being exposed or the eavesdropping of the Soviet mobile communication systems by the US). The European GSM system designed in the late 1980s provided already much better security, even if many flaws remain; these flaws did not stop the system from growing to more than 2 billion subscribers in 2006. Most of these flaws have been addressed in the 3GSM system, but even there no end-to-end protection is provided. One can expect that in the next generation of smart phones users will install software with this capability, either directly for the 3GPP voice stream or in a VoIP protocol.

This paper intends to give a brief overview of the security approaches taken in a selected number of protocols, focusing on wireless communications. Section 2 discusses the general approach to the security architecture. Section 3 gives an overview of mobile phone systems (GSM and 3GSM), while Section 4 deals with WLAN (IEEE 802.11) and Section 5 with Bluetooth as an example of a Personal Area Network technology. Mobile and wireless security is a very broad area; this paper does not have the ambition to give a complete overview of all security issues – topics that are not treated in this paper include mobile IP [MSCP04], ad hoc networks [ZH99], viruses and worms for mobile devices and the integration of wireless security with the existing Authentication-Authorisation-Access control (AAA) infrastructure.

2. Security Architecture

First we discuss the security requirements; we restrict ourselves here to the case where a mobile terminal wants to establish a wireless link with a fixed point (the base station or access point), which is the common scenario in the systems described later on. A first requirement is confidentiality of the information communicated. There are clear advantages in end-to-end confidentiality, that is, between the sender and receiver across heterogeneous networks. However, most wireless connections continue over a fixed network, and confidentiality protection is often restricted to the wireless link. For data communications,

data authentication should also be provided as well as protection against replay. In order to achieve data confidentiality, data authentication and replay protection, authenticated encryption should be used; this service requires the establishment of a secret key between the mobile terminal and the network. Most wireless services also require access control; in order to limit access to authorized user and/or for billing purposes. On the other hand, the mobile terminal wants to make sure that it is connected to a legitimate access point. This is very important since the access point has access to all the information sent by the mobile terminal and since the network may also upgrade settings or services on the mobile terminal. Access control and terminal authentication are typically achieved by running a protocol for mutual entity authentication; in practice one combines this protocol with the establishment of a session key, resulting in an authenticated key agreement (AKA) protocol. Users want to protect their privacy, hence the protection of the identity of mobile entities is an important requirement: third parties should not be able to identify or track mobile terminals or to perform traffic analysis. Finally, denial of service is a growing concern: in this case an attacker wants to degrade the performance of the network. In a wireless setting this can always be achieved by jamming the frequencies used, but sometimes more subtle attacks can be launched by exploiting the communication protocols to flood one of the nodes. Denial of service attacks are very hard to protect against in a wireless environment; we will not discuss them in the remainder of this article.

The main architectural decision is at which layer to implement security. The simplest solution is to provide data confidentiality and authentication services at the data link layer. This is an attractive option, since the wireless link is typically the most vulnerable part of the connection and one has to deal with only one network technology. Moreover, it protects all the protocol information from the higher layers, but has as disadvantage that the information is not protected in the access point. Protection at the network or transport layer offers the possibility of end-to-end protection independent of the application but it brings more interoperability problems. Some applications may offer security in the application itself; this has as advantage that the security is connected to the end user but this approach requires a different solution for every application. For sensitive applications, protection at the data link layer can be combined with protection at a higher layer.

In order to control access to the network, a mutual entity authentication protocol is executed between the mobile terminal and the access point as explained above. However, the information needed to identify users or devices or to determine access rights is not always available locally; in that case an additional protocol has to be run with a server or central database.

3. Mobile Phone Systems: GSM, 3GSM

In mobile phone systems the phone establishes a wireless connection to a base station; the area served by the base station is called a cell, which explains the name cellular communications. The base station forwards the connection to a base station controller,

which in turn forwards the connection to the fixed network. The first generation mobile phone systems were analog (e.g., AMPS in the US and TACS, ETACS and NMP in Europe). The only security service offered by the first generation systems was a secret user identifier that was sent in clear over the network, similar to user name and password in a computer system. Very quickly cloning attacks were launched: criminals simply captured the secret and reprogrammed it into their own phones, thus allowing them to place phone calls at the expense of another user. This resulted in the development of advanced security services for the second generation mobile systems; these services were further improved for the third generation systems.

3.1. GSM Security

The second generation mobile phone systems are digital; they include GSM and IS-95. The discussion in this paper will focus on the former. The security goals of the GSM system are user identity confidentiality, user identity authentication, user data confidentiality and signaling information confidentiality [V93]. These security goals (except for the first one) are achieved by running an authenticated key agreement (AKA) protocol between the mobile phone and the base station. This protocol requires a long term secret or key that is stored in a smart card called the Subscriber Identity Module (SIM). A smart card is a small tamper resistant microprocessor that can securely store secrets and perform cryptographic computations. The smart card is inserted in a special slot of the mobile phone.

When a user registers with a mobile operator, a user name (International Mobile Subscriber Identity or IMSI) and a 128-bit secret key K_i are stored on the SIM; a second copy of K_i is kept by the operator in an on-line database known as the authentication center. When a user turns on his phone, the protocol of Figure 1 is run between the base station and his phone. In a first step, the phone sends the IMSI to the base station; in response, the base station sends a 128-bit random challenge *RAND* to the phone. The phone forwards this challenge to the SIM. The SIM applies the authentication function to compute the 32-bit response $RES = A3(K_i , RAND)$ and sends *RES* to the phone which forwards it to the base station. Here A3 is a MAC algorithm (see [MOV97] for the definition of a MAC algorithm). The base station can check this response; if it is correct, it is convinced that it is indeed talking to the right phone. If the protocol would stop here, it would still be feasible for an attacker to take over the connection after a successful authentication of the phone. In order to preclude such a connection hijacking, both parties derive from *RAND* a 64-bit secret session key $K_c = A8(K_i , RAND)$, where A8 is a pseudo-random function [MOV97]. This key is computed in the SIM card but forwarded to the mobile phone. All subsequent communication between the phone and the base station is encrypted in the phone in hardware using K_c. An attacker does not know K_c and hence he is unable to hijack the connection. The encryption algorithm is called A5. Note that while A3 and A8 are computed in the SIM and can be operator specific, A5 is a GSM-wide standard implemented in every GSM phone.

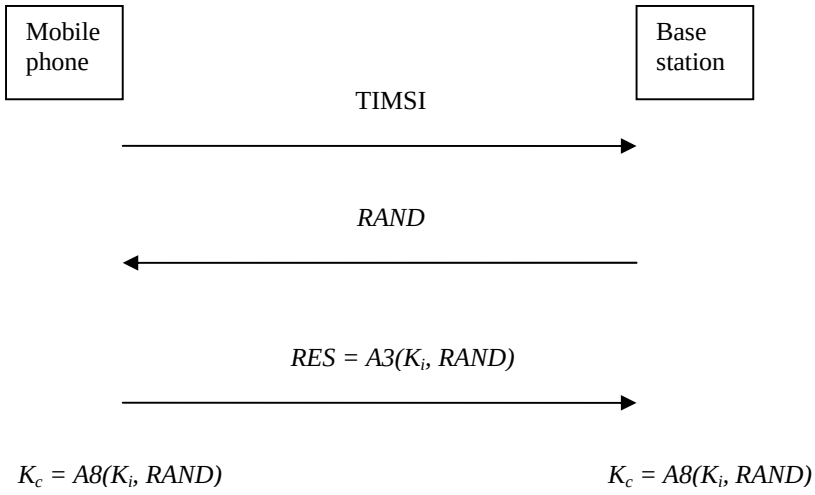

Figure 1: GSM Authenticated Key Agreement Protocol.

There are still some problems with the above protocol. First, for security reasons it would not be appropriate to store the keys K_i of all the users in the base stations or to forward these keys to a base station. If the mobile phone is in the area covered by the home network of the user, a real-time connection is set up with the authentication center which forwards to the base station the triplet (*RAND*, *RES*, K_c). Knowledge of this triplet is sufficient to complete the AKA protocol. If the mobile phone is roaming, that is, it is connected to another network, a small set of triplets is sent over the network by the home network in order to reduce the overhead and network delays. A second problem is that the protocol of Figure 1 does not offer subscriber identity confidentiality. Indeed, by eavesdropping the wireless link one can obtain the IMSI which allows to trace a mobile phone. This is resolved by assigning a new temporary IMSI (TIMSI) to the user after each authentication. This TIMSI is sent to the phone encrypted under the session key K_c which implies that an attacker has no access to this pseudonym. For the next run of the AKA protocol, the mobile phone sends in the first protocol step the TIMSI rather than the IMSI. A third issue is that the AKA protocol authenticates the SIM card to the network, which is not the same as authenticating the user of the mobile phone to the network. User authentication can be achieved by a locally verified PIN code.

In the absence of active attacks, that is, attackers who impersonate the network to the user, the GSM AKA protocol achieves its security goals, which were intended to increase the security level of the wireless link roughly to that of the fixed system. However, it should be pointed that security level of the fixed networks is very low and one can argue

that even this security level has not been achieved. The weaknesses of the GSM security system can be divided into three classes: weaknesses of cryptographic algorithms, lack of protection against active attacks (a cryptographic protocol weakness) and architectural weaknesses.

The GSM cryptographic algorithms were developed by the ETSI group SAGE in the late 1980s, at a time when governments wanted complete control over the use of cryptography and only deliberately weakened confidentiality algorithms were approved for consumer use. Initially, two secret encryption algorithms were provided: A5/1 was designed in 1987 and A5/2 was added in 1989 for use outside Europe. Both A5/1 and A5/2 are additive stream ciphers with clock control. A5/2 looks slightly more complex than A5/1, but it is far less secure. The general design was leaked in 1994 and the algorithms were entirely reverse engineered in 1999. Very quickly, cryptanalytic attacks were published. The currently best known attacks on A5/1 and A5/2 are the ciphertext only attacks of Barkan *et al*. [BBK03]. Their attack on A5/1 requires a few minutes of ciphertext and 50 Terabytes of disk space to recover the 64-bit key K_c in a few minutes on a PC; their attack on A5/2 requires a few milliseconds of ciphertext and a few Gigabyte of disk space to recover the key in less than a second on a PC. In 2006 A5/2 has been removed from the GSM standards. The above attacks exploit the weaknesses of A5/1 and A5/2, but they also rely on the fact that in GSM adding redundancy for channel coding is performed before encryption, while every textbook in cryptography explains that one should perform channel coding after encryption. These attacks could be further improved because operators reduce the effective key size from 64 bits to 54 bits by setting ten key bits to 0. This practice reduces the cost of a brute force key search with dedicated hardware well below $1 per key. In the late 1990s, during the development of 3GSM, a new block cipher KASUMI was developed by SAGE. By then the political climate had changed: the algorithm was reviewed by academic research teams and published for open review. In 2003 a mode of this block cipher was introduced in GSM under the name A5/3. Unfortunately, because A5/3 needs to be implemented in hardware in the phones and the base stations, the upgrade of A5/1 to A5/3 progresses very slowly. Moreover, the limitation to 64-bit security is a serious concern: today the cost to recover such a key in a day is about $100,000 and the cost per key is $50-100; by Moore's law, this cost will be halved every 18 months [ECRY06].

The secret MAC and key derivation algorithms A3 and A8 are implemented in the SIM and in the authentication center, which means that they are operator specific. The GSM Memorandum of Understanding (MoU) document provided a secret algorithm COMP128 as "example" for A3/A8. The algorithm, which was used by many operators, leaked out in 1998, and was quickly shown to be very weak: the 128-bit key K_i can be recovered with 2^{17} chosen *RAND* values that can be obtained in a few hours [BGW98,HP98]. At the time the MoU was written, it was well known by experts that COMP128 was insecure; some observers believe that the algorithm may have been left in the MoU by the large telecom players to mislead the newcomers in the mobile market.

An even more serious security concern of the GSM AKA protocol is that the mobile phone does not authenticate the base state or the network: there is only unilateral entity

authentication. This implies that active attacks are feasible, in which an attacker sets up a false base station and impersonates the network to the user. A false base station can tell the mobile phone to switch off encryption or to revert to a weaker encryption algorithm (possibly with the same session key). Subsequently the communication can be hijacked allowing for so-called dynamic cloning attacks, in which fraudulent calls can be made at the user's expense. An active attacker can also inform the mobile phone that he lost the current TIMSI and request to send (in clear) the IMSI, which allows to track users.

Some architectural decisions also present problems: while one can argue that data authentication for voice is overkill, a data authentication service should be required for signaling and for data communication such as SMS. The encryption is restricted to the wireless link between the mobile phone and the base station, while often base stations communicate over microwave links to base station controllers, resulting in transmissions in clear over these links. The AKA triplets are typically sent in clear over the fixed network, which means that they can be intercepted. The standards do not impose that the mobile phone indicates to the user whether encryption is on or off – this is probably the consequence of the requirements by national security or law enforcement in order to have the ability to switch off encryption in a stealthy way. There is no law enforcement interface, which means that under certain circumstances operators could be forced to turn over databases with user keys K_i or to derive the user keys from a single master key. The home network cannot verify if authentication is properly used when users roam to other networks, which may lead to fraud (overcharging of roaming users). A very fundamental problem is that the GSM system has not been designed with sufficient flexibility in mind: there are no procedures for introducing new algorithms and protocols. Finally several sources have indicated that it may be possible to remotely activate the microphone of a mobile phone in order to eavesdrop on a conversation even when no call is being placed or when the phone is switched off.

In spite of these security problems, GSM has been a massive commercial success, which suggests that attacks exploiting these weaknesses for financial gain can be kept under control. On the other hand, there should be no doubt that law enforcement, national security and organized crime have the ability to eavesdrop on GSM conversations.

The security of the other technologies for second generation mobile phones deployed in the US is definitely not better. Serious weaknesses have been identified in the cryptographic algorithms, encryption is not widely supported and in areas with poor coverage some phones fall back to analog mode (without any security).

3.2. 3GSM Security

The goal of 3GSM security architecture was to stay as close to the GSM system as possible for backwards compatibility reasons, while correcting the weaknesses of GSM [BHHN02]. 3GSM is the marketing name for a technology that was developed under the name of UMTS (Universal Mobile Telecommunication System) and managed by 3GPP (Third

Generation Partnership Project). The security architecture of 3GSM was completed in the late 1990s; the deployment of 3GSM technology started only in 2003. Competing technologies are CMDA2000 (US, India and China) and the Chinese standard TD-SCDMA.

3GSM extends the GSM AKA protocol to support mutual entity authentication (see also Figure 2). For this purpose, the network sends together with the 128-bit string *RAND* a 64-bit MAC value $MAC = f1(K_i, SQN \| RAND \| AMF)$ on the 48-bit sequence number *SQN*, *RAND* and a 16-bit authentication management field *AMF* that allows to trigger session key changes or cryptographic algorithm upgrades. By verifying that the sequence number is in the expected range and by checking the MAC value, the phone can check that it is talking to the correct network and that the message is not a replay of an old message. Subsequently it sends a response $RES = f2(K_i, RAND)$ to the network; *RES* can be between 32 and 128 bits long. The additive encryption is complemented by a MAC algorithm on the user and signaling data, hence providing authenticated encryption. This implies that two session keys are derived, a ciphering and an integrity key ($CK = f3(K_i, RAND)$ and $IK = f4(K_i, RAND)$ respectively). A third (optional) anonymity key $AK = f5(K_i, RAND)$ can be exored with *SQN*, in order to prevent attacks that trace the users based on their *SQN* values. The GSM triplet ($RAND, RES, K_c$) is thus replaced by a quintet (*RAND*, *RES*, *IK*, *CK*, *AUTN*) with $AUTN = SQN \oplus AK \| MAC \| AMF$. An additional protocol (not described here) is provided to re-synchronize the value of *SQN* using the function f1*. The 3GSM AKA protocol has been thoroughly analyzed, and its security properties are well understood.

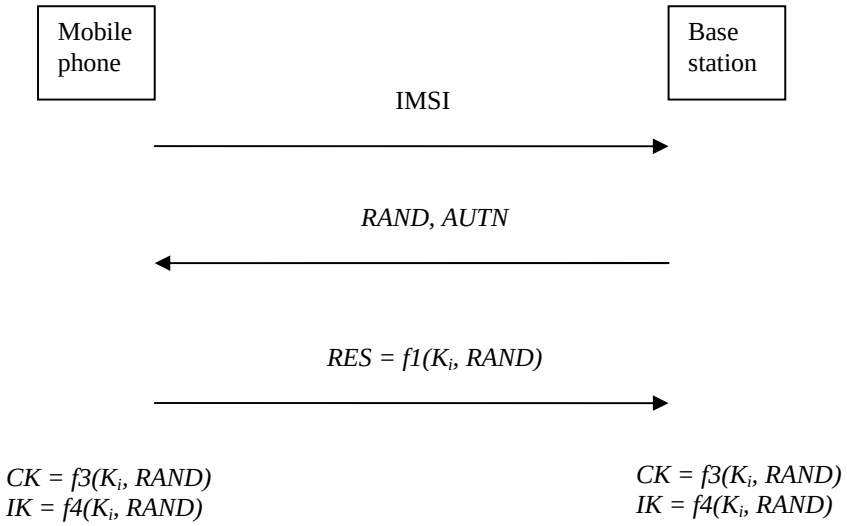

Figure 2: The 3GSM Authenticated Key Agreement Protocol. Here $AUTN = SQN \oplus AK \,||\, MAC \,||\, AMF$, $AK = f5(K_i, RAND)$ and $MAC = f1(K_i, SQN \,||\, RAND \,||\, AMF)$

The encryption of voice, data and signaling information is performed using f8, which is a mode of the block cipher KASUMI (64-bit block length and 128-bit key); this mode combines the Output Feedback Mode (OFB) with the CounTeR Mode (CTR) [SP800-38a]. The same information is authenticated using a MAC value computed using f9, that is a variant of CBC-MAC again based on the block cipher KASUMI. Both f8 and f9 are implemented in hardware in the phone (similar to A5). The algorithms f1, f1*, f2, f3, f4, and f5 are operator specific and are implemented in the USIM (the 3GSM SIM). For various reasons, 3GSM operators tend to prefer to use proprietary algorithms, but a well evaluated suite called MILENAGE is described in the 3GSM standard; it is based on the AES algorithm [FIPS197].

The cryptographic algorithms have been designed by ETSI SAGE, but KASUMI and MILENAGE have been made public and have so far withstood all cryptanalytic attacks. The two session keys *CK* and *IK* can be up to 128 bits long; even if strong encryption is deemed to be undesirable, active attacks and session high-jacking can be prevented by using a 128-bit integrity key *IK*.

The encryption in 3GSM is continued beyond the base station to the base station controller. 3GSM also provides application level security and network security, for example to securely transmit quintets over the network. The 3GSM security supports many other features such as law enforcement access, fraud information gathering, location services security, and mobile IP security, but a detailed treatment of this topic is beyond the scope of this article (see e.g., [BHHN02]). One can conclude that the access security of 3GSM is more than adequate and that substantial progress has been made on network level security. On the other hand, it is also clear that if these networks move towards closer integration with the Internet, all network level attacks which are visible there (including viruses, worms, denial of service attacks, DNS level attacks) will need to be addressed on mobile networks as well.

4. Wireless LAN

The IEEE 802.11 standard for wireless local area networks (also known as Wi-Fi) currently supports multiple over-the-air modulation techniques in the 2.4 GHz and 5 GHz frequency bands with speeds between 11 and 540 Mbit/s. In the most common setup, the infrastructure mode, a computer or a mobile phone connects to an access point, which offers further connection to the fixed Internet. The area covered by a single access point is known as a hotspot. The IEEE 802.11 standard also allows for mesh networks and for peer-to-peer (wireless ad hoc) connections. In this paper we will restrict ourselves to the infrastructure

mode, which is very similar to the GSM and 3GSM setup with the role of the base station played by the access point.

The security goals of the IEEE 802.11 security architecture are entity authentication, authorization, data confidentiality and data integrity. Note that anonymity of users w.r.t. third parties is not a requirement. Security is offered at the data link layer. The first solution included in the 1999 standard IEEE 802.11 was the Wired Equivalent Privacy (WEP) [IEEE802.11]. Very quickly multiple security flaws were discovered in WEP, resulting in the adoption in 2002 of Wi-Fi Protected Access (WPA) as an intermediate solution by the industry consortium Wi-Fi Alliance. In 2004, IEEE ratified the 802.11i standard [IEEE802.11i], also known as WPA2, that resolves the security weaknesses of WEP. These protocols are discussed in more detail below.

4.1. Wired Equivalent Privacy (WEP)

The WEP protocol is an optional security protocol for IEEE 802.11 that intends to offer authenticated encryption at the data link layer. The encryption is provided by the stream cipher RC4 and the data authentication is implemented using a MAC computed with a linear function CRC-32 (Cyclic Redundancy Check). RC4 is a stream cipher that stretches a short key *K* (here 42 or 104 bits) to a long key stream to be added to the data. In order to generate a different key stream for each packet (packets contain up to 1500 bytes), the secret key is prepended with a 24-bit Initial Value (*IV*) that is chosen for each packet. WEP offers the choice between open systems (without authentication of the mobile nodes) or a shared key protocol. The latter uses a challenge response protocol: a random challenge is sent (equivalent to the GSM *RAND*) by the access point and the mobile node applies the combination of RC4 and CRC-32 as described above. In addition, vendor specific schemes are deployed that are based on the MAC address or the network identifier known as the Service Set Identifier (SSID) – these mechanisms can be defeated easily. No key management is provided, hence each access point uses a single key shared by its users, that has to be installed manually and is thus updated infrequently.

The WEP scheme is an ideal didactical example, since it makes virtually all the mistakes that can be made by a cryptographic protocol designer in a single protocol [BGW01,HA03].

- The 40-bit key size is too small; the cost of recovering such a key is very low today (less than $1). A 104-bit key offers sufficient long-term protection, but in some implementations it is derived from a short password resulting in weak security.
- The *IV* size is too small, so *IV*s will repeat very quickly. If the *IV* repeats, the sum of two plaintexts can be recovered by exoring the two ciphertexts (the identical key stream will cancel out); statistical analysis will easily yield the two plaintexts. If *IV*s are chosen at random, they will repeat by the birthday paradox [MOV97] after about 5000 packets. Some implementations use a counter, which is frequently initialized at 0, which resulted in repeats after resets. Even if the starting point is chosen at random, one can expect repetitions after resets based on the birthday

paradox. One could also use a known plaintext attack to build a 24 Gigabyte table that contains the key stream for each *IV* value; knowledge of such a table is essentially equivalent to knowledge of the secret key.

- If entity authentication is active, a passive attacker can observe a challenge *RAND* and the response RC4(*IV* || *K*) ⊕ *RAND*. This yields the key stream RC4(*IV* || *K*), which allows to impersonate a mobile node, by letting this false node always use the same value *IV*.
- A linear MAC algorithm does not offer any data integrity: one can add an arbitrary string to the plaintext and compute the correction that needs to be added to the MAC value without knowing any secret. If modifications are inserted in the packet headers, the packet will be decrypted by the access point and may be diverted to a machine of the attacker's choice.
- Until 2006, the most effective attack was based on a cryptographic weakness of RC4. Fluhrer *et al.* [FMS01] showed that the concatenation of *IV* and *K* in RC4 allows for the extraction of *K* in a passive attack (only eavesdropping); their attack requires about 1 million *IV*s and a few days to recover a 104-bit key; shorter keys can be recovered in a few hours. Filtering of some weak *IV*s could increase the complexity of this attack, but subsequently a more sophisticated active variant of was developed [A01]. These attacks were made available as tools that required a few hours even for longer keys. The response of the vendors was to provide key management protocols for frequent rekeying of WEP using EAP (cf. Section 4.2).

In 2006, a large fraction of WLANs is still unprotected; among those that use security, more than 75% still use the WEP protocol. This observation motivated Bittau *et al.* [BHL06] to further optimize the existing attacks by exploiting fragmentation. Their attack requires less than a minute to allow an opponent to send and divert packets (hence making frequent rekeying useless) and requires fifteen minutes to recover 40-bit keys and two hours for 104-bit keys.

4.2. Wi-Fi Protected Access (WPA)

WPA is a short term solution for WEP that was developed in 2002 in anticipation of the 2004 publication of the IEEE 802.11i standard (WPA2). As WPA had to be compatible with the deployed hardware with limited computational power, it kept using RC4, but replaced the CRC-32 with a stronger MAC algorithm with a 64-bit result called Michael. The Michael algorithm had to execute in less than 5 cycles per byte and even if it is much more secure than CRC-32 it is known to have weaknesses [MRH04]. The Temporal Key Integrity Protocol (TKIP) doubles the *IV* space to 48 bits with sequencing rules and adds a mechanism to derive per-packet WEP keys from a temporal secret key, the MAC address of the device and the packet sequence number. This ensures unique keys even if multiple nodes share the same secret key. The temporal key is derived from the pair-wise master key (PMK) and is changed every 10,000 packets. The PMK is computed either from a manually installed pre-shared key (typical for home networks) or from a key established using the EAP protocol discussed below. None of the existing attacks against WEP seem to apply to WPA.

WPA allows the use of the IEEE 802.1X framework for port based network access control; this protocol authenticates user or devices and allows for establishment of the PMK key [IEEE802.1X]. This framework is based on the Extensible Authentication Protocol (EAP, [RFC3748]) that supports multiple authentication methods, such as smart cards, one-time password tokens, Kerberos, and public key authentication. When a mobile node requests access, the access point opens a port only for EAP packets to an authentication server (e.g., RADIUS or Diameter) on the fixed network. All other traffic is blocked at the data link layer. After successful entity authentication, the access point will allow normal traffic. EAP also provides for a log-off message.

4.3. Robust Secuurity Network (RSN) or WPA2

The IEEE 802.11i standard ratified in 2004 defines a new type of wireless network called a robust security network (RSN). For backward compatibility, 802.11i supports the parallel use of RSN and WEP. The architecture of 802.11i is very similar to that of WPA. RSN supports IEEE 802.1X for port based access control and EAP for authenticated key agreement. The main difference is that authenticated encryption is provided based on TKIP (as in WPA) or on CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol) based on AES [FIPS197]. The CCMP mode uses the CounTeR (CTR) mode for data confidentiality and CBC-MAC for data authentication; it offers a higher security level than the RC4-based mechanisms but requires more computational power.

RSN supports a rather complex key hierarchy with four or five levels: pre-shared key or AAA key (the latter is established using EAP), pairwise master key (PMK), pairwise transient key established during a 4-way handshake and temporal key. For TKIP a per-packet key is derived from the temporal key, while for CCMP the temporal key is used to protect multiple packets. In addition, a group key can be established for multicast or broadcast traffic. A detailed discussion of the complete functionality is outside the scope of this article; the interested reader is referred to [IEEE802.11i,SP800-97].

5. Personal Area Network: Bluetooth

The Bluetooth standard describes how mobile phones, computers, PDAs, headsets and other mobile devices can establish a short-range wireless channel. The Bluetooth Special Interest Group (SIG) was founded in 1998, and in 2000 the standard was included in the Wireless Personal Area Network Working Group [IEEE802.15]. We do not cover alternative technologies such as Zigbee (IEEE 802.15.4) and Ultrawideband, since for now their deployment is very limited and little information is available on their security. Unlike in Sections 3 and 4, the interaction between two Bluetooth devices is a symmetric interaction, which explains the name “pairing.”

The Bluetooth authenticated key agreement protocol is rather complex. Each Bluetooth device generates at first power-up a unit key, which is stored in non-volatile memory. The key is generated as a function of the 48-bit Bluetooth address and a random number. When two Bluetooth devices want to run an AKA protocol, they go through the following steps:

1. Device A that initiates the communication computes the initialization key K_{in} from a random number R generated by the device, a Personal Identification Number (PIN) entered by the user and the length L of the PIN. The random number R is transmitted to device B and the PIN has to be entered in device B, hence B can also compute K_{in} .
2. Device B now authenticates itself to device A as follows: B sends his Bluetooth address ADR_B to A, A generates a random number R_A and sends it to B and B sends the value $E_1(ADR_B \| K_{in} \| R_A)$ to A. Similarly, A authenticates itself to B.
3. The two devices generate a shared link key K_{lin} from K_{in} and the values exchanged so far. Subsequently they discard K_{in} and run a new mutual entity authentication protocol based on K_{lin}.
4. The encryption key K is now computed from K_{lin}, a random number (generated by A) and a value computed by both devices during step 2.
5. In order to generate the stream to encrypt the data, the encryption key K (possibly reduced in length) is sent to E_0 together with the Bluetooth address and some clocking information. All data sent between A and B is encrypted using the key stream generated by E_0.

The Bluetooth protocol offers several fallback modes: for devices such as earsets in which a PIN cannot be entered, a default PIN of 0000 is assumed. If one of the devices cannot store a link key for each connection, the unit key of this device is used as link key; it is sent to the other device encrypted under the initialization key K_{in}. This clearly creates a weakness since the unit key of a device is a permanent key. Fortunately the use of the unit key is deprecated in version 1.2 of the Bluetooth specifications. Note also that devices can be configured to operate without any security.

The cryptographic algorithms in Bluetooth are SAFER+ [MKK98] for entity authentication and key establishment and the stream cipher E_0 for data encryption. No weaknesses have been identified in SAFER+, but today more lightweight block ciphers are available that offer a similar security level. The stream cipher E_0 is much less secure than anticipated: even if it has a 128-bit key, the best attack on E_0 requires knowledge of the first 24 bits of $2^{23.8}$ frames and time 2^{38} to recover the secret key [Lu04]. Note also that the Bluetooth protocol provides data confidentiality but it does not offer data authentication.

The protection is provided at the data link level, which means that all addresses are sent in the clear. By eavesdropping on a Bluetooth exchange, or by contacting a device that is in discoverable mode, an attacker can obtain the 48-bit Bluetooth address, and subsequently track the device [JW01]. Even if a device is in non-discoverable mode, the 48-bit address may be discovered since it is known to be not random (24 bits are specific for the manufacturer, and some manufacturers preserve certain ranges for specific devices); depending on the information available to an attacker and the number of devices used in the

attack and being targeted simultaneously, recovering an address requires between a few hours to a few years. Moreover, passive attacks allow for the recovery of the PIN by exhaustive search; recovering a PIN up to 6 digits requires less than a second, while recovering an 8-digit PIN requires a few minutes.

The security of the Bluetooth protocol could be improved substantially by using password-based authenticated key exchange (PAKE) protocols [WSC07]; these protocols prevent off-line PIN guessing attacks, but require public-key operations that are more expensive than the symmetric cryptographic operations used in Bluetooth. An alternative improvement would be to regularly update the link key K_{lin}, since an attacker can only guess the PIN if he would be present during each such update.

Bluetooth enabled devices are potentially vulnerable to denial of service attacks: rogue devices could repeatedly attempt connections, and in this way reduce the battery life of the victim [WSC07]. If a blacklisting system is implemented to reject future connections, a denial of service attack could be launched on the blacklisting mechanism itself, for example by initiating a large number of connections on behalf of another legitimate device without ever completing a successful authentication.

The flaws in Bluetooth which have the most impact are related to implementation weaknesses. As an example the Bluesnarf attack [LL03] allows to connect to some mobile phones and to gain access to the restricted portions of the data in the phone, including the entire phone book, calendar, IMEI (International Mobile Equipment Identity), etc. without the owner being alerted. The Bluejacking attack [BLUEJ] exploits the fact that up to 248 characters of the name of the other device is displayed during a pairing protocol to send advertisement to another device. Finally, mobile phone worms have been reported that spread using the Bluetooth functionality (e.g., the Cabir worm).

Attacks on Bluetooth are believed to be manageable since Bluetooth devices have a typical range of up to 10m (class 2), hence it is believed that attackers need to be close to the victim. However, attackers do not need to stick to the communication protocols: it has been demonstrated that with directional antennas the range of a class 2 Bluetooth radios could be extended to 1.8 km, which clearly shows that remote attacks are feasible. At least this result puts the name Personal Area Network in a very different light. A similar comment applies to WLAN, which is designed for an indoor range of 30m and about 100m outdoors. Again, with special antennas the range can be extended to several km.

6. Conclusions

This paper has presented an overview of security in wireless and mobile communications. GSM was the first mass consumer communication system with cryptography; it was well ahead of its time, but it was probably not planned that the system would be still widely used 20 years after its introduction. 3GSM has addressed most of the security flaws in the late

1990s. In view of this, it is rather surprising that the WLAN world did not learn from the GSM mistakes, and had to go through a major redesign between 1999 and 2004. WiMAX (IEEE 802.16), which was not discussed in this paper, seems to have gone through a similar phase: an initial design with flaws based on solutions for fixed networks, that was later on improved using solutions from the WLAN standards [JW04]. Finally the Bluetooth system seemed to have learned from earlier mistakes, but could still benefit from an upgrade to enhance its security.

The major mistake that seems coming back is the lack of mechanisms to upgrade cryptographic algorithms and protocols. While it is clear that adding such a functionality increases complexity and cost, one should balance this to the problem of millions of fielded devices with security problems that cannot be addressed. In addition, while issues related to authenticated encryption, mutual entity authentication and access control are now well understood, most solutions still have problems related to password guessing, privacy and denial of service. Surprisingly, 30 years after its invention public key cryptology is not widely deployed at lower layers in wireless environments. One can expect that in the next decade the cost and usability of public key technologies will be reduced substantially, which will allow to deploy more advanced solution offering better privacy and robustness.

Bibliography

[A01] W. A. Arbaugh. An Inductive Chosen Plaintext Attack Against WEP and WEP2, 2001.

[BGW98] M. Briceno, I. Goldberg, D. Wagner, GSM Cloning, http://www.isaac.cs.berkeley.edu/isaac/gsm-faq.html, 1998.

[BGW01] N. Borisov, I. Goldberg, D. Wagner, Intercepting mobile communications: The insecurity of 802.11," *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking*, ACM Press, 2001, pp.180-189.

[BHL06] A. Bittau, M. Handley, J. Lackey, The Final Nail in WEP's Coffin, *Proceedings IEEE Symposium on Security and Privacy*, IEEE Computer Society, 2006, pp. 386-400.

[BHHN02] K. Boman, G. Horn, P. Howard, V. Niemi, UMTS Security, *Electronics & Communication Engineering Journal*, Vol. 14, No. 5, 2002, pp. 191-204.

[BLUEJ] Bluejacking, http://www.bluejackq.com/.

[DL98] W. Diffie, S. Landau, *Privacy on the Line: The Politics of Wiretapping and Encryption*, MIT Press, 1998.

[ECRY06] ECRYPT, *Yearly Report on Algorithms and Keysizes*, SPA.16 Rev 1.0, IST-2002-507932 ECRYPT, January 2006, http://www.ecrypt.eu.org/

[FIPS197] NIST FIPS 197, *Advanced Encryption Standard*, Federal Information Processing Standard, NIST, U.S. Dept. of Commerce, November 26, 2001.

[FMS01] S.R. Fluhrer, I. Mantin, A. Shamir, Weaknesses in the key scheduling algorithm of RC4, *Workshop on Selected Areas in Cryptography, SAC'01, LNCS 2259*, S. Vaudenay, A.M. Youssef, Eds., Springer-Verlag, 2001, pp. 1-24.

[HA03] R. Housley, W.A. Arbaugh, Security problems in 802.11-based networks, *Communications of the ACM*, Vol. 46, No. 5, 2003, pp. 31-34.
[HP98] H. Handschuh, P. Paillier, Reducing the Collision Probability of Alleged Comp128, *Smart Card Research and Applications, CARDIS 1998, LNCS 1820*, J.-J. Quisquater, B. Schneier, Eds., Springer-Verlag, 2000, pp. 366-371.
[IEEE802.1X] IEEE Computer Society, *Standards for Local and Metropolitan Area Networks: Port-Based Network Access Control*, IEEE Standard 802.1X, 2001, http://standards.ieee.org/getieee802.
[IEEE802.11] IEEE Computer Society, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11, 1999, http://standards.ieee.org/getieee802.
[IEEE802.11i] IEEE Computer Society, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Amendment 6: Medium Access Control (MAC) Security Enhancements, IEEE Standard 802.11i, 2004, http://standards.ieee.org/getieee802.
[IEEE802.15] IEEE Computer Society, *Wireless Medium Access Control (MAC) and physical layer (PHY) specifications for: Wireless Personal Area Networks*, IEEE standard 802.15, 2002, http://standards.ieee.org/getieee802.
[JW01] M. Jakobsson, S. Wetzel, Security Weaknesses in Bluetooth, *Proceedings of the Cryptographer's Track at the RSA Conferecen, CT-RSA '01, LNCS 2020*, D. Naccache, Ed., Springer-Verlag, 2001, pp. 176-191.
[JW04] D. Johnston, J. Walker, Overview of 802.16 Security, *IEEE Security & Privacy Magazine*, Vol. 2, No. 3, 2004, pp. 40-48.
[LL03] A. Laurie, B. Laurie, Serious Flaws in Bluetooth Security Lead to Disclosure of Personal Data, http://bluestumbler.org/, 2003.
[LS04] Y. Lu, S. Vaudenay, Faster Correlation Attack on Bluetooth Keystream Generator E0, *Advances in Cryptology, CRYPTO 2004, LNCS 3152*, M.K. Franklin, Ed., Springer-Verlag, 2004, pp. 407-425.
[MKK98] J. Massey, G. Khachatrian, M. Kuregian, Nomination of SAFER+ as Candidate Algorithm for the Advanced Encryption Standard (AES), June 1998.
[MRH04] V. Moen, H. Raddum, K.J. Hole, Weaknesses in the Temporal Key Hash of WPA, *Mobile Computing and Communications Review*, Vol. 8, No. 2, 2004, pp. 76–83.
[MSCP04] R. Maier, V. Sdralia, J. Claessens, B. Preneel, Security Issues in a MobileIPv6 Network, *Security for Mobility, IEE Telecommunications 51*, C.J. Mitchell, Ed., The Institution of Electrical Engineers, 2004, pp. 269-284.
[MOV97] A. Menezes, P. van Oorschot, S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.
[RFC3748] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz, *Extensible Authentication Protocol (EAP)*, RFC 2284, June 2004.
[SA99] F. Stajano, R. Anderson, The Resurrecting Duckling: Security Issues in Ad-Hoc Wireless Networks, *Workshop on Security Protocols, LNCS 1796*, B. Christianson, B. Crispo, M. Roe, Eds., Springer-Verlag, 1999, pp. 172-194.
[SP800-38a] NIST, SP 800-38A, *Recommendation for Block Cipher Modes of Operation - Methods and Techniques*, December 2001.
[SP800-97] NIST, *Guide to IEEE 802.11i: Robust Security Networks*, Draft Special Publication 800-97, June 2006.

[V93] K. Vedder, Security Aspects of Mobile Communications, *Computer Security and Industrial Cryptography, LNCS 741*, B. Preneel, R. Govaerts, J. Vandewalle, Eds., Springer-Verlag, 1993, pp. 193-210.
[WSC07] F.-L. Wong, F. Stajano, J. Clulow, Repairing the Bluetooth Pairing Protocol, *Workshop on Security Protocols 2005, LNCS*, Springer-Verlag, in print.
[ZH99] L. Zhou, Z. Haas, Securing Ad Hoc Networks, *IEEE Network Magazine Special Issue on Network Security*, Vol. 13, No. 6, 1999, pp. 24-30.